

**AN INTELLIGENT APPROACH TO DESIGN A PERSONALIZED
SEARCH SYSTEM USING NEXT GENERATION
BIG DATA ANALYTICS**

A Thesis

Submitted for the award of the Ph.D. degree

In Computer Science and Informatics

(Faculty of Science)

to the

UNIVERSITY OF KOTA

By

Dheeraj Malhotra



Under the Supervision of

Dr. O.P. Rishi

Associate Professor

**Department of Computer Science & Informatics
UNIVERSITY OF KOTA, KOTA**

RAJASTHAN, INDIA

2019

SUPERVISOR'S CERTIFICATE

I feel great pleasure in certifying that the thesis entitled “**An Intelligent Approach to Design a Personalized Search System using Next Generation Big Data Analytics**” has been carried out by **Dheeraj Malhotra** under my guidance. He has completed the following requirements as per Ph.D. regulations of the University.

- a) Course work as per the University rules
- b) Residential requirements of the University (200 days)
- c) Regularly submitted the annual progress report
- d) Presented his work in the departmental committee
- e) Published two research papers in a refereed research journal and two papers in conference proceedings

I recommend the submission of the thesis.

(Dr. O.P. Rishi)
(Supervisor)

ANTI-PLAGIARISM CERTIFICATE

It is certified that Ph.D. Thesis Titled “**An Intelligent Approach to Design a Personalized Search System using Next Generation Big Data Analytics**” by **Dheeraj Malhotra** has been examined with the anti-plagiarism tool. We undertake the follows:

- a. The thesis has significant new work/knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced
- b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as the author's own work
- c. There is no fabrication of data or results which have been compiled and analyzed
- d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record
- e. The thesis has been checked using URKUND software and found within limits as per HEC plagiarism policy and instructions issued from time to time

(Dheeraj Malhotra)

(Research Scholar)

Place: Kota

Date:

(Dr. O.P. Rishi)

(Research Supervisor)

Place: Kota

Date:

ABSTRACT

In the present era of big data, web page searching and ranking in an efficient manner on the World Wide Web to satisfy the specific search needs of the modern user is undoubtedly a major challenge for search engines. Generally, each user has different information requirements. Thus, search results should be adapted to the user's needs. Even though a large number of web search techniques have been developed, some problems still exist while searching with generic search engines as none of the search engines can index the entire web. The issue is not just the volume but also the relevance concerning the user's requirements.

Moreover, if the search query is partially incomplete or is ambiguous, then most of the modern search engines tend to return the result by interpreting all possible meanings of the query. Concerning search quality, more than half of the retrieved web pages have been reported to be irrelevant. Hence web search personalization is required to retrieve search results while incorporating the user's interests. The present research work first compares various existing approaches to personalization. This research work then carries out the detailed comparison between different available big data deployment platforms over the cloud and chooses the second generation of the Hadoop, i.e. Hadoop 2 based cloud framework due to its highly optimized features for deployment of the recommended personalized search system.

The contribution of the current research work is the merging of machine learning capabilities with big data analytics for the design and development of a novel, personalized page search, and ranking algorithm, i.e., Advanced Cluster Vector Page Ranking (ACVPR) algorithm. The ACVPR algorithm is further used to implement an Intelligent Meta-Search System (IMSS) tool to assist the end user in

listing the most relevant web pages in the ranking order of his or her preferences effectively and efficiently. The suggested approach can significantly sort out the incomplete indexing of the web as the recommended tool being metasearch tool uses three popular background search engines, Google, Qwant and Bing in the background to retrieve web links and then re-rank the results to best match the personalized search requirements of the user. The Hadoop implementation of the recommended algorithm is helpful in handling big data in the form of a massive number of web links returned by the three giant search engines working in the background of deployed meta-search tool and hence to sort the web links in real time.

To check the personalized search effectiveness of the recommended approach, we have implemented the Advanced Cluster Vector Page Ranking (ACVPR) algorithm in the form of the Intelligent Meta Search System (IMSS) tool. This tool is a machine learning enabled program that uses a Python interpreter on the server side for the implementation of data analysis and recommended algorithm. The results of the analysis are processed using PHP, HTML 5, and CSS 3. This tool is deployed on the Google cloud engine with Hadoop 2 features enabled for the computation of user similarity, rating based recommendations. The tool uses MySQL engine for the user and search query management. The recommended search system employs logistic regression and collaborative filtering based machine learning techniques for personalized search recommendations. The system requires a user to sign up and answer a few questions for the first time within the *Add Skills* section for emotional and behavior analysis. The system will then determine an existing best match user ID for a new user through similarity score calculation. Detailed information about the best match along with similarity information will be shown to the user under the heading of the machine learning summary on the tool interface. If the search query of a new user includes

keywords of the search query of previous users with the best match ID or same profession type, then the system will recommend the search queries of the prior user to the current user. All web links in the result presented to the user will allow the user to rate the rank and relevance of the output link. The rating provided by the user will be used to alter the priority or rank of the output link when another or the same user is searching a query including keywords of previously searched queries by users with a similar profile and preference given to the queries of the best match user as recommended by the system. The deployed system can work in two modes: personalized search mode, and advanced search mode. The advanced search mode, unlike the personal search mode, lets the user select search engines among Google, Bing, and Qwant to be used as background search engines for the deployed meta-search tool. Moreover, a user can also sort the output links in the order of page loading speed within advanced search mode.

The extensive experimental evaluation leads to the determination of various evaluation metrics depending on the machine learning model employed. The present research work, one by one, uses two machine learning models, i.e., (i) logistic regression and (ii) collaborative filtering. The logistic regression model is used to predict feedback of a user for a listed web link and hence to determine the correct rank of the web link in the output of deployed meta-search tool. The collaborative filtering model is used to predict the best match user ID for query disambiguation and personalized search query recommendations. The evaluation metrics for deployed collaborative filtering model include Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The lower values of MAE and RMSE when compared with baselines indicate the effectiveness and efficiency of the deployed collaborative model. The evaluation matrices are also determined such as DF matrix, user similarity matrix, actual response matrix, and prediction matrix. These evaluation metrics and matrices are used to determine the best

match user ID by the system. However, evaluation metrics like specificity, sensitivity, precision, and recall are calculated for the regression-based learning model. The improved machine learning capabilities are demonstrated by Receiver Operating Characteristic (ROC) curves. These ROC curves are generated from the regression-based model using a popular statistics tool, R. The improvement in different evaluation metrics and user survey confirms the improved effectiveness and efficiency of the recommended approach for web search personalization when compared with the professional search engines, popular recommendation approaches and baselines studies discussed in the literature.

Keywords: Web Search Personalization; Meta Search Tool; ACVPR Algorithm; IMSS Tool; Machine Learning; Hadoop 2 and Map Reduce; Big Data Analytics; Collaborative Filtering; Logistic Regression

CANDIDATE'S DECLARATION

I, hereby, certify that the work, which is being presented in the thesis, entitled “**An Intelligent Approach to Design a Personalized Search System using Next Generation Big Data Analytics**” in partial fulfillment of the requirement for the award of the degree of Doctor of Philosophy, carried under the supervision of **Dr. O.P. Rishi** and submitted to the University of Kota, Kota represents my ideas in my own words and where others ideas or words have been included. I have adequately cited and referenced the original sources. The work presented in this thesis has not been submitted elsewhere for the award of any other degree or diploma from any Institutions. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will cause for disciplinary action by the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date:

(DHEERAJ MALHOTRA)

This is to certify that the above statement made by Dheeraj Malhotra (Regd. No. RS/2304/16) is correct to the best of my knowledge.

Date:

(Dr. O.P. Rishi)

(Supervisor)

Dept. of Computer Science & Informatics,
University of Kota- Kota

ACKNOWLEDGMENT

Words fall short of thanking the Almighty God who has given me this life, family, opportunities, and education. Therefore it is indeed a matter of great pleasure to express the deepest gratitude for a never-ending shower of His blessings and to thank all those who gave a helping hand in the completion of this work.

First and foremost, I would like to express my appreciation and sincere gratitude to my research supervisor, Dr. O.P. Rishi, Associate Professor, Department of Computer Science and Informatics, University of Kota for his guidance, support, understanding, and patience. I have been amazingly fortunate to have a best mentor and advisor I could have ever wished. I learned a lot from his advice and useful insights throughout my Ph.D. program. His regular encouragement regarding referred publications helped me a lot to overcome my doubts and to finish this dissertation work.

I convey my deepest gratitude to my parents, my wife, sibling, and all other family members for their loving support and for always standing behind me to cheer me in good times and encourage me in bad times.

I want to convey my sincere appreciation to Dr. S.C Vats, Chairman of Vivekananda Institute of Professional Studies, GGSIPU, Delhi for his continuous motivation and support. Apart from being a great leader, he is also a great philosopher and an inspirational wildlife photographer. I am profoundly thankful to respected Management Members, Principal Director and Dean-VSIT of Vivekananda Institute of Professional Studies, Delhi, for their support during this journey.

I extend my gratitude to Mr. David Pallai and Ms. Jennifer Blaney of Mercury Learning and Information, Duxbury, USA. They enriched my publication skills while pursuing alongside journey towards realizing a dream to author and publish my first book on “Data Structures and Program Generation Using C” in the United States of America.

I would also like to express my sincere thanks towards the library staff, teaching and non-teaching staff of the Department of Computer Science & Informatics and Directorate Research of the University of Kota for assisting me throughout my doctoral studies.

Finally, I convey my special acknowledgments to the number of National and International researchers for referring and citing my publications indexed in Web of Science-Clarivate Analytics, Scopus, Google Scholar, and Mendeley Stats.

Date:

(Dheeraj Malhotra)

Place: Kota

CONTENTS

TITLE	PAGE NO.
Title Page	i
Supervisor's Certificate	ii
Anti- Plagiarism Certificate	iii
Abstract	iv
Candidate's Declaration	viii
Acknowledgment	ix
List of Tables	xviii
List of Figures	xix
List of Abbreviations	xxv
CHAPTER 1 INTRODUCTION	1 - 23
1.1. INTRODUCTION	1
1.2. WEB INFORMATION RETRIEVAL	4
1.2.1. Search Queries	4
1.3. RESEARCH PROBLEM	5
1.4. OBJECTIVES OF RESEARCH	7
1.4.1. Objectives Accomplishment	8
1.5. CONTRIBUTION FROM THE STUDY	9
1.6. COMPARISON OF PLATFORMS FOR BIG DATA ANALYTICS	10
1.6.1. Types of Existing Deployment Paradigms	10
1.6.2. Hadoop 2 : Recommended Framework	11

1.6.3. BDAS vs. Hadoop 2	12
1.6.4. Ranking Comparison of Existing and Prescribed Platforms	14
1.6.5. Google Cloud Platform for Big Data Analytics	15
1.6.5.1. Cloud Cluster Configuration of Meta Search Tool	15
1.7. MOTIVATION	18
1.8. ORGANIZATION OF THE THESIS	21
1.9. CHAPTER SUMMARY	23
CHAPTER 2 LITERATURE REVIEW	24-46
2.1. INTRODUCTION	24
2.1.1. Review of Search Systems based on Hyperlinks	25
2.1.2. Review of Search Systems based on Content Personalization	27
2.1.3. Review of Search Systems based on Contextual Knowledge	31
2.1.4. Review of Search Systems based on Recommendation	34
2.2. DETAILED COMPARISON OF VARIOUS VERSIONS OF INTELLIGENT META SEARCH SYSTEM	37
2.3. LIMITATIONS OF EXISTING SYSTEMS	41
2.4. COMPARATIVE STUDY ON SEARCH SYSTEMS	43
2.5. CHAPTER SUMMA	46

CHAPTER 3	SYSTEM ARCHITECTURE	47-66
3.1.	INTRODUCTION	47
3.2.	CLOUD SERVICE MODELS	49
3.3.	COMPARISON OF DEPLOYMENT PLATFORMS	49
3.3.1.	Types of Deployment Platforms	50
3.3.2.	Second Generation HDFS	50
3.3.3.	Ranking Comparison of Deployment Platforms	53
3.4.	CLOUD ARCHITECTURE OF THE IMSS SYSTEM	54
3.4.1.	The Map-Reduce Programming Model	55
3.5.	HADOOP IMPLEMENTATION OF ACVPR ALGORITHM AND IMSS TOOL	60
3.6.	ADVANTAGES OF CLOUD DEPLOYMENT FOR IMSS TOOL	65
3.7.	CHAPTER SUMMARY	66
CHAPTER 4	GOOGLE CLOUD PLATFORM	67-100
4.1.	INTRODUCTION	67
4.2.	VIRTUAL MACHINE(VM)	67
4.3.	GOOGLE CLOUD CLUSTER CONFIGURATION	70
4.3.1.	Single Node Configuration	70
4.3.1.1.	Java Installation	71
4.3.1.2.	Hadoop Installation	78
4.3.1.3.	Configuring Environment Variables	80

4.3.1.4. Configuring .XML Files	81
4.3.1.5. Creating Directories and Changing Ownership	85
4.3.1.6. Rebooting	85
4.3.1.7. RSA Key Generation and Authorization	87
4.3.1.8. NameNode Cleaning and Service Verification	89
4.3.1.9. Firewall Rule Settings	89
4.3.1.10. DFS Health Check-up	91
4.3.2. Multi-Node Configuration	91
4.3.2.1. Networking and SSH Syncing	93
4.3.2.2. Editing Masters and Slaves File	96
4.3.2.3. Property Modification in .XML Files	98
4.4. CHAPTER SUMMARY	100
CHAPTER 5 IMSS: SYTEM DESIGN OF INTELLIGENT META SEARCH SYSTEM	101-129
5.1. INTRODUCTION	101
5.2. SYSTEM DESIGN	101
5.2.1. Phase 1: Best Match Prediction using Machine Learning Model	102
5.2.1.1. Steps for Generating and Testing Machine Learning Model	104
5.2.2. Phase 2: Query Disambiguation and Web Page Retrieval	104
5.2.3. Phase 3: Web Page Ranking using Advanced	105

Cluster Vector Page Ranking (ACVPR)	
Algorithm	
5.2.3.1. Map() and Reduce() Methods	111
5.3. INTELLIGENT META SEARCH SYSTEM (IMSS)	114
TOOL	
5.3.1. IMSS Tool- Sign Up and Sign In	115
5.3.2. Behavior Analysis	116
5.3.3. Personalized Search Mode and Tracking	117
Recent Changes in the User Preferences	
5.3.4. Advanced Search Mode	119
5.3.5. Machine Learning Statistics	120
5.3.6. Security- Personalized Privacy Protection	121
5.3.7. Page Ranking and User Rating	122
5.3.8. Various Tables in IMSS	125
5.4. CHAPTER SUMMARY	129
CHAPTER 6 PREDICTIVE ANALYTICS	130-145
6.1. INTRODUCTION	130
6.2. PYTHON FOR IMSS TOOL DEPLOYMENT	130
6.3. LEVELS OF ANALYTICS	132
6.4. MACHINE LEARNING	134
6.4.1. Types of Machine Learning	134
6.4.1.1. Supervised Machine Learning	134
6.4.1.2. Unsupervised Machine Learning	137
6.5. RECOMMENDER SYSTEMS	139
6.6. KNOWLEDGE DISCOVERY PROCESS	140
6.7. MACHINE LEARNING FRAMEWORK	143
6.8. CHAPTER SUMMARY	145

CHAPTER 7	EXPERIMENTAL EVALUATION AND	146-203
	GRAPHICAL ANALYSIS	
7.1.	INTRODUCTION	146
7.2.	DATASETS	146
7.2.1.	Google Zeitgeist or Google Trends 2017	146
7.2.2.	Feedback.CSV	149
7.3.	MACHINE LEARNING MODELS	150
7.3.1.	Machine Learning Using Logistic Regression	151
7.3.1.1.	Steps for Generating Model	152
7.3.1.2.	Training and Testing of Model	165
7.3.2.	Result Analysis	167
7.3.2.1.	Evaluation Metrics	167
7.3.2.2.	ROC Curves	169
7.3.3.	Collaborative Filtering Model	172
7.3.3.1.	User Similarity and Web Triplet	172
7.3.3.2.	Common Subsequence Identification	173
7.4.	EVALUATION METRICS- COLLABORATIVE FILTERING MODEL	175
7.4.1.	Pearson Correlation Coefficient	176
7.4.2.	Mean Absolute Error (MAE)	179
7.4.3.	Root Mean Squared Error (RMSE)	179
7.4.4.	Comparison between MAE and RMSE	180
7.4.5.	MAE and RMSE of the Pioneered ACVPR Algorithm	180

7.4.6. Euclidean Distance	181
7.5. MACHINE LEARNING SUMMARY	181
7.6. EXPERIMENT DESIGN	186
7.6.1. Implementation	186
7.6.2. User Survey	186
7.7. RESULT ANALYSIS	189
7.7.1. Experimental Verification- Personalized Page Rank Improvement	192
7.8. COMPARATIVE ANALYSIS	195
7.8.1. Comparison with Baselines	195
7.8.2. Comparison between Various Versions of IMSS Tool	197
7.8.3. IMSS vs. Popular Recommendation Approaches	200
7.8.4. IMSS vs. Professional Metasearch Engines	201
7.9. CHAPTER SUMMARY	202
CHAPTER 8 CONCLUSION AND FUTURE SCOPE	204-207
8.1. CONCLUSION	204
8.1.1. Significance for End User	206
8.1.2. Significance for Online Businesses	206
8.1.3. Significance for Researchers and Developers	206
8.2. FUTURE SCOPE	207
SUMMARY	208
BIBLIOGRAPHY AND WEBLIOGRAPHY PUBLICATIONS	209
	232

LIST OF TABLES

Table No.	Title	Page No.
Table 1.1	Comparison of various big data deployment frameworks	14
Table 2.1	Comparison between existing and proposed personalized search systems	44
Table 3.1	Ranking Comparison of existing and proposed platforms	54
Table 7.1	Categories of search queries- Google Trends 2017 in India	147
Table 7.2	Statistics of various search parameters as calculated by the generalized linear model	156
Table 7.3	Model deviance statistics	156
Table 7.4	Statistics of various search parameters for recast generalized linear model	159
Table 7.5	Model deviance statistics for recast model	160
Table 7.6	Vif value for various search parameters	161
Table 7.7	Confusion matrix	167
Table 7.8	Evaluation metrics summary for deployed collaborative filtering model	185
Table 7.9	Baseline Comparison	195
Table 7.10	Comparison between various versions of IMSS tool	198
Table 7.11	IMSS vs. Popular Recommendation Approaches	200
Table 7.12	Comparison between IMSS tool and professional meta-search engines	202

LIST OF FIGURES

Figure No.	Title	Page No.
Fig. 1.1	HDFS framework for second generation big data systems	12
Fig. 1.2	Components of BDAS	13
Fig. 1.3	Single-node cluster setup of IMSS tool on GCP	17
Fig. 1.4	News article regarding search bias by Google	19
Fig. 1.5	News article regarding record fine on Google by EU	19
Fig. 2.1	Interface of page ranking tool	38
Fig. 2.2	Interface of IMSS-AE tool	39
Fig. 2.3	Interface of IMSS-E tool	42
Fig. 2.4	Interface of IMSS tool	43
Fig. 3.1	Overview of cloud computing	48
Fig. 3.2	Second vs. first generation HDFS	52
Fig. 3.3	Map-Reduce functionality of the IMSS tool	56
Fig. 3.4	Architecture of the IMSS tool	57
Fig. 3.5	Simplified information flow diagram in HDFS	60
Fig. 3.6	Nodes information in the Hadoop cluster of IMSS tool	61
Fig. 3.7	IMSS tool interface with a specific external IP address	61
Fig. 3.8	NameNode information of IMSS system	62
Fig. 3.9	NameNode DFS cluster information on HDFS	62
Fig. 3.10	NameNode journal status information of the IMSS tool	63
Fig. 3.11	DataNode information of the IMSS tool	63
Fig. 3.12	Hadoop scheduler metrics of IMSS tool	64

Fig. 3.13	Hadoop start-up progress for the IMSS tool	64
Fig. 4.1	Interface of GCP showing details of IMSS project	68
Fig. 4.2	Creation of a VM instance in GCP	69
Fig. 4.3	VM configurations available in GCP	69
Fig. 4.4	Secure Shell (SSH) command prompt	72
Fig. 4.5	Creation of Java8-debian.list	72
Fig. 4.6	Downloading Java packages from PPA repositories	73
Fig. 4.7	Installation of the directory manager	73
Fig. 4.8	Downloading key with ID EEA14886	74
Fig. 4.9	Updating Java packages	74
Fig. 4.10	Command to install Java	75
Fig. 4.11	Oracle JDK license agreement	75
Fig. 4.12	Oracle binary code license agreement	76
Fig. 4.13	Downloading Java in .tar format	76
Fig. 4.14	Setting Java as default	77
Fig. 4.15	Checking installed version of Java	77
Fig. 4.16	Downloading Hadoop-2.8.3	78
Fig. 4.17	Extracting Hadoop	79
Fig. 4.18	Copying Hadoop to /usr/local/hadoop	79
Fig. 4.19	Copying environment variables to .bashrc file	80
Fig. 4.20	Editing JAVA_HOME variable in hadoop-env.sh file	81
Fig. 4.21	Configuring core-site.xml	82
Fig. 4.22	Configuring hdfs-site.xml	82
Fig. 4.23	Configuring yarn-site.xml	83

Fig. 4.24	Generating mapred-site.xml	83
Fig. 4.25	Configuring mapred-site.xml	84
Fig. 4.26	Creation of NameNode and DataNode directory	84
Fig. 4.27	Root ownership to subdirectories within /usr/local/hadoop	86
Fig. 4.28	Ownership change from root to metasearch tool	86
Fig. 4.29	Command to reboot the machine	87
Fig. 4.30	RSA key generation	88
Fig. 4.31	RSA key authorization	88
Fig. 4.32	NameNode formatting	89
Fig. 4.33	Commands to start and verify services	90
Fig. 4.34	Adding firewall rules	90
Fig. 4.35	NameNode information	91
Fig. 4.36	Master node DFS	92
Fig. 4.37	NameNode and DataNode recreation	92
Fig. 4.38	External IP address information of all three instances	94
Fig. 4.39	Copying IP address information in /etc/hosts	94
Fig. 4.40	Key generation at the master instance	95
Fig. 4.41	Command to copy master key within slave instance	95
Fig. 4.42	SSH syncing between master and slave2 instance	96
Fig. 4.43	Commands to edit masters and slaves file	97
Fig. 4.44	Editing masters file	97
Fig. 4.45	Editing slaves file	98
Fig. 4.46	Property tag modification in core-site.xml	99
Fig. 4.47	Replication factor modification in hdfs-site.xml	99

Fig. 4.48	Appending additional property tag in mapred-site.xml	100
Fig. 5.1	System design of IMSS tool	102
Fig. 5.2	Phase 2 : Query disambiguation and web page retrieval	106
Fig. 5.3	Flowchart of ACVPR algorithm	113
Fig. 5.4	Sign-In interface of the IMSS tool	115
Fig. 5.5	Sign up- User registration	116
Fig. 5.6	Adding user skills	117
Fig. 5.7	Search recommendations in personalized search mode	118
Fig. 5.8	Web page ranking using IMSS tool	118
Fig. 5.9	Advanced search mode- search engine selection	120
Fig. 5.10	Page loading speed based web page ordering in advanced search mode	120
Fig. 5.11	Machine learning statistics on IMSS interface	122
Fig. 5.12	Page ranking and user rating	123
Fig. 5.13	Search results and developer console	123
Fig. 5.14	Web link selection and page attributes listing	124
Fig. 5.15	Web page visit via IMSS tool	124
Fig. 5.16	Tables in IMSS	125
Fig. 5.17	Structural description about user table	126
Fig. 5.18	Structural description about similarity metrics table	126
Fig. 5.19	Structural description about user_query table	127
Fig. 5.20	Structural description about user_rating table	127
Fig. 5.21	Structural description about profession table	128
Fig. 5.22	Structural description about question table	128
Fig. 5.23	Structural description about response table	129

Fig. 6.1	Levels of analytics	132
Fig. 6.2	Machine learning techniques	135
Fig. 6.3	Content-based filtering	140
Fig. 6.4	Collaborative filtering	141
Fig. 6.5	Knowledge discovery process	142
Fig. 7.1	Top search queries in the category of “How to” on Google Trends	148
Fig. 7.2	Top search queries in the category of “What is” on Google Trends	149
Fig. 7.3	Search suggestions by IMSS tool	152
Fig. 7.4	Generalized linear model generation for IMSS tool	155
Fig. 7.5	Feedback_Model diagnostic plot- Residuals vs. Fitted	157
Fig. 7.6	Feedback_Model diagnostic plot- Normal Q-Q	157
Fig. 7.7	Feedback_Model diagnostic plot- Scale-Location	158
Fig. 7.8	Feedback_Model diagnostic plot- Residuals vs. Leverage	158
Fig. 7.9	Feedback_Model2 diagnostic plot- Residuals vs. Fitted	163
Fig. 7.10	Feedback_Model2 diagnostic plot- Normal Q-Q	163
Fig. 7.11	Feedback_Model2 diagnostic plot- Scale-Location	164
Fig. 7.12	Feedback_Model2 diagnostic plot- Residuals vs. Leverage	164
Fig. 7.13	TPR vs. FPR for the deployed regression model	170
Fig. 7.14	Sensitivity vs. specificity for the deployed model	171
Fig. 7.15	Precision vs. recall for the deployed regression model	171
Fig. 7.16	Web triplets for a search query	173
Fig. 7.17	Evaluation metrics for collaborative filtering model	175
Fig. 7.18	Correlation analysis- User1	177
Fig. 7.19	Prediction graph- User1	178

Fig. 7.20	Correlation analysis- User2	178
Fig. 7.21	Best match – User similarity graph	179
Fig. 7.22	MAE, RMSE calculation for the search query “IMSS”	180
Fig. 7.23	Machine learning summary for eight users	182
Fig. 7.24	Machine learning summary for seven users	183
Fig. 7.25	Machine learning summary for six users	183
Fig. 7.26	Machine learning summary for five users	184
Fig. 7.27	Machine learning summary for four users	184
Fig. 7.28	Machine learning summary for three users	185
Fig. 7.29	Precision comparison IMSS and Dogpile Response Time	190
Fig. 7.30	Precision comparison between IMSS and Dogpile – Page Freshness	191
Fig. 7.31	Precision comparison between IMSS and Dogpile – Personalized Relevancy	191
Fig. 7.32	Precision comparison of IMSS with Google, Bing and Qwant	191
Fig. 7.33	User1 search query – apple iPhone	192
Fig. 7.34	User1- high rating to web link at rank # 7	193
Fig. 7.35	Results presented to User2	193
Fig. 7.36	Search suggestion for an incomplete query	194
Fig. 7.37	Improvement of link rank to rank #4	195

LIST OF ABBREVIATIONS

ACVPR: Advanced Cluster Vector Page Ranking Algorithm

BDAS: Berkley Data Analysis Stack

CMS: Coexistence Match Similarity

CPS: Conditional Probability Similarity

CRV: Content Relevancy Vector

CSS: Cascading Style Sheets

FRV: Feedback Relevancy Vector

GCP: Google Cloud Platform

HDFS: Hadoop Distributed File System

HTML: Hyper Text Markup Language

IMSS: Intelligent Meta Search System

JT: Job Tracker

PHP: PHP Preprocessor

PV: Privacy Vector

ROC: Receiver Operating Characteristic

RTV: Reply Time Vector

SRV: Similarity Relevancy Vector

SS: Subsequence Similarity

TT: Task Tracker

VM: Virtual Machine

CHAPTER 1

INTRODUCTION

1.1. INTRODUCTION

Information seeking is one of the most natural needs of human behavior and how it is retrieved is dependent upon many factors varying from personal choices to technical requirements (Malhotra, 2014). Moreover, in the present generation of big data, the information searching process is reformed a lot because of the massive growth in web resources. The modern generation user prefers to search for specific information via search engines because of the easy availability of Internet Service Providers (ISP). The extreme competition among ISPs in a country like India leads to affordable Internet rental and hence unexpected growth in the number of Internet users in a small duration of the last two years is observed (Malhotra and Rishi, 2018b). However, most of the popular search engines cannot consider user interaction with the web for page ranking. Traditional web mining algorithms are not capable enough to utilize the relevance implied by user surfing patterns to improve the ranking of web pages (Malhotra, 2014). Therefore, searching and listing a relevant website on the top to satisfy the personalized requirements of the user quickly is not easy as web users are mostly reliant on the generic search engines like Bing, Yahoo, Google to choose a most relevant website among top three to five links on the first page (Malhotra and Rishi, 2018b). However, as discussed by Gomez-Nieto et al. (2014), when different users input the same search query, most of the popular search engines fetch the same links in the result. The modern search engines return the search result without considering the personalized preferences of the user.

Moreover, as discussed by Malhotra and Verma (2013), if the search query is partially incomplete or is ambiguous, then most of the modern search engine tends to return the result by interpreting all possible meanings of the query. For example, if we consider a partial search query "Jurassic World" by two different web users on the Google Search engine in June 2018. The search engine shows top web links of the latest released movie on "Jurassic World: Fallen Kingdom." It may be very much possible that one of the web users is interested in web pages to read reviews or to buy movie tickets for the latest movie. However, the same is not necessarily applicable for another user, who might be interested in visiting a themed water park with "Jurassic World" name. This problem may be resolved by an intelligent metasearch system designed and developed using logistic regression or collaborative filtering based machine learning model and Hadoop 2 based advanced big data analytics platform. The personalized search system will fetch the query from a user and will first modify the search query based on user preferences as mentioned in his or her profile. The personalized preferences of a user can also be retrieved from his or her short-term browsing history. For instance, a user usually booking tickets for themed parks will be shown web pages related to themed water parks consisting of keywords used in his or her search query, i.e., Jurassic World on the top of the search result and not the links of a movie as discussed above (Malhotra and Rishi, 2018b). Recently, web search personalization has attracted researchers to deal with the problem of ambiguous page ranking (Pare, S., & Vasgi, B., 2014). The personalized search allows the user to have easy and accurate information access (www.ijmer.com [165]). With the constant development of personalization focused researches, the intelligent search technology with the feature of adaptability and learning to satisfy the personalized search needs of the modern user is also transiting rapidly from the algorithmic stage to practical application stage. At present, personalized page ranking supported by machine learning and big data analytics has become the key technology and core idea of Internet information retrieval (Malhotra, 2014).

The essential and alternative aspect of personalized page ranking is to devise various means to handle massive on stream accumulated data on the web efficiently. This extensive data is popularly known as 'Big Data' with more insistence on the Volume of data besides other V's to characterize the data, i.e., Velocity, Variety, Value, and Veracity (Malhotra and Rishi, 2018b). Big data is defined as an extensive collection of datasets and sources that are beyond the capabilities of traditional search and page ranking systems to process effectively and efficiently. The detailed ranking comparison of various big data analytics and cloud deployment platforms in the present research work justify the choice of Hadoop 2 as the best analytics platform for the deployment of the intelligent metasearch system. The Hadoop 2 advances the capabilities of Hadoop 1 by introducing two new modules, i.e., Yarn and HDFS federation. The Yarn module enables segregation of resource management responsibilities from processing engines. The HDFS federation allows the creation of multiple name nodes as compared to a single node in Hadoop 1. These advancements help in building a more reliable and robust system architecture for efficient big data analytics.

The machine learning model based on logistic regression or collaborative filtering may be developed using the R- statistical tool or advanced machine learning using Python. The machine learning model assists in predicting the suitable extension of an incomplete or ambiguous search query and hence to determine the suitability of a web page listed in the output of a search tool to satisfy the personalized search needs of a user. The model will learn the most suitable page ranking order for a specific web user concerning various parameters such as page loading speed, response time, security while browsing the page and personalized relevancy. The scientific evaluation concerning the calculation of the confusion matrix, specificity, sensitivity, etc. easily verifies the fitment of the model for current research work (Malhotra and Rishi, 2018b).

The overall objective of present research work is to merge the machine learning with big data analytics for implementation of the new personalized page search and ranking

algorithm, i.e., Advanced Cluster Vector Page Ranking (ACVPR) algorithm. The pioneered algorithm is deployed in the form of Intelligent Meta Search System (IMSS) tool to assist the end user in listing most relevant web pages in the ranking order of his or her preferences.

1.2. WEB INFORMATION RETRIEVAL

Web information retrieval stands for searching and finding unstructured web documents to satisfy individual information needs from an extensive collection of web pages on the World Wide Web (WWW). In web search, a retrieval system is required to search for relevant documents among billions of web documents available on the WWW. The users of traditional information retrieval systems were technically professional in phrasing search queries. However, within modern age, it's not necessary that only professional user will use the retrieval system, studies show that an average number of keywords in a search query varies in between 2-5 words with seldom use of operators. Moreover, the search engines are more often used by ordinary people including illiterates rather than just IT engineers. The expected characteristics of a modern search tool include:

- Personalized information retrieval rather than generic information retrieval to meet the specific needs of an individual issuing the query, i.e., the focus of search should be on personalized search precision rather than recall.
- User-friendly interface to quickly search for relevant tab or feature.

1.2.1. Search Queries

There exist three broad categories of search queries. However, some of the queries may overlap within two types while few queries may lie outside these categories. The general categories are as follows:

- Navigational Search Query: These queries represent that user is interested in browsing a specific web page, for instance, a search query like “*University of*

Kota” means that user wants to see the home page of the University of Kota as its first output link rather than paid or advertised links of other universities.

- Informational Search Query: These queries require web pages on the top with detailed informational content about the keywords within the search query, For instance, a search query like “*Hepatitis B*” needs web pages on the top having explicit informational material about the disease.
- Transactional Search Query: These queries represent that user wishes to perform an online transaction such as E-Commerce queries, for instance, a search query like “*IRCTC online train ticket purchase*” represent the fact that user wishes to book a train ticket on the IRCTC website.

1.3. RESEARCH PROBLEM

The availability of web data on WWW is quite huge. Such a huge repository of data may be termed as *Big Data*. In this scenario, it becomes quite challenging for a user to find relevant information from the Internet. One of the approaches is to use a popular search engine. However, none of the search engines can completely solve the problem of complete relevant information retrieval as each search engine can index only a subset of information available on WWW due to gigantic size and dynamic nature of the web. There are various limitations of traditional search engines such as low precision, low recall, irrelevant search results (Malhotra and Rishi, 2017). Moreover, a typical search engine returns the same result corresponding to the same query, regardless of the user who submitted the query. This generic ranking is not suitable for users with different information needs. Let us take an example, a user searching for “Hotel Taj in Mumbai” on Google in November 2015 found Hindustan Times Article explaining the Terrorist attack happened in Hotel Taj. However user wanted to search for booking information in Hotel Taj, and hence page retrieved was not relevant for the user.

Some of the modern search engines provide personalized search to address the problem as mentioned above. However, they fail to address changing user's requirement over time. Even with popular search engines users are often required to alter their query, many times to retrieve relevant documents (Malhotra et al., 2017). Moreover, the credibility of information extracted through search engines is declining as evident through news articles highlighting record fine to a popular search engine, Google for *Search Bias* by European Union and Competition Commission of India. These news articles reflect the requirement of new platforms based on improved and unbiased algorithms for reliable web search as an end user usually believe that top few links in the output of a popular search engine are most reliable links to satisfy the informational requirement or to buy a service or product. The news articles highlighting search bias by Google is discussed in detail within section 1.7 and is also shown through screenshots as available on *Thomson Reuter's and BBC website* in Fig. 1.4 and Fig. 1.5.

We may use a metasearch engine to overcome limited indexing and hence low recall problem of generic search engines. A metasearch engine is built on the top of the number of search engines. Whenever a query is submitted to a metasearch engine, it will search the query on all of its backend search engines followed by processing and merging of results obtained from each of the search engines to display more satisfactory search result to the user due to enhanced recall. However, a conventional metasearch engine has its associated challenges; first of all, the number of web documents returned in response to a user query by multiple backend search engines is quite voluminous. Secondly, if the search query is incomplete or ambiguous, the results suffer from low precision and become vaguer as conventional search engines try to retrieve documents corresponding to all possible meanings of the query. Thirdly, the traditional data mining and page ranking algorithms used by traditional search engines are quite a time consuming and resource intensive to mine useful ranking patterns from such voluminous search data obtained from multiple backend search engines. Hence, merging and precisely ranking such a

massive amount of data requires a lot of efforts. The modern search systems require novel page ranking algorithms well supported by next-generation big data analytics and effective personalization capabilities. Web page searching and ranking systems using conventional techniques have many issues like (Malhotra and Rishi, 2017):

- Traditional web search and page ranking approaches do not seem to focus on application and infrastructure scalability, consistency, component recovery, data recoverability, and partial failure support, ability to respond in real time as required by next-generation metasearch systems or general search engines to search in continuously growing *Big Data* environment.
- Most of the popular search systems include a syntactic search of resources which means they implement a matching process concerning frequency count, proximity, etc. between a user search query and candidate web page. This syntactic matching lack semantics, as a result, the product queries which can be interpreted in various contexts are likely to produce wrong results, and the user usually ends up with thousands or even more links and sometimes not even a single link in the output.

1.4. OBJECTIVES OF RESEARCH

The existing personalized search and page ranking systems are unable to mine useful patterns especially from *Big Data* stored in modern search engine's vast and dynamic databases. The overall objective of the proposed research work is to unite the *Personalized Metasearch* process with the benefits of *Next Generation Big Data Analytics*.

Keeping this in mind the specific objectives of the present research work are as follows:

1. To underline the flaws in existing mining techniques to extract useful page ranking patterns from *Big Databases* of search engines.

2. To make a comparative analysis of various traditional personalized search systems along with the review of different cloud-based big data deployment frameworks such as Hadoop Distributed File System (HDFS) etc. to choose the best deployment framework for the proposed personalized search system
3. To develop a model for a novel Intelligent Meta Search System (IMSS) that does not mandate user efforts in the form of explicit ratings or feedback for extracting personalized search information.
4. To develop an intelligent page search and personalized ranking algorithm, i.e., Advanced Cluster Vector Page Ranking (ACVPR) algorithm.
5. To evaluate the effectiveness and efficiency of the proposed approach using scientific and mathematical methods such as the Pearson correlation coefficient and other relevant metrics.

1.4.1. Objectives Accomplishment

The objectives are addressed in detail within various chapters of the thesis, and the same is also published in our research publications (Malhotra & Rishi, 2018a), (Malhotra & Rishi, 2018b), (Malhotra & Rishi, 2017), (Malhotra & Rishi, 2016). The objective one is addressed within chapter 2, literature review, where category specific literature about conventional mining based search systems is discussed and their shortcomings to extract useful page ranking patterns from big databases is discussed in detail. Objective two is discussed in chapter 1 and chapter 2. The detailed ranking comparison between various big data deployment platforms is introduced within section 1.6 followed by the detailed discussion within chapter 3. The objective three about the Intelligent Metasearch System, i.e., IMSS tool to assist the end user in personalized web search is addressed in chapter 5 and chapter 7. The IMSS tool does not mandate user efforts in the form of explicit ratings or feedback for extracting personalized search information in the form of disambiguated or expanded personalized search query suggestions especially while using *Search*

Recommendation feature. The objective four regarding the development of an intelligent page search algorithm, i.e., Advanced Cluster Vector Page Ranking (ACVPR) algorithm is addressed within chapter 5. The objective five about detailed experimental analysis of the proposed approach through scientific and mathematical method in the form of various evaluation metrics like Pearson Correlation Coefficient, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) for collaborative model and metrics like Accuracy, Specificity, Sensitivity, Precision and Recall for Regression based model are discussed in detail within chapter 7.

1.5. CONTRIBUTION FROM THE STUDY

To the best of our knowledge as procured from the literature, this study is the first formal attempt to design, and development of an effective and efficient personalized metasearch system using next-generation big data analytics. Various vital contributions of the current research work may be summarized as follows:

- The current research work lead to the design and development of machine learning enabled personalized web search algorithm, that is, *Advanced Cluster Vector Page Ranking Algorithm (ACVPR)* and its deployment in the form of a future-ready tool i.e., *Intelligent Meta Search System(IMSS)* to assist the end users in carrying out personalized web search and page ranking on the WWW
- The implemented metasearch system addresses the limitations of traditional mining approaches to extract useful web search and page ranking patterns from databases of search engines. The deployed system possess features like scalability, partial failure support, etc. through its deployment using Hadoop 2 on Google cloud platform
- The implemented system through its machine learning capabilities can easily determine user search preferences and can assist the end user in easily framing

nonambiguous search queries to satisfy his personalized search requirements on the web

1.6. COMPARISON OF PLATFORMS FOR BIG DATA ANALYTICS

The massive on stream accumulated data on the web is popularly known as '*Big Data*' with more insistence on the Volume of data besides other V's to characterize the data, i.e., Velocity, Variety, Value, and Veracity. *Big Data* is defined as an extensive collection of datasets and sources that are beyond the capabilities of traditional search and page ranking systems to process effectively and efficiently (Malhotra and Rishi, 2018b).

To choose a suitable deployment framework for a personalized web search application, firstly it is required to consider and compare various deployment platforms by multiple factors such as capabilities for scaling, fault tolerance, real-time processing, iterative execution, etc. (Malhotra and Rishi, 2018a). Here, present research work discusses various existing deployment paradigms in section 1.6.1, section 1.6.2 and 1.6.3, highlighting some of the inherent features and components of modern platforms like Hadoop Distributed File System (HDFS) and Berkley Data Analysis Stack (BDAS) useful for deployment of a novel, next-generation personalized search system followed by their ranking comparison with other existing frameworks in section 1.6.4.

1.6.1. Types of Existing Deployment Paradigms

Various existing deployment paradigms are explained as follows (Malhotra et al., 2017):

- In the first type, clusters use a blob store as primary storage such as S3 or Azure blob store. Here clusters are transient and exist till the duration of workflow execution. The important key is the blob store which is a source and destination of the workflow. Here virtual machines are thought of as task execution containers
- The second type uses the first generation Hadoop Distributed File System (HDFS) as primary storage. Here virtual machines are persistent and can perform

execution as well as store data. This category may even use blob stores for periodic backups and to provide data to HDFS. This type of deployment paradigm is useful for workloads like Ad Hoc batch and Ad Hoc interactive based workloads due to the requirement of persistent clusters.

1.6.2. Hadoop 2: Recommended Big Data Deployment Framework

Hadoop Distributed File System (HDFS) is now adapted as a long-term store from which applications read their initial data and write their final results. However one of the significant drawbacks of HDFS lies in running iterative algorithms. Map function needs to read data at the start of each & every iteration and to write back the data to the disk at the end of the iteration. This repeated access to disk in reading and writing is responsible for degradation of performance (Malhotra et al., 2017). The Hadoop 2 advances the capabilities of Hadoop 1 by introducing two new modules, i.e., Yarn and HDFS federation. The Yarn module enables segregation of resource management responsibilities from processing engines. The HDFS federation allows the creation of multiple name nodes as compared to a single node in Hadoop 1. These advancements help in building a more reliable and robust system architecture for efficient big data analytics on the web (Malhotra and Rishi, 2018b).

With the changes in environmental trends and technological shifts, second-generation big data systems not just require scalability, partial failure support, etc. but also need to support multiple analytic methods on varied data types, as well as the ability to respond in near real time as shown in Fig. 1.1, depicting big data evolution (Malhotra et al., 2017).

There are two significant trends of the second generation big data systems that are responsible for choosing HDFS as a preferable deployment framework in the current research work primarily to implement effective and efficient web search personalization (Malhotra et al., 2017)

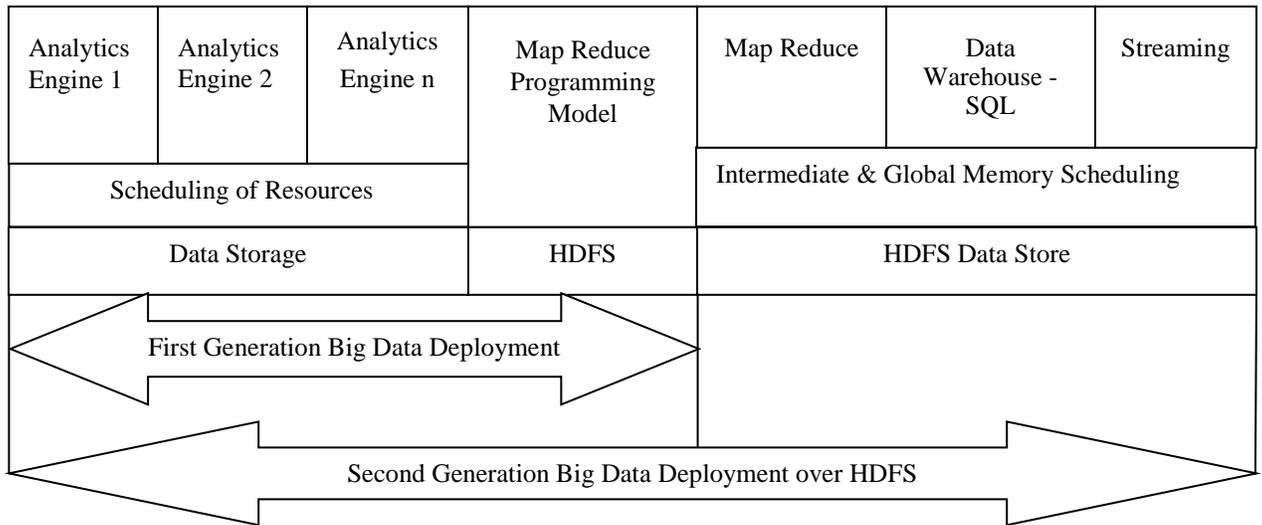


Fig. 1.1 HDFS framework for the second generation big data systems

- There is a rapid growth in the network bandwidth as compared to hard drive bandwidth over the period. Hence, HDFS is chosen to deploy the metasearch tool, i.e., IMSS to easily handle the dynamic load in the form of web links retrieved from three giant background search engines
- Support of *In-Memory* computation model by HDFS allows intermediate results to be kept in the memory and hence reduces the overhead of iterative analytics for real-time response to the end user. Consequently, the end user will not experience any performance lag due to personalization of search queries or metasearch tool implementation as discussed in the current research work

1.6.3. BDAS vs. Hadoop 2

BDAS, i.e., Berkley Data Analysis Stack is an alternative platform available for proposed research work. BDAS is based on spark and HDFS based data processing stack to overcome the limitations of generic Map-Reduce platform while running the iterative processes. Some of the essential components of BDAS (Malhotra et al., 2017) are discussed and shown in Fig. 1.2.

Tachyon: It is the lowest level component of BDAS and is based on HDFS. Map-Reduce programs are compatible with Tachyon and can run without alterations. The advantage of Tachyon over HDFS is minimized disk access by caching the files that are frequently read and enables data to be read at memory speed.

Spark Streaming	Shark SQLShark SQL	MLBASE
Spark		
Mesos		
Tachyon		

Fig. 1.2 Components of BDAS

Mesos: It serves the role of cluster manager that provides efficient resource allocation across distributed frameworks. It also supports HDFS and helps in improving horizontal scalability.

Spark: It is the substitute for Map-Reduce component of HDFS and allows memory caching to overcome the problem of iterative tasks processing.

Even though there are many advantages of the BDAS, however, we have given here preference to Hadoop 2 as a deployment framework for ACVPR algorithm and IMSS tool implementation. The very first reason is that the topmost layers of BDAS consist of many applications that are still in the early stages of development. Secondly, Hadoop 2 is one of the widely used distributed deployment frameworks for big data due to easy availability of required infrastructure and compatible tools.

1.6.4 Ranking Comparison of Existing and Prescribed Platforms

Here Table 1.1 shows the ranking comparison of various popular big data deployment frameworks concerning multiple characteristics such as scaling, fault tolerance, etc. Here Rank-1 shows the best option and Rank-5 for worst choice among all of the listed platforms. It may be noted that this ranking table provides a general idea regarding the strengths and weakness of various platforms and it mainly depends on the specific application or purpose. In general, for big data applications, there is a tradeoff between scaling and real-time processing capabilities. For example, as in current research work to deploy web search applications, indexing process requires a highly scalable platform to handle billions of web pages, so HDFS and Spark are the optimal choices for web search applications and hence these are preferred.

Table 1.1 Comparison of various big data deployment frameworks

Platform	Scaling Rank (Type)	Fault Tolerance Rank	Real-Time Processing Rank	Iterative Tasks Rank
HDFS	1 (Horizontal)	1	4	4
SPARK	1 (Horizontal)	1	4	3
PEER TO PEER	1 (Horizontal)	5	5	4
HPC CLUSTERS	3 (Vertical)	2	3	2
MULTICORE	4 (Vertical)	2	3	2
GPU	4 (Vertical)	2	1	2
FPGA	5 (Vertical)	2	1	2

However, HDFS based Hadoop 2 is preferred over Spark based BDAS as deployment frameworks for current research work due to easy availability of infrastructure and deployment tools as discussed in detail within section 1.6.3 (Malhotra and Rishi, 2018a).

1.6.5. Google Cloud Platform for Big Data Analytics

The big data analytics is vital for a metasearch tool like the IMSS tool to generate a most relevant page ranking order to be shown to the user to best suit the personalized requirements of a search tool user. The big data in the form of returned links by various background search engines can be easily analyzed using Hadoop 2 and Map-Reduce based analytics when deployed on a cloud platform such as Google cloud platform. The Google Cloud Platform (GCP) is used for Hadoop based multi-node cluster setup required to implement the intelligent metasearch system application, i.e., IMSS using ACVPR algorithm. The Google cloud platform is an ideal platform for exploring various cloud services. The compute engine module allows us to create and use Virtual Machines (VM) which are virtual copies of OS servers like Linux server, Window server, etc. This module lets the developer choose VMs with small to large configuration regarding CPU cores, memory, and OS image to best suit the metasearch project requirements like that of IMSS tool (Malhotra and Rishi, 2018b).

1.6.5.1. Cloud Cluster Configuration of Meta Search Tool

There are two possible ways to set up a cluster on Google cloud:

- Single Node Configuration
- Multi-Node Configuration

The step by step configuration for a metasearch application is discussed below. The node configuration will first create a VM instance known as a master instance. The master instance will serve the purpose of both the name node and data node. However, slave instances will act as data nodes to process the links returned by background search

engines. The common steps for master and slave instance setup for IMSS tool are listed as follows:

- Java Installation
- Hadoop Installation
- Configuring Environment Variables
- Configuring XML Files
- Creating Directories and Changing Ownership
- Rebooting
- RSA Key Generation and Authorization
- Name Node Cleaning and Service Verification
- Firewall Rules Settings
- DFS Health Checkup

The above steps are required for a single node setup. However, the additional steps required for a multi-node cluster are listed as follows:

- Networking and SSH Syncing
- Editing Masters and Slaves File
- Property Modification in XML Files

The name node and data node directories are first required to be deleted and then recreated. This recreation is required to avoid any possibility for junk data within these directories and hence to facilitate multi-node configuration and can be accomplished through the execution of commands. The master instance is responsible for directing all slave instances with web page links as returned by the number of background search engines to the metasearch tool. The master instance is also responsible for HDFS book-keeping tasks and monitoring the overall health of the cluster. The master instance is required to remain active all the time as failure or pause of the master instance will make

the entire cluster inaccessible and hence the breakdown of the IMSS application. The number of duplicate copies kept on different slave instances is determined by the replication factor. The master instance will keep track of all slave instances via a heartbeat connection. A daemon called Job Tracker (JT) on the master instance is responsible for accepting expanded search query requests from the client and allocating those tasks to Task Trackers (TT) residing on different slave instances. The JT can also reallocate the job to a new TT on the replica of a slave instance in case of a failure of a node to ensure that slave failure does not lead to the cluster breakdown or job failure. The TT is active on slave instance, unlike JT which is active on a master instance. TT and JT both are known as computing nodes. Besides, TT is also required to update its existing status to JT via heartbeat connection. The responsibility of a TT here in the cluster is to accept the search query request from a JT and to collect the page links returned by a corresponding search engine and to sort those links according to personalized preferences of the user. However, to reduce the communication time between master and slave processes, IMSS application is deployed as a single node cluster setup. The IMSS instance used as master cum slave instance to deploy single node cluster setup for metasearch application in the current research work is shown in figure 1.3.

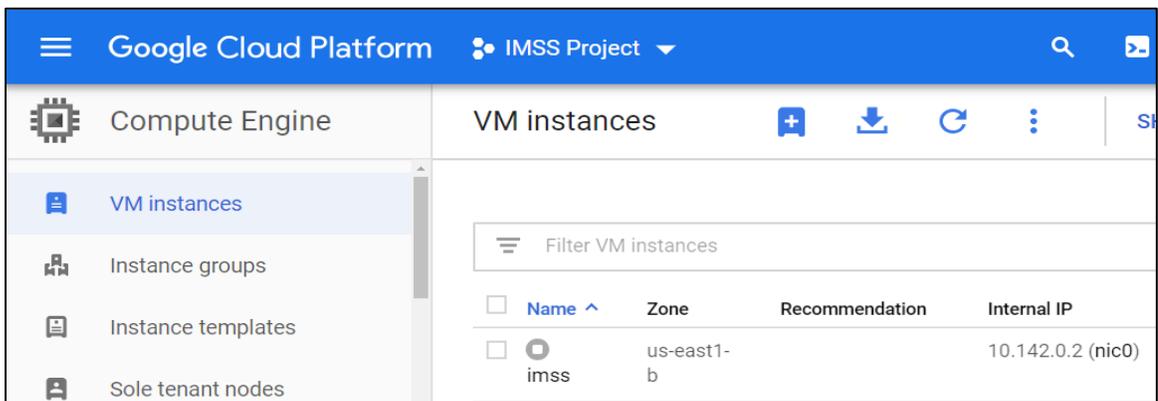


Fig. 1.3 Single-node cluster setup of IMSS tool on GCP

1.7. MOTIVATION

The data or information is increasing on the massive scale on the WWW. This vast amount of data on the web is known as big data. It is usually quite tedious for a web user to quickly search for significant information from WWW. Most of the web users tend to use a search directory or a search engine. However, there are several restrictions possessed by conventional search engines such as incomplete indexing, insufficient recall or precision, a non-personalized order of web links in the output of search engine results (Malhotra and Rishi, 2018b). A typical search engine does not take into account the context or interests to determine the personalized preferences of the web user. For instance, a user searching for a query “apple” after searching for “Galaxy phone” is likely to see web links of iPhone on the top of search results and not to that of a fruit. However, most of the search engines tend to show the same order of web links to all the users searching for the same query without considering such contextual or personalized relevance. There are few search tools or engines which provide an option for a personalized web search. However, they cannot keep track of changing user needs. Moreover, most of the popular search engines are biased and tend to show the paid links at the top of their search results irrespective of their relevance concerning the user's query. For instance, Indian antitrust watchdog imposes a fine of 21.17 million USD on Google for the search bias in February 2018 as highlighted by news on *Thomson Reuter's* website and is also shown in Fig. 1.4. The Competition Commission of India (CCI) found Google indulged in abusing its dominant position and using search bias to harm web user and other competitors. Earlier European Union also imposed a fine of record billion USD on Google for biased search output to devalue rival offerings as highlighted by BBC news website and is shown in Fig. 1.5. (Malhotra and Rishi, 2018b).

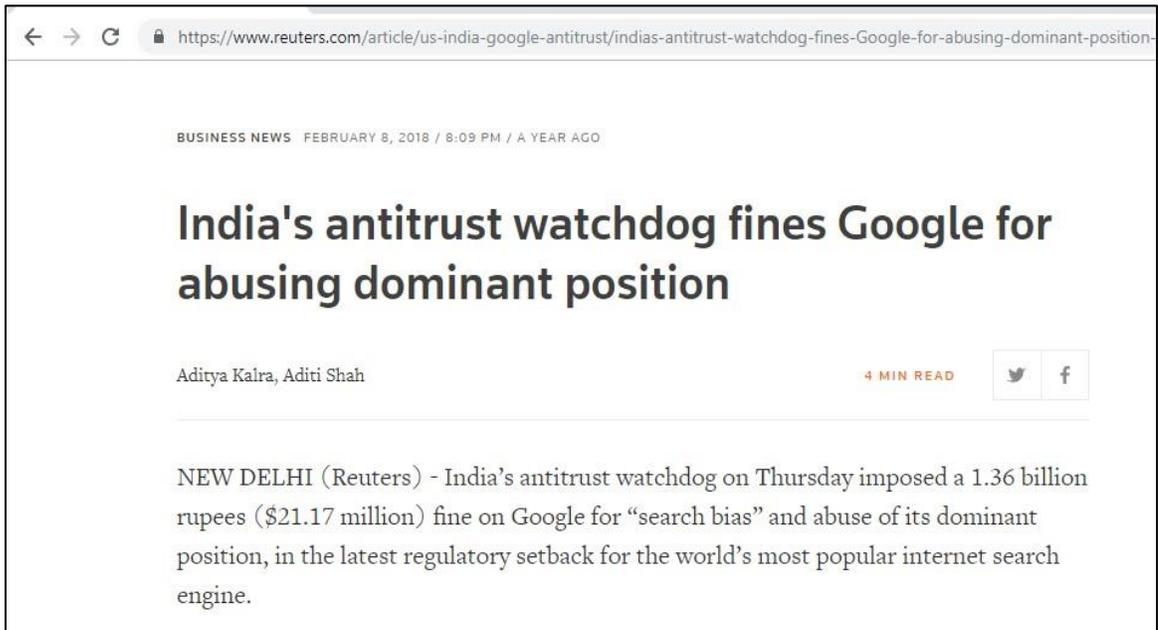


Fig. 1.4 News article regarding search bias by Google (www.reuters.com [132])

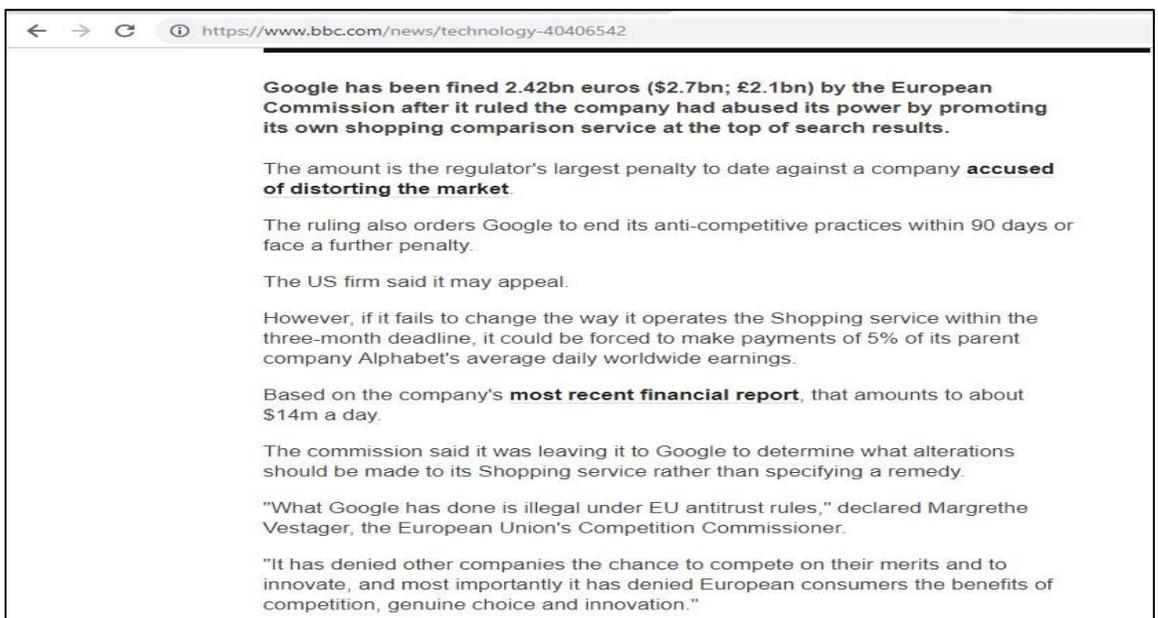


Fig. 1.5 News article regarding record fine on Google by EU (www.bbc.com/news [163])

Moreover, conventional personalized systems discussed in the literature also possess several shortcomings. For instance, as discussed in detail within section 2.1.1 of chapter 2, personalized search systems based on hyperlinks take more time to calculate relevance, and they do not satisfy user's contextual information needs because bookmarks, browsing history, etc. are not taken into consideration. The search systems based on content personalization, discussed in section 2.1.2. increases load on the user as they require the user to register his or her personal preferences, and also such systems can't adapt automatically with a change in user needs and users are expected to change recorded preferences whenever their interest changes. Recommender systems as discussed in section 2.1.3. require user's ratings to give good recommendations. Systems based on contextual knowledge as mentioned in section 2.1.4. also, need having user's explicit data for correct page ranking. However, many web users are not taking interest to spare time for giving exact ratings and precise information required to support such search systems. Systems based on Intelligent Technologies as discussed in section 2.1.5. lacks the personalization concept and fail to adapt to the changing needs of the user. Moreover, they can't satisfy the requirements of the second generation of big data systems such as an ability to respond in real time and support for multiple analytic engines (Malhotra and Rishi, 2017).

The above-stated reasons indicate the need for a novel and machine learning enabled personalized metasearch approach capable of handling big data analytics. The metasearch tool can partially overcome the limited indexing problem by using multiple search engines in the background. Moreover, the capability to employ next-generation big data analytics can lead to secure processing of a massive number of web links as returned by popular background search engines to easily satisfy the personalized preferences of the web user (Malhotra et al., 2017).

Unlike the previous research studies discussed in the literature review chapter (Malhotra and Rishi, 2018b), the present research work results into design and development of

Advanced Cluster Vector Page Ranking (ACVPR) in the form of an Intelligent Meta Search System (IMSS) and is novel in the following ways:

(i) ACVPR algorithm can easily predict the user preferences with accuracy by employing logistic regression or collaborative filtering based machine learning model

(ii) IMSS tool possess characteristics of next-generation big data analytics tool and is capable of performing elastic scaling, infrastructure offloading, real-time handling of search load spikes and resource management with high reliability.

(iii) IMSS tool provides both personalized and advanced search mode to assist the end user to use all features of a metasearch environment.

1.8. ORGANIZATION OF THE THESIS

This thesis is classified into eight significant chapters, which are organized as follows:

Chapter 1 presents the overview and motivation to pursue present research work about personalized information retrieval from the web. The ranking comparison between various popular platforms for big data analytics is also briefly discussed. This comparison is followed by an introduction to set up a single node and multi-node cloud cluster setup. This cluster set up discussion is further augmented by the detailed analysis of the research problem, objectives of research, expected contribution, motivation, and organization of thesis.

Chapter 2 discusses the category specific literature review broadly about four types of conventional search systems based on hyperlinks, content personalization, contextual knowledge, and recommender systems. This chapter is mainly focused on the first two objectives of this research work. The discussion and tabular comparison of different web personalization based search systems and IMSS tool are carried out by various

capabilities or features like metasearch, intelligent technology, big data analytics and various evaluation metrics used.

Chapter 3 discusses various available cloud computing platforms to design and deploy the metasearch application. This chapter carries out a detailed ranking comparison between different platforms like HDFS, Spark, etc. The tabular ranking comparison helps in identifying HDFS as the best cloud platform to implement IMSS tool. The detailed system architecture of the implemented IMSS tool is also discussed in this chapter. The system architecture is followed by various screenshots of the Hadoop cluster information for IMSS tool. The cluster information is supported by the detailed discussion of the advantages of choosing the second generation HDFS and Google cloud-based deployment for present research work.

Chapter 4 discusses features provided by the Google cloud platform. These features can be used to set up a single node and multi-node cluster for big data analytics. The multi-node cluster can be used to implement Hadoop 2 and Map-Reduce environment to perform personalized page ranking through implementation and deployment of a metasearch engine.

Chapter 5 discusses the research methodology and three phases of system design to implement intelligent meta-search tool. The website re-ranking process using a new ACVPR algorithm and its detailed flowchart is described in detail.

Moreover, map and reduce methods for keywords frequency calculation to determine CRV is also elaborated. The interface of the IMSS tool and database design deployed to assess the effectiveness of the ACVPR algorithm is also discussed.

Chapter 6 discusses the basics of machine learning deployed within the present research work. The detailed discussion about various forms of analytics, recommender systems, types of machine learning, the knowledge discovery process is included in this chapter.

This chapter further discusses multiple features of Python language to support machine learning framework implemented within the ACVPR algorithm and IMSS tool.

Chapter 7 discusses in detail regarding implementation and calculation of various evaluation metrics for logistic regression and collaborative filtering based model for the deployment of machine learning capabilities of IMSS tool. The experimental design, user survey, and verification of query expansion and personalized web page rank improvement are also demonstrated through screenshots of the live tool. The comparison of the deployed approach with baselines establishes significant improvement regarding capabilities required by modern web search personalization approach. Moreover, comparison of pioneered IMSS tool with professional metasearch engines demonstrates more powerful and user-friendly features of the IMSS tool.

Chapter 8 highlights various important conclusions and future work. The conclusion section highlighted the significance of present research work for multiple users i.e.

- Significance for the end user
- Significance for online businesses
- Significance for researchers and developers

1.9. CHAPTER SUMMARY

This chapter presents the overview and motivation to pursue current research work. The ranking comparison between various popular platforms for big data analytics is also briefly discussed. This comparison is followed by the introduction to set up a single node and multi-node cloud cluster setup. This setup discussion is further augmented by a detailed analysis of the research problem, expected contribution, motivation, and organization of the thesis. This chapter also gives a brief introduction to addressing the various objectives of the proposed research work.

CHAPTER 2

LITERATURE REVIEW

2.1. INTRODUCTION

Due to explosive growth in websites on WWW, researchers proposed many search systems from time to time to facilitate the end user to search for the relevant site easily. The literature review in this chapter will cover the journey of search systems from hyperlink based search and page ranking systems to intelligent and advanced technologies like big data, semantic web; machine learning based personalized search and page ranking systems. This research work carries out a detailed category specific literature review to quickly find the research gap between various studies from time to time within literature. This chapter addresses the first two objectives of this research work, i.e., (i) To underline the flaws in existing mining techniques to extract useful page ranking patterns from big data flooded indexed repositories of search engines. (ii) To make a comparative analysis of various traditional personalized search systems.

The conventional search systems based on personalized web search discussed in the literature are as follows (Malhotra and Rishi, 2018a, b)

2.1.1. Review of Search Systems based on Hyperlinks

2.1.2. Review of Search Systems based on Content Personalization

2.1.3. Review of Search Systems based on Contextual Knowledge

2.1.4. Review of Search Systems based on Recommendation

2.1.1. Review of Search Systems based on Hyperlinks

The hyperlink based personalized search systems can well assist the web user while searching for information resource on the web. It is usually assumed that customers who gave similar explicit or implicit feedback have same tastes while searching on the Internet and hence various web pages are recommended to the user based on response to the webpage by a previous user having a similar profile as the current user. Aoki et al., (2015) explained the system architecture of a personalized search system, i.e., the web index system which uses web index files that contain a pair of keywords and corresponding URL. The proposed method can perform *attach* operation to associate keywords to the hyperlinks. The attach operation consists of the following sub-steps (i) clicking bookmark link (ii) requesting server (iii) lexicographic matching (iv) hyperlink generation (v) receiving a response and displaying page (Malhotra & Rishi, 2018b). In lexicographic matching, if multiple hyperlinks are associated with a keyword then proposed system will list all hyperlinks in a pop-up window. However, only one of the listed URL is of interest to the user. This problem was addressed by an automatic recommendation to the most relevant URL. The $S(I, J) = \text{Cos } \theta_{i,j}$ between a viewed web page and candidate pages is calculated for recommendation to the user. The candidate pages with a significant value of $S(i, j)$ are referred to the user. However, the primary limitation of the proposed WIX system is more time required for relevancy computation. Alam and Sadaf (2014) suggested that the modern search engines retrieve a massive number of irrelevant and unmanageable web pages in response to a query especially when the query is incomplete or erroneous as most of the search engines tend to return result corresponding to all possible meanings of a user's query. However, clustering may be used to summarize a large number of documents in search engine output. The proper labeling of each cluster is necessary to define the content of the cluster and to assist the user in selecting a relevant cluster. They applied a heuristic search method to find all the pages of the cluster. The title of a document is an appropriate source to determine the

content of the document. The label of each cluster is defined by the keywords used in the title of documents sharing hyperlinks. They took top 100 hits by searching Jaguar query on Google. They applied the Apriori algorithm with support=2 for finding frequent two itemsets and found labels for cars, sports, and animals. The primary advantage of the proposed method is that a lot of computation time could be saved as only those documents sharing hyperlinks are considered for the labeling process. However, the proposed method could be improved by considering text within meta tags for labeling process (Malhotra & Rishi, 2018b).

Brin and Page (2004) discussed the prototype of a large scale search engine, Google mainly based on the structure used in hyperlinks. The goal of Google is to address the problem of scalability and relevance of web link. They highlighted that the output of keyword-based automated search engines is not up to the mark to satisfy the end user. Also, they emphasized that advertised links also affect the page ranking of search engines. They discussed that a search engine needs to possess a very high precision as the user is likely to see the result of his or her query only within the top 10 links or first page of the output. Further, they highlighted the need for fast crawling technology, efficient usage of space to store search indices, the volume of indexed web and the effect of irrelevant web links in the output. Google uses search techniques based on proximity, anchor text and page rank information to improve the page ranking process. The Google has two unique features (i) it uses page rank algorithm (ii) it uses link structure to improve the rank. They created maps with more than five hundred million hyperlinks for rapid calculation of page rank; page rank can also assist in keyword-based web search. The web is a massive store of various type of web pages. There may be a huge distinction between two web pages regarding style, language, page formatting, etc. They highlighted the use of hyperlinks in handling these significant challenges effectively and efficiently. The efficiency of Google may be credited to programming languages used in its implementation like C++ or C. The architecture of Google consist of several vital

components including URL server, web crawler, store server, indexer. First of all, the URL server sends the list of URLs to the web crawler to retrieve the web pages. These retrieved pages are then suitably compressed to be stored within the repository by the store server. A new DocID is assigned to each page whenever a URL is fetched from it. The indexer then performs three functions (i) Reading web page from the repository (ii) Uncompressing pages (iii) Parsing all the links from a web page and store the details within the anchor file. The URL resolver then read the anchors file to convert all the relative URLs into absolute URL and is also responsible for assigning an ID to each of the document. The page rank of each document is then calculated by generating a link database by indexer.

2.1.2. Review of Search Systems based on Content Personalization

Content personalization means to present different content to different users on the same web site to quickly meet their personalized preferences. Kuppusamy and Aghila (2014) discussed the architecture of a personalized model to detect structural and content changes within a web page. The proposed model, i.e., CaSePer uses a hashing technique to identify segments used for reducing the search space and hence to quickly detect the changes within web page content. The change detection process is accompanied within two steps (i) Segmenting web page into smaller components (ii) Hash value calculation on smaller components. However, the proposed model may be improved by using advanced machine learning and big data analytics.

Sugiyama et al., (2004) discussed several techniques to adapt search results to the changing needs of the web user. They carried out several experiments to verify the effectiveness of various possible approaches such as (i) Collaborative filtering based user profiling (ii) Implicit relevance feedback (iii) Browsing history based user profiling. However, the highest accuracy was achieved by using collaborative profiling as it is more adapted to the personalized needs of the user. The proposed approach can be improved by using long-term browsing history of the user (Malhotra & Rishi, 2018b).

Bai et al. (2016) studied various challenges associated with news personalization. The proposed user profiles developed by the interaction between the user and the search engine can be used for personalization. They used datasets from Yahoo to verify their proposal. They claimed that personalized interest of users could be easily fetched if the information related to user interaction with news sites is augmented with interaction information of the user with the web. They further revealed that (i) blend of news and search profiles can improve the personalization (ii) Rank method is better than score method (iii) Search profiles can enhance the experience of both active and passive web users (iv) Search precision can be improved by regular usage of the system (v) Search profiles developed using browsing history of last three months is observed to significantly improve the precision (vi) Recent search profile play a significant role in enhancing the search precision. They proposed and implemented SP_RANK and SP_SCORE by baseline method. It combines (i) Cosine function and content-based similarity between the user profile and content of news profile (ii) news article relevancy comparison concerning the user having a similar profile. They claimed that news personalization could be further improved by using web search history. However, the study may be further improved by including the possible relationships between profiles of two different users to quickly figure out a prospective web link that can satisfy the requirements of a specific user.

Sudhakar et al. (2012) highlighted that even a state of art search engine returns a large number of redundant and irrelevant results in response to a user's search query. They proposed a weighted page search and ranking technique to satisfy the personalized needs of the user. As search engines employ distinct search and page ranking techniques for a particular search query. The different ordered output of each of the search engine leads to a competition between various businesses to be ranked on top. This competition may lead to a biased ranking of pages due to the inclusion of paid or advertised links and leads to various complications related to the satisfaction of the personalized search needs

of the user. They measured the effectiveness of their proposed approach concerning F1Score, precision and recall by calculating True Positives, True Negatives, False Negatives, and False positive observations. They claimed accuracy of the proposed method is more than 90%. However, they focused only on text-based information retrieval while ignoring other datasets.

Ferretti et al. (2016) proposed intelligent and automatic web content adoption system to satisfy the personalized needs of a user. They addressed the personalization needs of a specific user by adopting those elements of a web page that represents a barrier to the reading. They proposed a system ExTraS with modules: (i) adaptation module let the user set his or her preferences through a contextual menu-driven interface (ii) learning module track the browsing habits of the user to learn his or her preferences (iii) profiling module is used to store user preferences and manage his or her profile. The user profile is created and stored locally in the user device. However, if the user is accessing his profile on multiple devices, then the cloud-based system can be used to manage the profile. They employed a reward and punishment based machine learning system. The proposed system will first look for the characteristics being adapted by the user and starts tracking his or her browsing habits. The system will frame a user's profile by rewarding opted characteristic. The user can adapt to new habits, and the system assigns reward or punishment. However, no change in characteristic will lead to no reward. The system can also perform some adaptations for the convenience of the user, i.e., (i) font size to adapt to easily meet user's reading requirements (ii) font face to meet personalized preference of a user regarding font face like Times New Roman, Calibri etc. (iii) text alignment can be adjusted to fit the user preferences (iv) language translation (v) changing foreground or background color to adjust the font size or type and background color to best suit the user's requirements.

Malthankar and Kolte (2016) discussed an approach to achieve client side privacy protection during a personalized web search. They highlighted that search engines

might return irrelevant results due to text ambiguity, varying contexts, and users. Personalized Web Search (PWS) systems can address these problems adequately. PWS can employ any of two approaches (i) profile centered approach to achieve personalization with the enhanced usage of personal and behavioral information about the user which is accumulated from browsing the history of the user (ii) Click-log approach to give preference to web links clicked by the user in his or her browsing history. They proposed a customizable and privacy protection system, UPS to generate user profiles based on privacy requirements as mentioned by the user. They developed GreedyIL and GreedyDP algorithms to reduce information loss and also to increase discriminating power respectively. Zhou et al. (2017) highlighted that search engines are required to retrieve results that are of personalized relevance to individuals and not just relevant to the search query. They discussed a cross-language search based personalized web adaptation for document association based page ranking method. To achieve personalization, they used the KL divergence method between the improved search query and web document. A model is developed to adapt to learn from browsing the history of the user. The model assumes that the user usually searches web queries within a specific language and occasionally within other languages. They used inverse document and term frequency, i.e., *idf* and *tf* for the generation of the model. They verified single vector space model generated using one language could effectively search in another language. They also discussed an algorithm for expansion of search queries using a pair of Wikipedia document. The evaluation metrics used by them include (i) NDCG to represent Normalized Discounted Cumulative Gain (ii) MRR to represent Mean Reciprocal Rank (iii) P@1 to represent precision of top document (iv) P@5 to represent precision of top 5 documents.. The models used for the comparative study are LDA, QE, LM, Lexical, IM, ODP, and LMRM. The IM method was evaluated to be better than ODP, LEXICAL, LDA and QE models. The experimental evaluation proves that all the personalized models work well over non-personalized models. However, the

personalization strategies investigated are required to be integrated to perform better in other applications.

2.1.3. Review of Search Systems based on Contextual Knowledge

The contextual knowledge is vital for a search tool to personalized search on the web by providing hints about user interest. Xiang et al., (2010) discussed the importance of using contextual knowledge while ranking web pages. They further explained various principles and learning to rank approach to support contextual ranking of web pages. They proposed an empirical approach to solving two main issues:

- (i) How to benefit web page ranking using contexts?
- (ii) How to integrate web page ranking model with contextual knowledge?

However, the suggested approach will be satisfactory to deal mainly with meta page ranking in today's era of big data is yet to be verified.

Tanapaisankit et al., (2012) proposed an approach for search query expansion by using contextual knowledge. They used knowledge of user's profile to make the query more personalized. The proposed method was experimentally verified to improve recall and precision parameters of page ranking. However, the proposed approach can be further enhanced by incorporating knowledge of semantics and concept tuples.

Limbu et al., (2006) suggested a method to modify search queries to correctly reflect personalized tastes of the web user by utilizing implicit and explicit information such as user's browsing history and lexical database knowledge respectively. They used a thesaurus for query disambiguation and hence improved precision. Moreover, they added meta keywords for improving recall parameter of web search and page ranking. However, the process of query enhancement can be further enhanced by using a Boolean approach (Malhotra & Rishi, 2018b).

Agichetin et al. (2006) highlighted that web page ranking could be significantly improved through the incorporation of user feedback. They proposed an implicit page ranking model to determine the exact page rank. They suggested two approaches:

- (i) Using implicit feedback for verification of ranking
- (ii) Augmentation of page ranking using implicit feedback

The first approach is based on the assumption of no interaction between the user's implicit feedback and web page ranking process. This approach re-ranks the results according to previous user interaction with WWW and search engines. They computed the implicit score for each result through previous user interactions and then use summation to calculate the merged rank. However, in the second approach training and testing of the search algorithm is implemented. This approach usually divides user query data into training and testing data.

Ahmad et al. (2017) discussed a model based on learning to rank approach by identifying featured subsets shared by users. They highlighted that personalized ranking through modern search engines is a significant issue. The storage of user-specific ranking model may be done. However, this may lead to the implementation problem with traditional data mining techniques due to infinite profiles required to be stored. However, they claimed that this problem could be addressed by considering the browsing history of the user and by identifying a specific class of the user. They proposed learning to rank approach to address personalized ranking based on the determination of average rank for a particular cluster of users. The proposed system can quickly determine the relevant page ranking for a specific cluster of users. The proposed model may be configured in any of the three possible ways:

- Listwise
- Pointwise
- Pairwise

Fox (2005) et al. claimed that fetching explicit feedback is little tricky from the web users and they focused on implicit feedback linked to time spent on the web page, printing documents, purchase transaction, etc. may be used to generate predictively and learning models to determine the precision of a web search system. They studied two main aspects related to web search:

- Association between explicit and implicit feedback of the web user
- Various implicit measures related most to user satisfaction

They collected user implicit feedback using an add-in programmed for Internet Explorer to monitor user browsing sessions and to retrieve implicit feedback in the form of mouse and keyboard activities.

Moreover, explicit feedback was collected in two ways including individual output link visit. Secondly, session-wise feedback was also gathered, and a state machine was also used to prompt the user for explicit feedback collection. The individual result level activities monitored by the proposed system are as follows:

- The time duration between leaving and returning to the output links page
- Percentage of page scrolled by the user
- Way to exit the visited page- closing browser window, timeout, clicking an embedded link, etc.
- Printing or adding a page to favorites etc.

However, session-wise implicit feedback was collected by monitoring the following activities:

- Number of queries searched
- Number of output web links returned in the result
- Number of web links visited by the user

- Way to exit the session
- Average number of results returned in a specific duration
- Average of scroll time on each page
- Average number of pages printed by the user
- Average number of pages added to favorites by the user

They used Bayesian modeling to establish that some implicit measures together can produce a reasonable prediction about web links that can satisfy the user.

Liu et al. (2004) highlighted that popular search engines are built to serve all the users without paying attention to the personalized needs of a specific user. They proposed a technique to retrieve user profiles from user browsing history. They mapped user profile to the set of categories to learn personalized needs of the user and hence to appropriately expand the incomplete or ambiguous search query for fetching the best search results. They highlighted that user profile could be created automatically without his or her intervention. They used two profiles, i.e., *user profile* and *general profile* and claimed to achieve effective personalization. They evaluated an adaptive and four batch learning algorithms to generate the user profile. They used a weighted and voting centered algorithm for merging output web links. They created various categories based on user and general profile and search queries made by the user. These generated categories, in turn, are used to determine the context and hence to achieve personalization to satisfy the web search needs of the end user. The experimental results were satisfactory. However, improvements can be further made by choosing an extended query set and via evaluation with more number of volunteers.

2.1.4. Review of Search Systems based on Recommendation

Recommender system uses information about web user profiles, browsing history, etc. to predict the relevance of a specific web link to a web user. They make recommendations to satisfy the personalized needs of the user. Hence, a recommender system may be used

as a critical module for implementation of a personalized search system like the proposed Intelligent Meta Search System in the current research work. Cacheda et al., (2011) carried out a detailed comparison between various collaborative filtering techniques, mentioning their strengths and limitations. They suggested two new metrics, i.e., GIM and GPIM to use prediction accuracy for determining the effectiveness of a collaborative algorithm. These two metrics can simplify the evaluation by utilizing datasets available offline. They can quickly detect any bias within prediction accuracy.

Wasid and Kant (2015) suggested an approach to collaborative filtering based on Fuzzy and particle swarm optimization. The discussed approach can be used to quickly learn the preferences of the user and hence to provide personalized recommendations to the web user. However, the proposed system lacks the idea of concepts to improve the accuracy of personalized recommendations further.

Adamopoulos (2014) discussed improvement of collaborative filtering method for enhancement of prediction accuracy for both users and businesses. The idea of unexpectedness is also addressed for meeting user expectations. However, the effectiveness of the proposed recommender systems in studying the behavior of the online user is yet to be verified (Malhotra & Rishi, 2018b). Business companies gather a massive amount of data on customer's purchase from their daily transactions from various supermarket stores. Later they apply multiple data mining techniques on these extracted purchasing behaviors of customers, and extracted patterns could be used in diversified businesses for better decision making.

Bouadjenek et al. (2016) discussed that finding relevant information is challenging through search engines due to the following reasons:

- Web user is usually unaware of the web page until finding the same
- Web user is generally unaware of how to formulate a correct query

The resulting web links produced in the output of search engines are the same irrespective of the user issuing the query. Hence, results are usually irrelevant. To improve the page ranking process, popular techniques discussed in the literature are as follows:

- Query expansion or disambiguation
- Re-ranking output links to match personalized user preferences as retrieved from the user profile
- Refining the way to represent web documents and search queries to improve the information retrieval system

They proposed to use social media information of the web user to enhance his or her personalized search experience through the implementation of the PerSaDoR framework. The proposed method can compare user query concerning the matching profile of the two users and can recommend web links to match the needs of the web user.

Verma et al., (2015) demonstrated usage of recommender systems based on semantic web and neural networks for correct page ranking of E-Commerce websites. They proposed five modules (i) Module for web dictionary implementation after preprocessing of pages (ii) Module to determine priority of web page based on its textual content (iii) Module to determine priority of web page based on time spent by previous user (iv) Module for semantics-based recommendations (v) Module to determine the priority of web page using *back propagation neural network*.

Ding et al. (2004) proposed a semantic web based metasearch tool, 'Swoogle.' The proposed retrieval system will calculate the proximity between various web documents using metadata. They computed the ontology rank to determine the semantic significance of the web document. They claimed that popular search engines could work well only with natural languages and hence can't take benefit of SWDs due to failure to understand their structure. The proposed system can explore various SWDs through

multiple crawlers. The Swoogle examined ontologies were also used to build the ontology dictionary.

Birukov et al. (2005) discussed that finding relevant information on the web is difficult due to the continuous increase in websites. They proposed an agent-centered recommendation system in assisting web search and the architecture for SICS, i.e., Systems of Implicit Culture Support consist of the following components:

- Observer - To store information about actions executed by the web user
- Inductive Module- To apply data mining techniques to discover hidden patterns from stored information about user behavior
- Composer – To make recommendations for final output links based on information gathered from an observer and inductive module. The recommendations made will satisfy the personalized search needs of the user effectively and efficiently

There are few improvements sought by the proposed system; for instance, a composer should consider balancing the number of acceptances and rejections to improve the personalized search experience of the user further. The proposed method can take acceptance and rejections only from Google, but the same can be developed by considering more agents to frame the personalized recommendations accurately.

2.2. DETAILED COMPARISON OF VARIOUS VERSIONS OF INTELLIGENT META SEARCH SYSTEMS

The current research work proposes a novel page ranking algorithm and its deployment in the form of an Intelligent Meta Search System (IMSS) tool. IMSS tool has gone over a lot many improvements in terms of personalized search precision and analytics capabilities as offered to the end user. This section discusses a detailed review of the various versions of the IMSS tool as proposed and published by us from time to time.

Malhotra et al., (2017a) discussed the implementation of a metasearch and page ranking tool to prove the effectiveness and efficiency of the proposed CPR algorithm. The

proposed interface of the tool is shown in Fig. 2.1. The tool can use any or all of the four background search engines, i.e., Yahoo, Google, Ask and Bing. The tool will rank the various links returned by these search engines by response time and security protocol used by the candidate web page. However, the system does not incorporate features of personalized page ranking to satisfy the specific needs of the user. Malhotra and Rishi (2018a, b) discussed various limitations of traditional page ranking systems. We highlighted that the general search and page ranking system is not evolved enough to work out effectively within the E-Commerce environment. In this paper, a relevancy vector-based page ranking algorithm is proposed that uses cloud technology and is based on the second generation big data analytics. The proposed algorithm is used to implement IMSS-AE tool especially adapted to rank E-Commerce websites to suit the personalized needs of the customer. The experimental & graphical analysis compare the page ranking precision between the IMSS tool and popular search engines like Google, Yahoo, Dogpile by response time, page freshness and personalized relevancy. However, the proposed work lacks accuracy in the prediction of user interest due to a missing machine learning module.

Meta Search and Page Ranking Tool			
Select Search Engine Tabs for Intermediate Document Retrieval			
GOOGLE	YAHOO	BING	ASK
Enter Search String: HDFS and Map Reduce			
	Search	Reset	
Ranking Box.....			
Rank	Web Links	Security	Response
1	https://en.wikipedia.org/wiki/Apache_Hadoop	HTTPS:	00:00:00:10ms
2	www.cloudera.com/content/cloudera/hdfs- -----	N/A	00:00:00:25ms
3	www.gttibm.org/software/datacom/infospher/	SSL	00:00:00:33ms

Fig. 2.1 Interface of page ranking tool by Malhotra et al. (2017a)

INTELLIGENT META SEARCH SYSTEM- ADVANCED E-COMMERCE			
SIGN UP/ New Customer		User ID: DM@UOK	Password: *****
Personalized Search		Advanced Criteria Search	
YAHOO	GOOGLE	DOGPILE	
Page Loading Speed	Transaction Security	Response Time	
<div style="border: 1px solid black; padding: 5px; display: inline-block; margin: 10px auto; width: 80%;">Enter Search String: online belt purchase</div>			
<div style="display: flex; justify-content: space-around; margin: 10px auto;"> <div style="border: 1px solid black; padding: 5px 15px;">SEARCH</div> <div style="border: 1px solid black; padding: 5px 15px;">RESET</div> <div style="border: 1px solid black; padding: 5px 15px;">FAST FORWARD >></div> </div>			
<div style="border: 1px solid black; padding: 5px; display: inline-block; margin: 10px auto; width: 80%;">Personalized Expanded Search String by IMSS-AE: online belt purchase for women</div>			
<div style="display: flex; justify-content: space-around; margin: 10px auto;"> <div style="border: 1px solid black; padding: 5px 15px;">SEARCH</div> <div style="border: 1px solid black; padding: 5px 15px;">RESET</div> <div style="border: 1px solid black; padding: 5px 15px;">FAST FORWARD >></div> </div>			
RANKING BOX...			
RANK	WEB LINK	RESPONSE TIME	FEEDBACK (CORRECT RANK?)
1	www.amazon.in/clothing/women	00:00:00:15ms	Yes <input type="radio"/>
			No <input type="radio"/>
2	www.myntra.com/women-belts	00:00:00:36ms	Yes <input type="radio"/>
			No <input type="radio"/>
3	m.jabong.com/women/access	00:00:00:49ms	Yes <input type="radio"/>

	ories		No <input type="radio"/>
--	-------	--	--------------------------

Fig. 2.2 Interface of IMSS-AE tool by Malhotra & Rishi (2018a)

The personalization approach and ACVPR algorithm in the present research work is the enhancement of the RV page ranking algorithm due to the incorporation of the machine learning model. The interface of IMSS-AE tool is shown in Fig. 2.2.

Malhotra et al. (2016) highlighted electronic commerce market in India has recorded significant growth of more than 400 % in last four years and is expected to grow more rapidly, i.e., almost five-fold by December 2016. As a result, companies are not taking a chance to satisfy the personalized purchase requirements of customers and hence to fetch good revenues and branding of their business. Moreover, the web doesn't possess any catalog like feature, so, most of the E-Commerce users are dependent on search engines like Google, ASK, Yahoo, etc. to search for relevant E-Commerce website for online purchase of a specific product. Moreover, none of the search engines could index more than 16% of the web concerning the literature. The issue is not just the volume but is also the relevancy concerning customer requirements, if the query is incomplete or ambiguous then search engines return a large number of links in the search output as they tend to return links by interpreting all possible meanings of a query. The proposed intelligent search tool, IMSS-E, for ranking of E-Commerce websites to assist an online customer in finding a suitable site on top while searching for a specific product. The tool can also support online retailer to structure his website well to satisfy the personalized purchase requirement of the customer better. The proposed research work utilizes Apriori mining - map reduces based big data analytics framework, supported by semantic web and back-propagation neural network to well adapt to personalized requirements of the customer by learning from previous errors in ranking E-Commerce websites. We experimentally verified the improved efficiency of the deployed tool by comparing the ranking precision of the IMSS tool with a popular metasearch engine, Dogpile. The earlier proposed IMSS-E tool is shown in Fig. 2.3.

Malhotra et al. (2017) highlighted that in this present era of big data, different search engine users have different information requirements at different intervals of time. Thus, search results should be adapted to the user's needs. In this published paper, we proposed a novel approach to adaptive web search augmented with the capabilities of carrying out *Big Data Analytics* using the second generation HDFS. Moreover, unlike conventional personalization techniques, the proposed method does not require additional efforts from the user such as reporting feedback or ratings, etc. The proposed system can be implemented in the form of Intelligent Meta Search System (IMSS Tool) to overcome the problem of irrelevant web page retrieval faced by the user of generic search engines (<https://link.springer.com> [166]). The adaptive search, when supported by HDFS-cloud framework, leads to a smooth & efficient analysis of big data available on WWW to retrieve useful personalized page ranking patterns. Search engines are known to extract far more extensive information, but still, no search engine can index more than about 16% of the indexable web. The interface of the proposed IMSS tool is shown in Fig. 2.4.

The issue is not just only the volume but is also the relevancy concerning user's information needs. When different users search the same query, even a state of art search engine returns the same result, irrespective of the user submitting the query. For example, if a user is tech-savvy and usually searches for laptop or mobiles, then an incomplete query search like Blackberry should return documents related to Blackberry mobiles by intermediately expanding the query rather than returning the documents of some fruit. There are various types of conventional personalized search systems as discussed in the literature. However, these search systems fail to satisfy the personalized user requirements without having explicit ratings or feedback from the user. Moreover, such systems can't handle the second generation big data as they do not just require scalability, partial failure support, etc. but also need to support multiple analytic methods on varied data types, as well as the ability to respond in near real time.

2.3. LIMITATIONS OF EXISTING SYSTEMS

As discussed in section 2.1.1, personalized search systems based on hyperlinks take more time to calculate relevance, and they do not satisfy user's contextual information needs because bookmarks, browsing history, etc. are not taken into consideration. The search systems based on content personalization, discussed in section 2.1.2 increases load on the user as they require the user to register his or her personal preferences, and also such systems can't adapt automatically with a change in user needs and users are expected to change recorded preferences whenever their interest changes. Recommender systems as discussed in section 2.1.3 require user's ratings to give good recommendations. Systems based on contextual knowledge as mentioned in section 2.1.4, require having user's explicit data for correct page ranking. However, many web users are not taking interest to spare time for giving explicit ratings and precise information required to support such search systems. The earlier proposed versions of Intelligent Meta Search Systems proposed and published by us as discussed in section 2.2 step by step overcome various limitations of the conventional search systems and achieve multiple objectives of the current research work. (Malhotra and Rishi, 2017b).

INTELLIGENT META SEARCH SYSTEM E-COMMERCE			
Sign Up	User ID: Cus@EC	Password: *****	
Select Meta Search Engine Tabs for Document Retrieval			
Dogpile	Mamma	Kartoo	MetaCrawler
Adaptive Search		Criteria based Search	
Response Time	Page Loading Speed	Transaction Security	Page Freshness
Enter Search String: Online purchase of SamsungS7 Mobile			
Search		Reset	
Rank	Web Links	Security	Response
1	https://www.samsu...ngabc.com	https:/	00:00:00:39 ms
2	www.xmobile.com	SSL	00:00:00:67 ms

Fig. 2.3 Interface of IMSS-E tool by Malhotra et al. (2016)

INTELLIGENT META SEARCH SYSTEM			
Create New User Profile	User ID: Dheeraj@UOK	Password: *****	
Select Search Engine Tabs for Intermediate Document Retrieval			
GOOGLE	YAHOO	BING	ASK
Take me fast (Personalized Search)		Advanced Search (Select Criteria)	
Response Time	Loading	Security	Page freshness
Enter Search String: HDFS and MapReduce			
<input type="button" value="Search"/>		<input type="button" value="Reset"/>	
Rank	Web Links	Security	Response
1	https://en.wikipedia.org/wiki/Apache_Hadoo	https:/	00:00:00:10ms
2	www.gttibm.org/software/datacom/infosphere/hadoop/mapreduce	SSL	00:00:00:33ms

Fig. 2.4 Interface of IMSS tool by Malhotra & Rishi (2017)

2.4. COMPARATIVE STUDY ON SEARCH SYSTEMS

Table 2.1 summarizes the comparison between various intelligent systems proposed in the literature and different versions of IMSS tool time to time proposed and published by us on the basis of following parameters and each one is represented by a corresponding Roman numeral in table 2.1:

- (i) Personalization Type
- (ii) Meta Search Enabled
- (iii) Intelligent Technology Used
- (iv) Support for Big Data Analytics and Name of Platform Used
- (v) Evaluation Metrics

Table 2.1 Comparison between existing and proposed personalized search systems

S. No.	Publication	(i)	(ii)	(iii)	(iv)	(v)
1	Arya et al. (2018)	Job Search Personalization	No	Social Network based Recommender System	No	Similarity Score
2	Malhotra et al. (2017b)	E-Commerce websites ranking	Yes	No	Yes-BDAS	Personalized Precision
3	Malhotra et al. (2017a)	Personalized Web Page Ranking	Yes	No	Yes-HDFS, MR	Precision and Relevancy Vectors
4	Malhotra & Rishi (2017)	Web Search Personalization	Yes	Semantic Web	Yes-HDFS, MR	Page Rank Precision
5	Malhotra & Rishi (2016)	E-Commerce Personalization	Yes	Semantic Web and Neural Network	No	Advanced & Personalized Search Precision
6	Bouadjenek et al. (2016)	Social Information based Personalization	No	Vector Space Models	No	Mean Average Precision and Mean Reciprocal Rank
7	Verma et al. (2015)	E-Commerce Personalization	Yes	Semantic Web	Yes - HDFS	Relevancy Vectors-SRV,CRV
8	Shafiq et al. (2015)	Web Search Personalization	No	Social Network Recommender System	No	Click Through Rate, MAP, Click Entropy
9	Bibi et al. (2014)	Web Search Personalization	No	Concept-based filtering and	No	Snippet Frequency in

				Semantic Web		Title, URL, and Summary
10	Malhotra (2014)	Web Search Personalization	No	Back-propagation Neural Network	No	P(Y)-Precision at Y Metric
11	Malhotra & Verma (2013)	Web Page Priority Determination	No	No	No	Keyword Match Frequency-Found, Nfound and Time Spent Statistic
12	Moawad et al. (2012)	Web Search Personalization	Yes	Multi-Agent System and Semantic Web	No	Search Precision
13	Collins-Thompson et al. (2011)	Web Search Personalization based on Reading Proficiency	No	Semantic Web	No	Mean Reciprocal Rank (MRR), Page Reading Level, etc.
14	Kim et al. (2010)	Personalized Web Search	No	Concept Network based Recommender System	No	MAP-Mean Average Precision, K-Precision

From the above-shown comparison of various essential parameters of web personalization search systems discussed in the literature and earlier proposed versions of IMSS tool helps in choosing various parameters to study, and are suitably incorporated in the final deployed version of the Intelligent Meta Search System (IMSS) and Advanced Cluster Vector Page Ranking (ACVPR) algorithm. The personalization approach discussed in the current research work has the following attributes:

- (i) *Personalization Type*: Web Search Personalization
- (ii) *Meta Search Enabled*: Yes, Google, Qwant, and Bing as Backend Search Engines

(iii) *Intelligent Technology Used*: Machine Learning Techniques- Logistic Regression and Collaborative Filtering

(iv) *Support for Big Data Analytics and Name of Platform Used*: Yes, HDFS

(v) *Evaluation Metrics*: MAE, RMSE, Specificity, Sensitivity, Precision, and Recall

2.5. CHAPTER SUMMARY

This chapter discusses the category specific literature review broadly about four types of conventional search systems. This chapter mainly focused on the first two objectives of this research work, i.e., (i) To underline the flaws in existing data mining techniques to extract useful page ranking patterns from big databases of search engines. The objective is accomplished by highlighting various shortcomings of conventional search systems while operating within the modern generation of big data such as real-time response, elastic scaling, load spikes resistant and failure resistant. (ii) To make a comparative analysis of various traditional personalized search systems. The objective is accomplished by discussing category specific detailed features, limitations of conventional search systems and tabular comparison of personalized search systems and various earlier proposed versions of the IMSS tool.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1. INTRODUCTION

This chapter discusses deployed system architecture and compares and contrasts between various available cloud computing platforms to identify a best-suited platform for the implementation of personalized metasearch application. The detailed tabular comparison between different deployment platforms is followed by the discussion of the system architecture of the pioneered Intelligent Meta Search System (IMSS). This chapter first covers the basics of cloud computing. The objective of cloud computing is to move from the physical infrastructure and locally managed software services to virtualized services available on user's demand. A cloud may be public cloud, private cloud or a mix of both, i.e., a hybrid cloud which allows the authorized users and organization to take dual advantage of secure network storage along with public functionalities that are easily and anywhere accessible using the Internet. Cloud computing may be described as a model for enabling convenient, on-demand access to computing services and a configurable pool of resources like applications or servers that can be rapidly configured and released in real time. These on-demand resources and services are offered by IT giants like Google, Amazon, and Microsoft, etc. The overview of cloud computing is shown in Fig. 3.1.

Some of the key characteristics and properties of cloud computing are as follows:

1. Cloud services are elastic and are available as on-demand services. Cloud allows users and organizations to configure resources or services in real time.

2. Cloud computing allows multi-tenancy, i.e., multiple users can access a single server to process their data without purchasing applicable licenses for different applications.

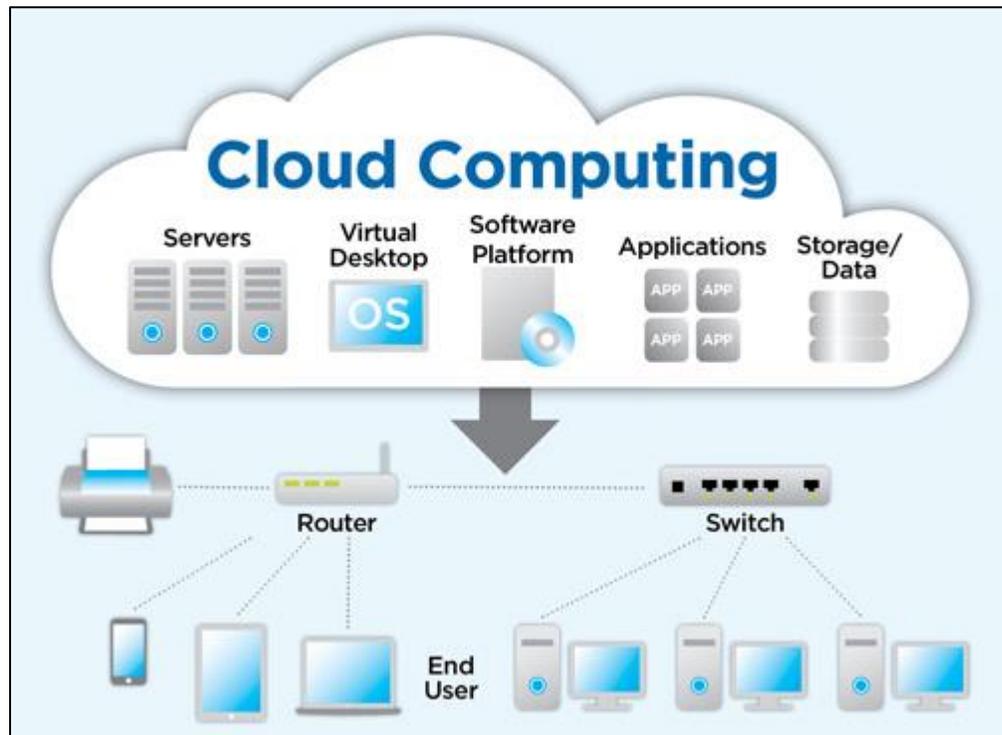


Fig. 3.1 Overview of cloud computing (<http://osaip.com>[164])

3. Cloud computing follow "*use and pay*" principle which means one need to pay as much as long as he or she uses the services.
4. Cloud computing uses a technique like client-server model which means that the user needs to connect to a server and his or her process may run on one or multiple computers using virtualization technique.
5. Cloud computing provides a robust user and task-centric programming platform.
6. Cloud computing is based on a strong foundation of services management and software development.

7. Cloud computing players include organizations, vendors, partners, and developers.

3.2. CLOUD SERVICE MODELS

The cloud computing models include IaaS, SaaS and PaaS. The detailed description of these three models is as follows:

- *IaaS (Infrastructure-as-a-Service)*: This model provides various resources like CPU, storage, bandwidth, power, etc. For example Rackspace, Amazon Web Service (AWS), Verizon, etc.
- *SaaS (Software-as-a-Service)*: This model provides user access to the application without buying licenses. However, here, the user is not equipped with a control to network, hardware, security or OS. For example Google Cloud Platform (GCP), Salesforce.com, Zoho, etc.
- *PaaS (Platform-as-a-Service)*: This model provides network, operating system, and hardware infrastructure to provide the necessary services and a complete platform for installing user's applications. For example Informatica-on-demand, Google cloud, Azure, etc.

3.3. COMPARISON OF DEPLOYMENT PLATFORMS

To choose appropriate deployment framework for a web search and ranking application, one needs to compare various aspects such as capabilities for partial failure support, fault tolerance, scaling, real-time processing and efficiency in iterative execution. Here, in the current research work, various existing deployment paradigms are examined as discussed in section 3.3.1, 3.3.2 and 3.3.3 to explain some of the characteristics of different cloud-based platforms useful for the implementation of Intelligent Meta Search System (IMSS) tool in the current research work (Malhotra & Rishi, 2018a).

3.3.1 Types of Deployment Platforms

Various existing cloud-based deployment platforms are explained as follows (Khurana, 2014)

- In one of a kind, cluster utilizes blob storage space as a primary storage space such as Azure blob store, S3. Here temporary clusters are implemented, and they exist only till the period of workflow execution. Blob store act as a source and destination of the workflow. Here, virtual machines may be considered as task execution containers.
- In another type, first generation HDFS (Hadoop Distributed File System) is used as a primary storage space. In contrast, here, persistent clusters are used for long-term storage. Moreover, virtual machines are persistent, and they can perform execution as well as data storage. This type may even use blob storage for cyclic backups and to give data to HDFS. This kind of cloud deployment platform is useful for workloads of type SLA batch workloads, Ad Hoc interactive, and Ad Hoc batch. For instance, interactive SLA workloads are usually deployed on HDFS due to virtual machines requirement as servers and blob storage requirement as a backup.

3.3.2 Second Generation HDFS

The progress from the first generation of HDFS to the second generation is due to the change in processing capabilities from a batch-oriented environment of the first generation HDFS to comparatively much more interactive processing capabilities of the second generation HDFS. The first generation HDFS is not suitable for a personalized web search application as it has limited support to machine learning necessary to determine personalized page ranking order. Furthermore, first-generation HDFS is more I/O intensive rather than a second-generation interactive version of HDFS which more

suits to a personalized web search application like IMSS tool implemented within current research work.

Moreover, due to recent technological shifts, second generation big data processing systems need to support multiple analytic methods on varied data types, and the ability to respond in real time. The essential characteristics of first-generation HDFS, i.e., partial failure support, scalability through data streaming and global memory scheduling is also required to be continued by second-generation HDFS as shown in Fig. 3.2. (Malhotra and Rishi, 2018a). There are two significant trends of the second generation HDFS based big data search and ranking systems (Khurana, 2014)

- There is rapid growth in network bandwidth as compared to hard drive bandwidth
- Development of In-memory computation models such as *Spark* allows intermediate results to be kept in memory and hence reduces the overhead of iterative analytics

Second generation HDFS is adapted as a long-term store from where web applications read their initial data and write back their final results. The data layer is subdivided into various segments for steady storage and provides storage for intermediate objects separately. However, one of the limitations of HDFS lies in running iterative algorithms efficiently. Map function requires to read data at the start of iteration and to write back the results to the disk at the end of the iteration. This frequent access to disk in writing and reading data is responsible for low performance and efficiency degradation. (Singh and Reddy, 2015)

The two major strengths of the second generation of HDFS are as follows:

- YARN
- HDFS Federation

YARN is a resource manager and is often treated as an operating system of Hadoop. It is responsible for managing workload and security controls to ensure high availability of Hadoop and Map- Reduce platform for efficient big data processing. YARN was created

as a result of the separation of resource management capabilities from processing engines capabilities of Map-Reduce implementation within the first generation of HDFS. Moreover, YARN in addition to Map-Reduce supports multiple processing models for big data analytics. HDFS federation allows creating multiple name nodes for managing a Hadoop cluster. However, only one name node was permitted in the first generation of HDFS. The second generation of HDFS, on the other hand, allows for horizontal scaling for performance improvement. The most significant benefits of the second generation HDFS over the first generation HDFS is processing speed and reliability improvements due to the introduction of YARN and HDFS federation.

The reliability feature is achieved by HDFS federation as it avoids complete disruption of the Hadoop cluster due to the failure of single name node available as it allows having multiple name nodes. Moreover, second-generation HDFS allows executing different type of jobs at the same time. The second generation of HDFS allows machine learning, SQL interaction for big data processing, etc.

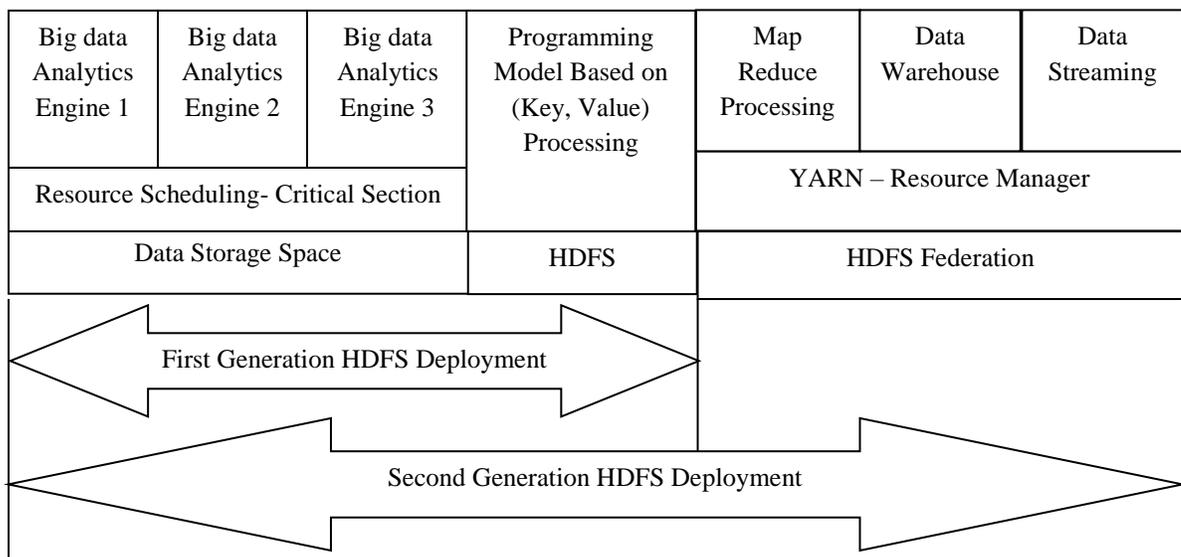


Fig. 3.2 Second vs. first generation HDFS (Malhotra and Rishi., 2017)

3.3.3 Ranking Comparisons of Deployment Platforms

Table 3.1 shows a ranking comparison of various possible big data deployment frameworks on different characteristics such as scaling, fault tolerance. Here Rank -1 shows the best option and Rank - 5 for worst choice among all of the listed platforms. It may be noted that this ranking table provides a general idea regarding the strengths and weakness of various platforms and it mainly depends on the specific application. In general, big data applications, there is a tradeoff between scaling and real-time processing capabilities. For example, in web search applications, indexing process requires a highly scalable platform to handle billion of web pages returned by some supporting search engines. This indexing accomplished via HDFS and Spark are the optimal choices for web search applications (Singh and Reddy, 2015), and hence these are preferred and proposed deployment frameworks for website search and ranking applications.

In the implementation of IMSS tool within current research work, Hadoop Distributed File System (HDFS) platform is chosen due to its high scaling and fault tolerance ranks which are the two most essential requisites in the implementation of a personalized metasearch tool, such as, Intelligent Meta Search System (IMSS). The HDFS platform allows the Map-Reduce framework for the analysis of *Big Data* and usage of Hadoop for storage in an effective and efficient manner.

Moreover, preference is also given to HDFS over SPARK platform due to easy availability and adaptability of hardware and software related infrastructural requirements for *HDFS- Map-Reduce* environment and hence to improve the probability of increased usage and popularity among prospective users (Malhotra & Rishi, 2018a).

Table 3.1.Ranking comparison of existing and proposed platforms

Platform	Scaling Rank (Type)	Fault Tolerance Rank	Real-Time Processing Rank	Iterative Tasks Rank
HDFS	1 (Horizontal)	1	4	4
SPARK	1 (Horizontal)	1	4	3
PEER TO PEER	1 (Horizontal)	5	5	4
HPC CLUSTERS	3 (Vertical)	2	3	2
MULTICORE	4 (Vertical)	2	3	2
GPU	4 (Vertical)	2	1	2
FPGA	5 (Vertical)	2	1	2

3.4. CLOUD ARCHITECTURE OF THE IMSS SYSTEM

This section discusses Hadoop 2 based cloud architecture of the implemented Intelligent Meta Search System (IMSS) in the current research work. Hadoop architecture is built for efficiently writing applications which process a massive amount of data in parallel on large clusters of commodity hardware in a reliable and fault tolerant manner in a cloud environment. Hadoop is an open source implementation of Google Map-Reduce architecture, sponsored by the Apache software foundation. Hadoop consists of two core components: The Hadoop Distributed File System (HDFS) and Map-Reduce (<https://backtobasics.com> [159]). A set of machines running HDFS and Map-Reduce is known as a Hadoop cluster which helps store data. There are many other projects based around core Hadoop often referred to as the "*Hadoop Eco System*" such as Pig, Hive, HBase, Flume, Oozie, Sqoop, Zookeeper, etc. (Malhotra et al., 2017)

3.4.1 The Map-Reduce Programming Model

Map-Reduce is the system (programming model) used to process data in the Hadoop cluster. It is used for parallel computing for large-scale data sets (more than 1TB). Map-Reduce deals with massive data sets on a cluster mainly through the two steps of "Map Phase" and "Reduce Phase". First of all, Map processes the input key and value pair to generate the intermediate results followed by shuffle and sort phase. Finally, the Reduce phase is used to merge the processed results to obtain the output appropriately. It is shown in Fig. 3.3. Here, Map method will accept a key as search engine ID for each retrieved web links cluster from various background search engines and the second argument is weblog to tokenize each of the entry of link entry in the weblog for counting frequency of each of the keyword in a search query. Insert () method is used to generate elements in the list by inserting numeric one corresponding to each occurrence of a keyword as we token. However, reduce method is implemented to cumulate over all the appearance of each keyword as indicated by Map () function through the insertion of numeric 1(one) to determine the frequency of the keyword in each of the web document and hence to conclude the content relevancy vector of retrieved web documents from various search engines. (Malhotra & Rishi, 2018a). Various components of the architecture of the proposed and implemented system, i.e., IMSS are shown in Fig. 3.4. These components are explained as follows (Malhotra et al., 2017):

Cloud Cluster Manager: It is a supervisor of metasearch tool architecture and is responsible for web page distribution, web page storage; inter-cluster communication in the HDFS, etc. Its design is crucial for the efficient performance of the IMSS tool. The cloud cluster manager is responsible for communication between various Hadoop clusters for effective and efficient processing of the web search data. It is also responsible for storage and retrieval of web pages within cloud storage.

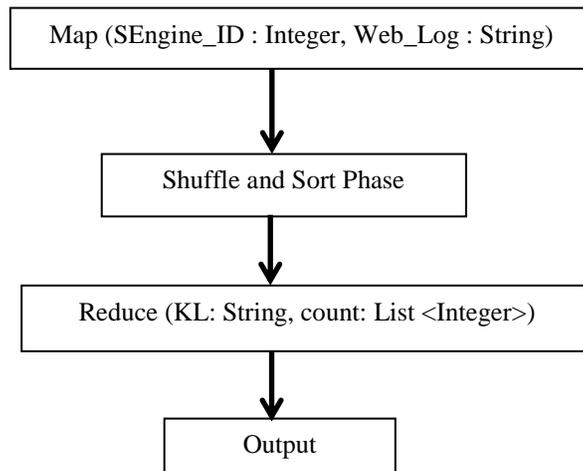


Fig. 3.3 Map-Reduce functionality of the IMSS tool

```

Map (SEngine_ID : Integer, Web_Log : String)    //Web Log Cluster processing
{
  List<String> TL: = Tokenize (Web_Log)           // TL- Token List
  While (Web-Token in TL)
  {
    Insert ((String) KL, (Integer) 1)           // KL- Keyword List
  }
}

Reduce (KL: String, count: List <Integer>)
{
  Integer Freq = 0
  While (KL)
  {
    Freq = Freq + 1 // Keyword frequency count
  }
  Insert ((String) Web-Token, (Integer) Freq)
}
  
```

NameNode (NN): In this architecture, NameNode is considered as a master that direct commands to all DataNodes (DN) which are regarded as slaves. NN handles all bookkeeping tasks of HDFS. It serves file system metadata and information related to DN entirely from RAM for faster access and monitors the overall health of the distributed file system. The NN uses two files, i.e., *Fsimage* and *Editlogs* to describe the entire state of the cluster and to track all updates in the file system respectively. NN is a node which is running all the time. However, there is a tradeoff; if NN fails then, the cluster becomes inaccessible. On the other side, if some of the DN fails then cluster is still accessible. (Malhotra et al., 2017)

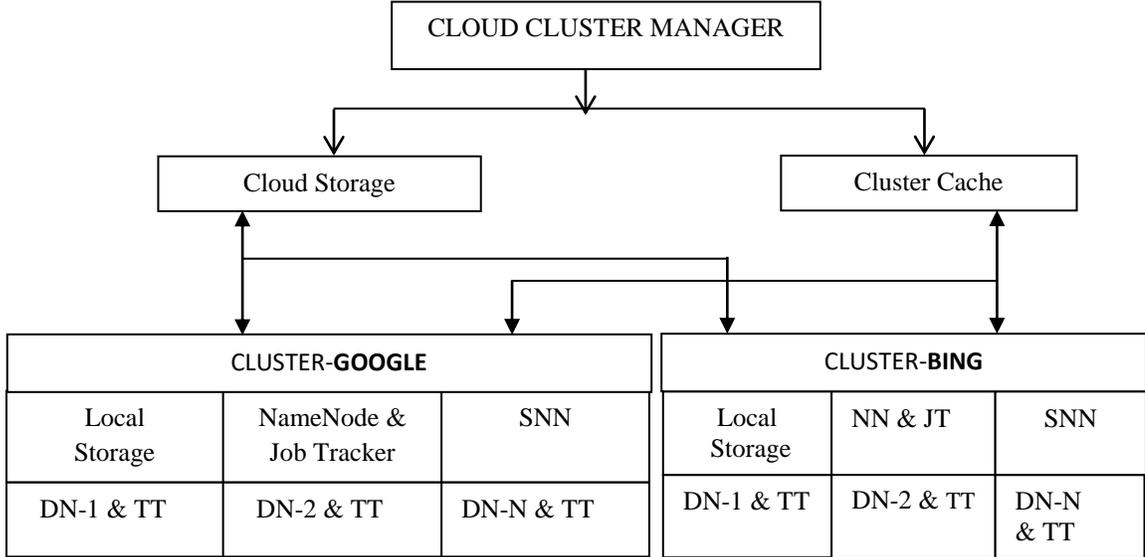


Fig. 3.4 Architecture of the IMSS tool

Moreover, NN is also responsible for maintaining the redundancy of data in the blocks by mentioning the replication factor. This information is vital so that cluster should remain accessible in case of failure of any of the DN. Hence system admin will take care to ensure that NN hardware is reliable and available for smooth functioning. NN is also responsible for tracking the health status of DN via heartbeat connection (<http://blog.socratesk.com> [161]).

Secondary NameNode (SNN): SNN is not a backup of NN, but it does some house-keeping tasks for NN. The primary difference between NN and SNN is that SNN does not keep track of the changes made to Hadoop Distributed File System, i.e., HDFS. However, SNN maintains *journaling* or *Editlog* where incremental modifications are made to the metadata. NN does not participate in *Editlog* or journaling activities as NN is usually occupied in critical activities like rack placement strategy, block read and writes, etc. SNN communicates with NN to take snapshots of HDFS metadata at pre-defined time intervals defined by Hadoop cluster for minimizing the downtime and loss of data. Hence whenever NN fails, then *Editlog* and image file can be used for back up to ensure smooth conduct.

Job Tracker (JT): In this architecture, JT is considered as a communication link between NN and client (Map-Reduce application). This daemon is responsible for accepting expanded search query requests from the client and allocating those tasks to various Task Trackers residing on various Data Nodes. The JT can also reallocate the job to a new TT on the replica of a DN in case of a failure of a node to ensure that DN failure does not lead to the cluster breakdown or job failure.

Task Tracker (TT): The TT is active on DN, unlike JT which is active on a master node, i.e., NN. TT and JT both are known as '*computing nodes*'. Besides, TT is also required to update its existing status to JT via heartbeat connection. The responsibility of a TT here in the system architecture of IMSS is to accept the search query request from a JT and to collect the page links returned by a corresponding search engine and to sort those links by personalized preferences of the user.

DataNode (DN): In this architecture, DN is considered a slave node and is responsible for storing data. Whenever it is required to read or write an HDFS file, the file is broken into blocks, and the NN will tell the client in which DN each block resides (<http://pramodgampa.blogspot.com> [162]). A client is a Java library or a service provided

by Hadoop. The client communicates directly with the DN to process the local files corresponding to the blocks (<http://ioenotes.edu.np> [160]). Furthermore, a DN may interact with other DN to replicate its data blocks for redundancy. DN's are regularly reporting to the NN. Upon initialization, each of the DN informs the NN of the blocks it is currently storing. After this mapping is complete, the DN's continually poll the NN to provide information regarding local changes as well as receive instructions to create, move or delete blocks from local disk (<http://a4academics.com> [167]). The storage of blocks in DN's is governed by the concept of *Rack Placement Strategy* (Malhotra et al., 2017), where

- The first copy of the block will be placed on any rack
- The second copy will always be different than the Rack of first copy called as off the rack
- The third copy will always be placed in the same Rack but with different DN

Information Flow in HDFS

The information flow within the Hadoop Distributed File System of metasearch tool, i.e., IMSS is shown in Fig. 3.5. The steps for information flow are as follows:

- Firstly, a Map-Reduce client process submits a job to be processed to the JT.
- The JT is responsible for retrieving data location information from NN and try to assign the job to TT based on the exact location of data within a specific DN.
- The TT initiates the map tasks and is responsible for giving the status of execution to the JT.
- The processed results are passed from the Map task to Shuffle and Sort phase and finally to the Reduce task.

The Reduce task is responsible for processing the intermediate results from Map task and writing the final results to the Hadoop Distributed File System.

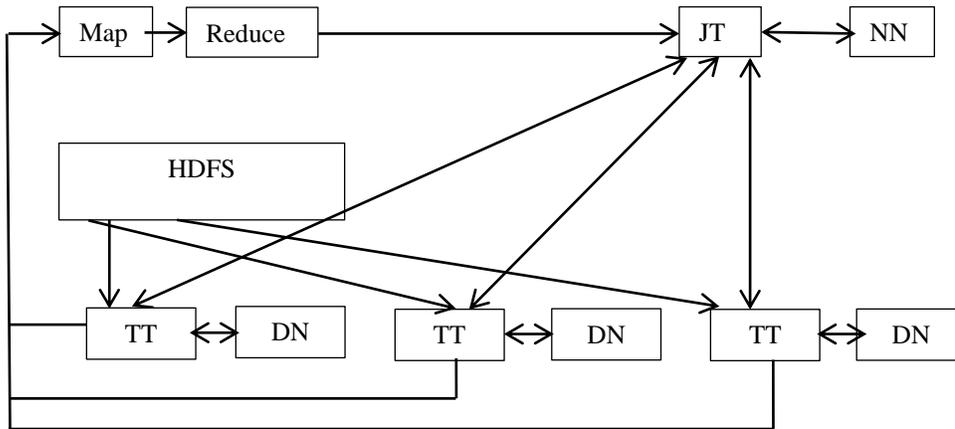


Fig. 3.5 Simplified information flow diagram in HDFS

3.5. HADOOP IMPLEMENTATION OF ACVPR ALGORITHM AND IMSS TOOL

The new web search personalization algorithm, that is, Advanced Cluster Vector Page Ranking (ACVPR) algorithm is implemented in the form of a metasearch tool, that is, Intelligent Meta Search System (IMSS) on Google Cloud Platform (GCP) using Hadoop and Map-Reduce configuration. The Google cloud platform provides essential features and resources like compute engines, VM instances with required configurations by "Pay on Usage" basis. The GCP provides credits to use the services free of cost initially. The IMSS tool is implemented and deployed using a single cluster setup on GCP. The detailed cluster information regarding deployment of IMSS tool within current research work is shown through screenshots from Fig. 3.6 to Fig. 3.13.

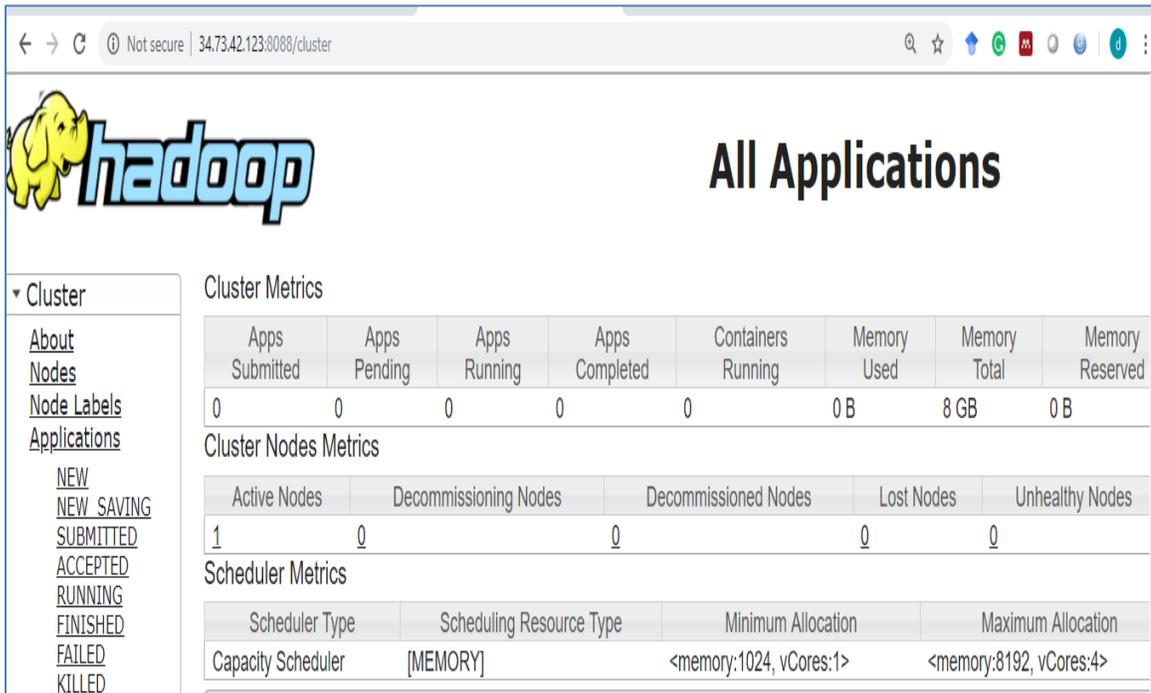


Fig. 3.6 Nodes information in the Hadoop cluster of IMSS tool

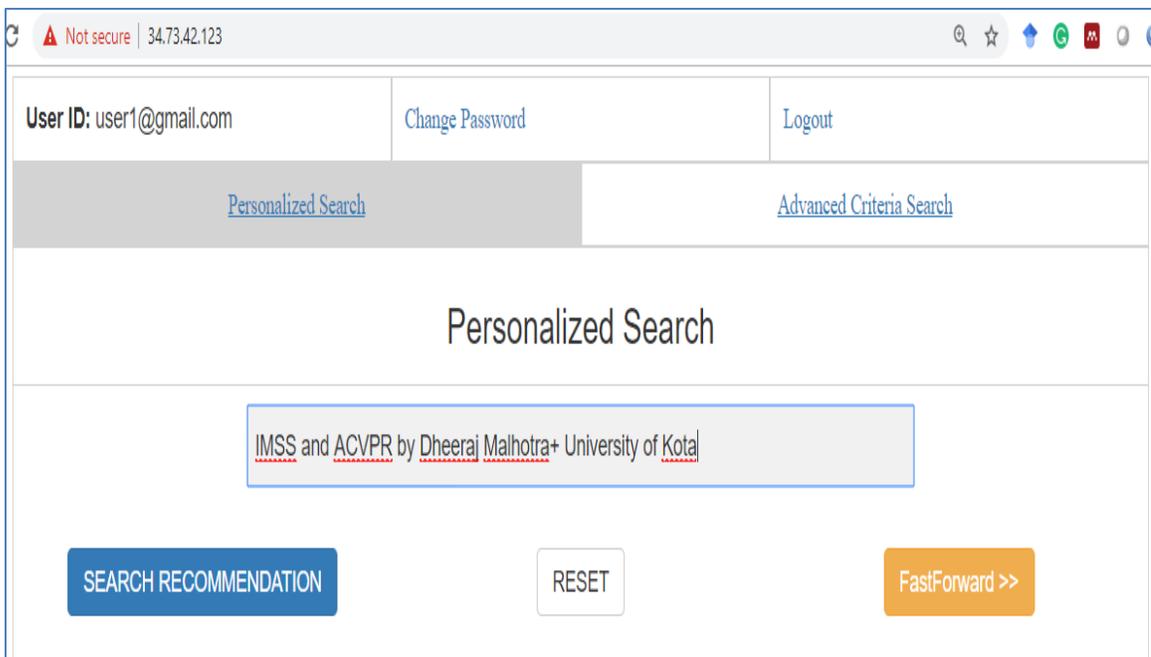


Fig. 3.7 IMSS tool interface with a specific external IP address

Overview 'localhost:9000' (active)	
Started:	Sat Mar 23 15:47:11 +0530 2019
Version:	2.8.3, rb3fe56402d908019d99af1f1f4fc65cb1d1436a2
Compiled:	Tue Dec 05 09:13:00 +0530 2017 by jdu from branch-2.8.3
Cluster ID:	CID-455ce458-86d0-4829-b7cf-e08e7c5f1b7e
Block Pool ID:	BP-170894042-10.142.0.2-1537366384610

Fig. 3.8 NameNode information of IMSS system

Configured Capacity:	19.62 GB
DFS Used:	48 KB (0%)
Non DFS Used:	10 GB
DFS Remaining:	8.71 GB (44.38%)
Block Pool Used:	48 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	Sat Mar 23 15:47:11 +0530 2019
Last Checkpoint Time	Sat Mar 23 15:03:38 +0530 2019

Fig. 3.9 NameNode- DFS cluster information on HDFS

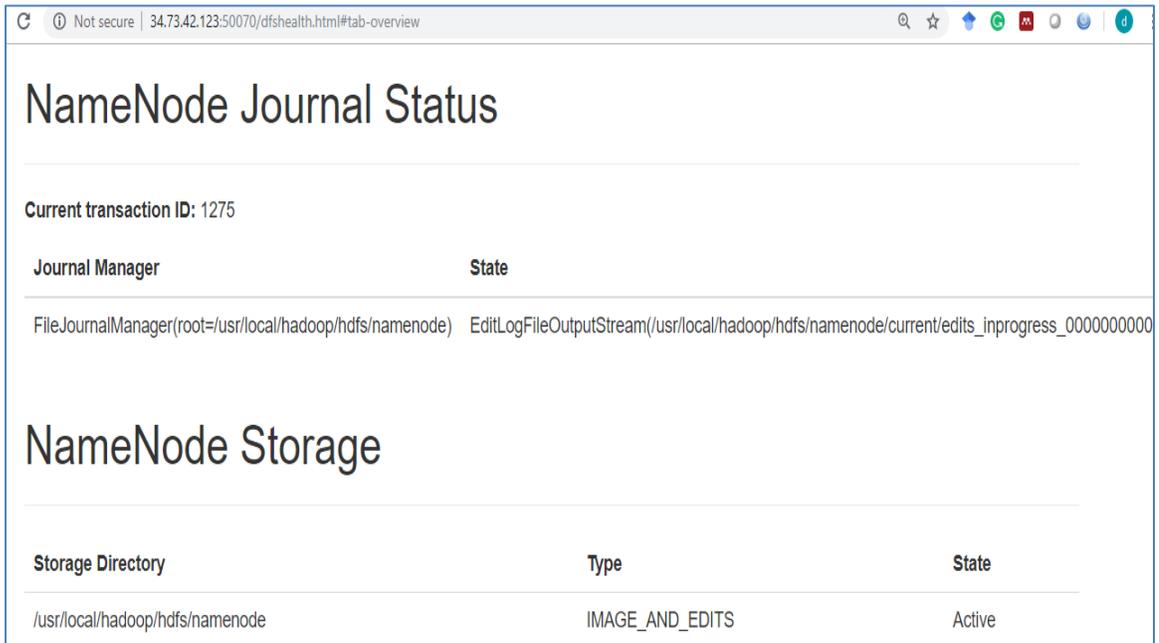


Fig. 3.10 NameNode journal status information of the IMSS tool

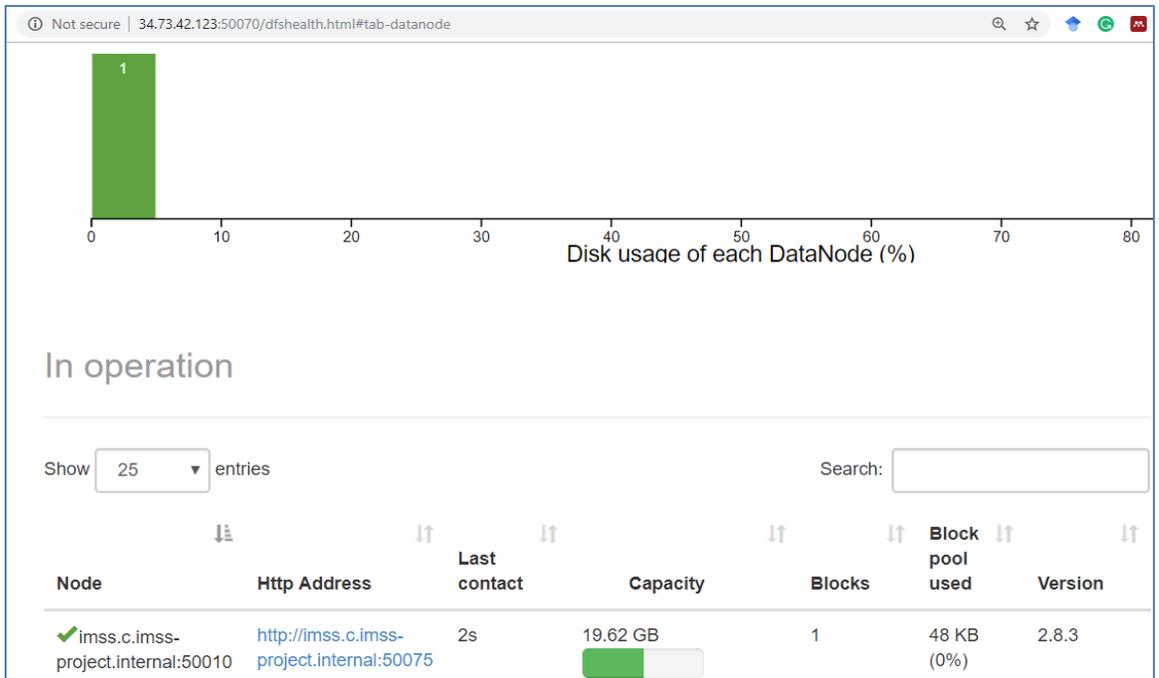


Fig. 3.11 DataNode information of the IMSS tool

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime
application_1553336258937_0007	dr.who	hadoop	YARN	default	-1	Sat Mar 23 16:08:33 +0550 2019	N/A
application_1553336258937_0006	dr.who	hadoop	YARN	default	-1	Sat Mar 23 16:08:33 +0550 2019	N/A
application_1553336258937_0005	dr.who	hadoop	YARN	default	-1	Sat Mar 23 16:08:33 +0550	N/A

Fig. 3.12 Hadoop scheduler metrics of the IMSS tool

Startup Progress

Elapsed Time: 2 sec, Percent Complete: 100%

Phase	Completion	Elapsed Time
Loading fsimage /usr/local/hadoop/hdfs/namenode/current/fsimage_000000000000001245 425 B	100%	0 sec
inodes (2/2)	100%	
delegation tokens (0/0)	100%	
cache pools (0/0)	100%	
Loading edits /usr/local/hadoop/hdfs/namenode/current/edits_000000000000001246-000000000000001267 1 MB (22/22)	100%	0 sec

Fig. 3.13 Hadoop startup progress for the IMSS tool

3.6. ADVANTAGES OF CLOUD DEPLOYMENT FOR IMSS TOOL

There are various advantages of deploying the Intelligent Meta Search System (IMSS) on the cloud based on the second generation HDFS. These advantages are discussed as follows:

1. **Elastic Scaling:** The virtual machine instances created to run IMSS application may be added or removed depending upon the load. Thus, depending on the number of links returned by background search engines will determine the number of data nodes to be used by the application and is also referred to as dynamic scaling.
2. **Resource Management:** The IMSS application is deployed on the Google Cloud Platform (GCP). Google provides the necessary virtual machine instances, network bandwidth, and other infrastructure. In case of unpredicted failures or some hardware updates, GCP will automatically locate new VMs for IMSS application. The GCP will save a lot of effort required in the form of IT skills or hardware skills necessary to successfully deploy and manage the application.
3. **Infrastructure Offloading:** The IMSS application can save a lot on GCP as it is comparatively expensive to manage its own data center, hardware infrastructure, network bandwidth and software licensing, etc. GCP allows keeping this infrastructural cost to a minimum as well as helps in implementing pay-on-demand.
4. **Search load Spikes:** The number of users of IMSS application may unexpectedly increase on a specific day. GCP based second generation HDFS deployment is an ideal platform to scale out during peak loads to handle load spikes in a user and developer friendly manner.
5. **High Reliability:** The GCP based second generation HDFS allow to maintain redundant copies of the data by keeping the same data in multiple DNs. This

concept is also popularly known as “Rack Placement Strategy” where first copy is placed randomly in any rack. The second copy is kept on a different rack. However, the third copy will be maintained on the same rack but with different DN. This redundancy in information will ensure the high availability even in the case of failure of a data node.

6. Lesser Issues: Due to backup of GCP and reliable second-generation HDFS, IMSS tool will have lower operational issues as highly professional Google experts maintain network and software infrastructure. This reliable infrastructural assurance leads to the high availability of the search tool to the end user.

3.7. CHAPTER SUMMARY

This chapter discusses various available cloud computing platforms to implement and deploy the metasearch application, that is, IMSS. This chapter carries out a detailed ranking comparison between different platforms like HDFS, SPARK, etc. The tabular ranking comparison helps in identifying HDFS as the best cloud platform to implement IMSS tool. The detailed system architecture of the IMSS is also discussed in this chapter. The system architecture is followed by various screenshots of the Hadoop cluster setup information on the Google Cloud Platform (GCP). The cluster information is followed by a detailed discussion of the advantages of choosing the second generation HDFS and GCP based deployment for current research work.

CHAPTER 4

GOOGLE CLOUD PLATFORM

4.1. INTRODUCTION

This chapter discusses the Google Cloud Platform (GCP) and various steps to demonstrate how to set up a single-node or multi-node cluster configuration for big data analytics. The role of GCP in the deployment of the recommended system architecture is also discussed in detail. The big data analytics is vital for the implementation of a metasearch application like Intelligent Meta Search System (IMSS) tool within current research work. The big data in the form of a vast number of returned links by various background search engines can be easily analyzed using Hadoop- MapReduce analytics platform in real time. This analytics, in turn, assist in quickly choosing the most relevant page ranking order to be shown to the user to best suit his or her personalized requirements. (Malhotra and Rishi, 2018b)

4.2. VIRTUAL MACHINE (VM)

The Google Cloud Platform (GCP) is an ideal platform for exploring various cloud services (Malhotra and Rishi, 2018b) required to deploy a metasearch application like Intelligent Meta Search System (IMSS) tool. The compute engine module allows the developer to create and use VM machines which are virtual copies of OS servers like Linux server and Window server. The GCP let the user choose VMs with small to a large configuration in terms of CPU cores, memory, and OS image to best suit the project requirements like that of a personalized metasearch tool. The number and configuration of VMs are dependent on the load, i.e., the number of simultaneous users of the IMSS

tool. The free trial account allows the developer to have 300\$ credit to explore various services offered by GCP. The interface and multiple functionalities offered by GCP are shown through multiple screenshots in this chapter. In Fig. 4.1, GCP interface displays details of IMSS project information such as project ID, project number, billing status, platform status to represent whether all services are normal or not. The left pane provides access to various tabs like a compute-engine to create VM instances, cloud launcher, billing, App launcher, etc. The error reporting option allows the user to get assistance for reporting various errors in cloud setup from Google expert team. The VM instances are required to be created as shown in Fig. 4.2 by choosing OS, access scope, CPU configuration, and firewall rule to allow HTTP traffic. To develop multi-node cluster setups, one needs to create multiple Virtual Machine (VM) instances.

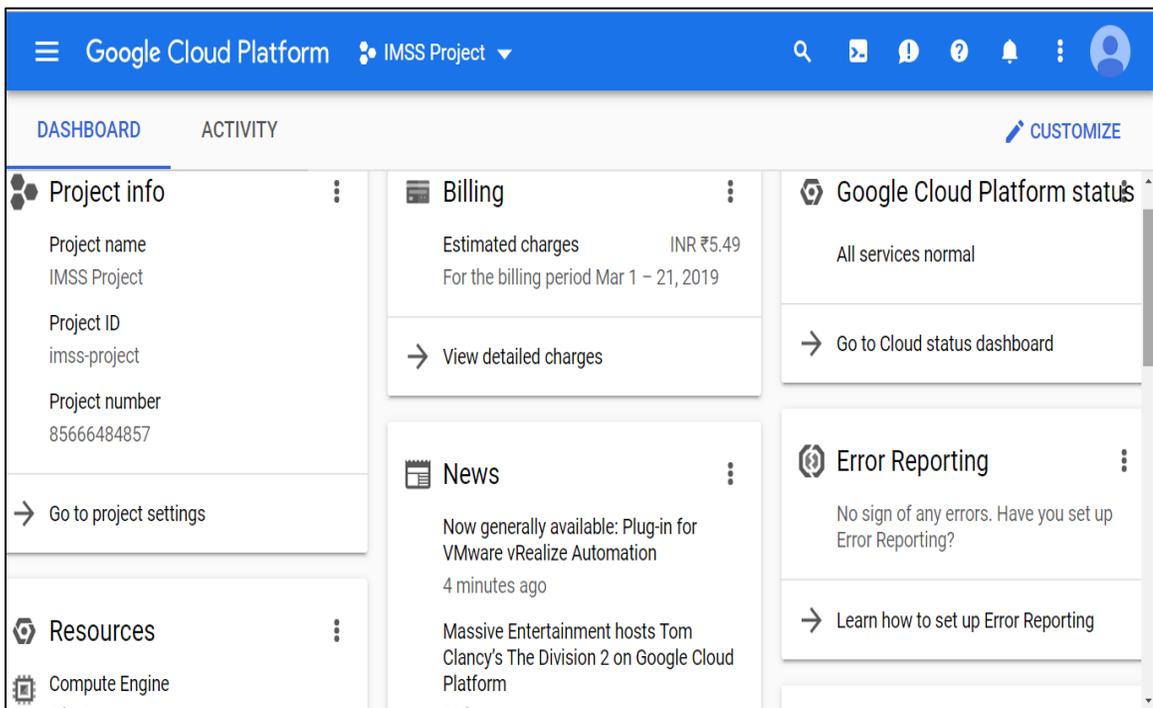


Fig. 4.1 Interface of GCP showing details of IMSS project

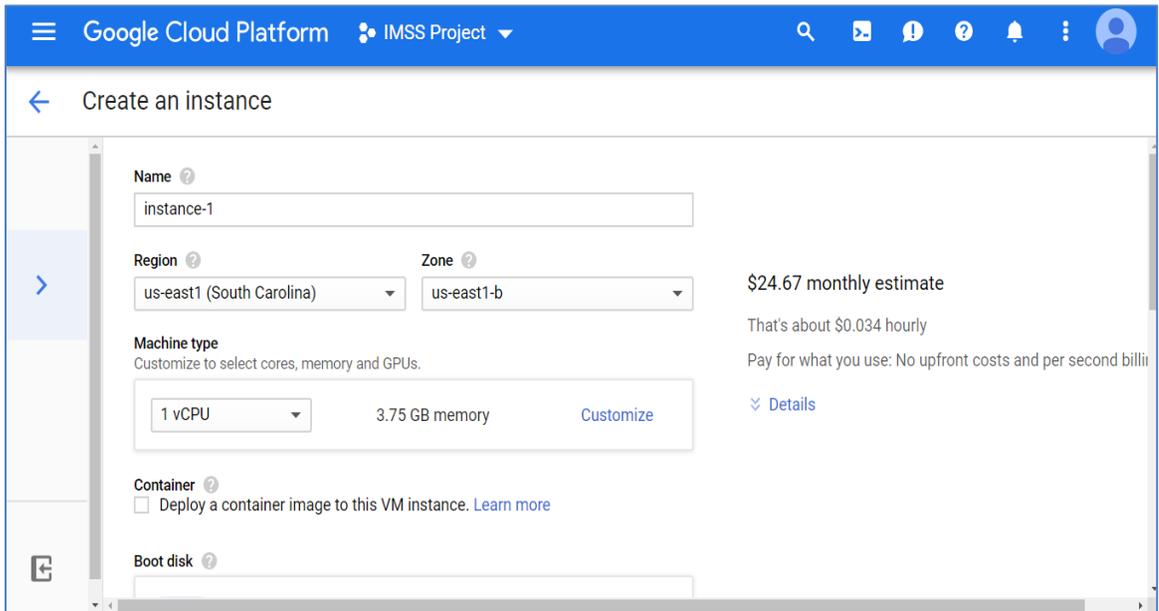


Fig. 4.2 Creation of a VM instance in GCP

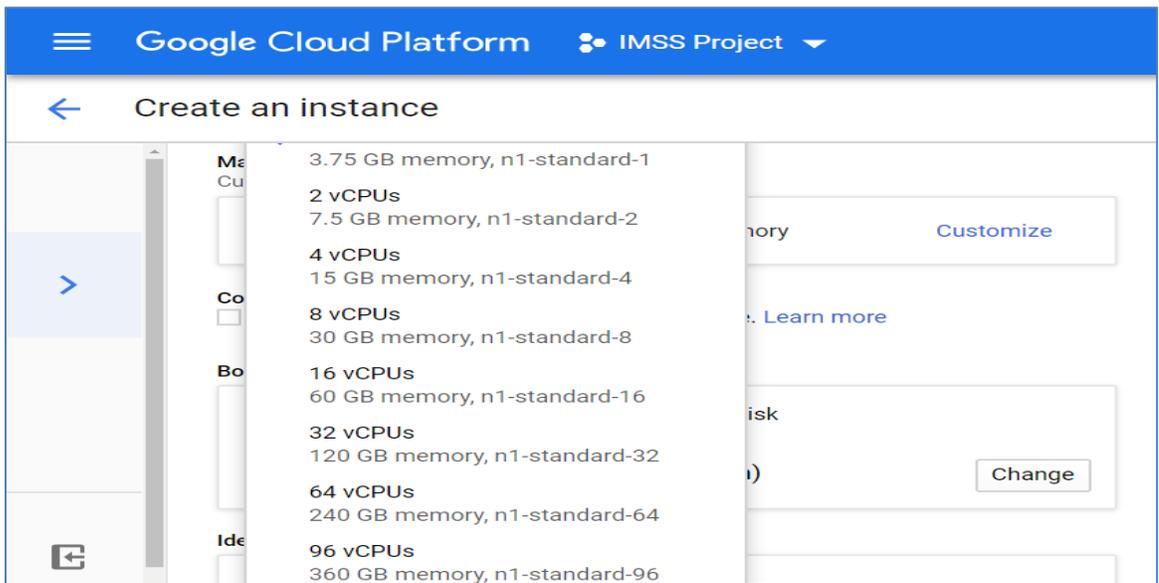


Fig. 4.3 VM configurations available in GCP

As shown in Fig. 4.3, various VM configurations varying from 1 core CPU with 3.75 GB memory to 96 cores CPU with 360 GB memory is available in GCP. Thus a developer

can choose any of these configurations for master and slave instances. The master instance is responsible for managing NameNode and DataNode. It will be responsible for providing data by distributing web links to slave nodes returned by the backend search as well as to collect processed data from each of the slave nodes through its job tracker component in a multi-node cluster setup. The number of slave instances may be created in proportion to Data Nodes required by implemented personalized metasearch application, i.e., IMSS. It may be noted that the adequate number of DataNodes should be generated depending upon the volume of the big data needed to be processed, either more or less number of DataNodes than required will severely affect the performance of the cluster and hence the efficiency of the application.

4.3. GOOGLE CLOUD CLUSTER CONFIGURATION

There are two possible ways to set up a cluster on the Google cloud as required by the current deployment of IMSS tool:

- (i) Single Node Configuration
- (ii) Multi-Node Configuration

The single node configuration is a sub-step for multi-node configuration. Hence steps to set up a single node configuration will be common to both of them. Section 4.3.1 discusses various sub-steps of the single node configuration.

4.3.1. Single Node Configuration

The single node configuration will create one VM instance known as master instance and will serve the purpose of both NameNode and DataNode (www.tutorialspoint.com [157]). The main steps for a single node cluster configuration are as follows (www.linode.com [158]):

- (i) Java Installation
- (ii) Hadoop Installation
- (iii) Configuring Environment Variables

- (iv) Configuring.XML Files
- (v) Creating Directories and Changing Ownership
- (vi) Rebooting
- (vii) RSA Key Generation and Authorization
- (viii) NameNode Cleaning and Service Verification
- (ix) Firewall Rules Settings
- (x) DFS Health Checkup

4.3.1.1. Java Installation

After clicking SSH, Google will transfer secure shell keys to the virtual machine, and as soon as the keys are transferred, the command prompt is required to execute various commands for Java installation and single node cluster setup as shown in Fig. 4.4. To install Java, firstly, *java8-Debian.list* file is created using nano editor via superuser privilege command (*sudo*) within the specified path as shown in Fig. 4.5. This process of file creation is followed by downloading the latest packages of Java from the launch pad of Personal Package Archives (PPA) repositories using *deb* command as shown in Fig. 4.6. This process of downloading packages is followed by the installation of the directory manager using superuser privilege as shown in Fig.4.7. The directory manager will assist in downloading keys with ID EEA14886 from key server to validate Java packages as shown in Fig. 4.8. This key download is followed by updating packages as shown in Fig.4.9 followed by installation of Java using superuser privilege as shown in Fig.4.10. The Java installation will lead to additional memory occupancy of 30.5 MB of space. The *yes* acknowledgment is then followed by two prompts to agree to Oracle binary code license agreement to use Oracle-Java Development Kit (JDK) and downloading .tar format of Java as shown in Fig.4.11, 4.12 and 4.13. This process is followed by accepting Java as default on the single node Linux instance under configuration as shown in Fig.4.14. The successful installation of Java may be confirmed by checking version details of the Java as shown in Fig.4.15.

```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
Connected, host fingerprint: ssh-rsa 2048 A5:F9:1A:23:1F:B2:CE:29:95:CC:25:AE:CC:
00:D9:73:F6:4C:C2:23:AC:4E:D6:71:21:8E:61:54:DF:59:79:42
Linux master 4.9.0-6-amd64 #1 SMP Debian 4.9.88-1+deb9u1 (2018-05-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
metasearchtool@master:~$
```

Fig. 4.4 Secure shell (SSH) command prompt

```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$
metasearchtool@master:~$ sudo nano /etc/apt/sources.list.d/java8-debian.list
```

Fig. 4.5 Creation of *java8-debian.list*

```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
GNU nano 2.7.4 File: /etc/apt/sources.list.d/java8-debian.list Modified

deb http://ppa.launchpad.net/webupd8team/java/ubuntu trusty main
deb-src http://ppa.launchpad.net/webupd8team/java/ubuntu trusty main

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page M-/_ First Line
^X Exit ^R Read File ^A Replace ^U Uncut Text ^T To Spell ^G Go To Line ^V Next Page M-| Last Line
```

Fig. 4.6 Downloading Java packages from PPA repositories

```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$ sudo apt-get install dirmngr
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  dbus-user-session libpam-systemd pinentry-gnome3 tor
The following NEW packages will be installed:
  dirmngr
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 595 kB of archives.
After this operation, 1,110 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian stretch/main amd64 dirmngr amd64 2.1.18-8-deb9u1 [595 kB]
Fetched 595 kB in 0s (4,173 kB/s)
Selecting previously unselected package dirmngr.
(Reading database ... 32393 files and directories currently installed.)
Preparing to unpack .../dirmngr_2.1.18-8-deb9u1_amd64.deb ...
Unpacking dirmngr (2.1.18-8-deb9u1) ...
Processing triggers for man-db (2.7.6.1-2) ...
```

Fig. 4.7 Installation of the directory manager

```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$
metasearchtool@master:~$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys EEA14886

Executing: /tmp/apt-key-gpghome.RVtwfGgTfi/gpg.1.sh --keyserver keyserver.ubuntu.com --recv-keys EEA14886

gpg: key C2518248EEA14886: public key "Launchpad VLC" imported
gpg: Total number processed: 1
gpg:    imported: 1
metasearchtool@master:~$
metasearchtool@master:~$
metasearchtool@master:~$
metasearchtool@master:~$
metasearchtool@master:~$
metasearchtool@master:~$
metasearchtool@master:~$
```

Fig. 4.8 Downloading key with ID EEA14886

```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$
metasearchtool@master:~$
metasearchtool@master:~$ sudo apt-get update
Ign:1 http://deb.debian.org/debian stretch InRelease
Get:2 http://security.debian.org stretch/updates InRelease [94.3 kB]
Get:3 http://deb.debian.org/debian stretch-updates InRelease [91.0 kB]
Get:4 http://packages.cloud.google.com/apt cloud-sdk-stretch InRelease [6,377 B]
Get:5 http://deb.debian.org/debian stretch-backports InRelease [91.8 kB]
Hit:6 http://deb.debian.org/debian stretch Release
Get:7 http://packages.cloud.google.com/apt google-compute-engine-stretch-stable InRelease [3,843 B]
Get:8 http://ppa.launchpad.net/webupd8team/java/ubuntu trusty InRelease [15.5 kB]
Hit:9 http://packages.cloud.google.com/apt google-cloud-packages-archive-keyring-stretch InRelease
Get:10 http://packages.cloud.google.com/apt cloud-sdk-stretch/main amd64 Packages [37.9 kB]
Get:11 http://security.debian.org stretch/updates/main Sources [142 kB]
Get:12 http://security.debian.org stretch/updates/main amd64 Packages [356 kB]
Get:13 http://security.debian.org stretch/updates/main Translation-en [166 kB]
Get:14 http://deb.debian.org/debian stretch-backports/main Sources.diff/Index [27.8 kB]
Get:15 http://deb.debian.org/debian stretch-backports/main amd64 Packages.diff/Index [27.8 kB]
Get:16 http://deb.debian.org/debian stretch-backports/main Translation-en.diff/Index [27.8 kB]
Ign:14 http://deb.debian.org/debian stretch-backports/main Sources.diff/Index
Get:18 http://deb.debian.org/debian stretch-backports/main amd64 Packages 2018-05-11-0221.47.pdiff [315 B]
Get:19 http://packages.cloud.google.com/apt google-compute-engine-stretch-stable/main amd64 Packages [1,126 B]
Get:20 http://deb.debian.org/debian stretch-backports/main amd64 Packages 2018-05-11-0820.55.pdiff [1,824 B]
Get:21 http://deb.debian.org/debian stretch-backports/main amd64 Packages 2018-05-11-1423.59.pdiff [242 B]
Get:22 http://deb.debian.org/debian stretch-backports/main amd64 Packages 2018-05-12-0821.22.pdiff [309 B]
Get:23 http://deb.debian.org/debian stretch-backports/main amd64 Packages 2018-05-12-2023.51.pdiff [1,149 B]
Get:24 http://ppa.launchpad.net/webupd8team/java/ubuntu trusty/main Sources [676 B]
Get:25 http://deb.debian.org/debian stretch-backports/main amd64 Packages 2018-05-13-0220.41.pdiff [229 B]
Get:26 http://deb.debian.org/debian stretch-backports/main amd64 Packages 2018-05-14-2009.11.pdiff [332 B]
```

Fig. 4.9 Updating Java packages

```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$
metasearchtool@master:~$ sudo apt-get install oracle-java8-installer
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils gsfonstools gsfonstools-x11 java-common libfontenc1 libxfont1 oracle-java8-set-default x11-common xfonts-encodings
  xfonts-utils
Suggested packages:
  binutils-doc binfmt-support visualvm ttf-baekmuk | ttf-unfonts | ttf-unfonts-core ttf-kochi-gothic | ttf-sazanami-gothic
  ttf-kochi-mincho | ttf-sazanami-mincho ttf-arphic-uming firefox | firefox-2 | iceweasel | mozilla-firefox
  | iceape-browser | mozilla-browser | epiphany-gecko | epiphany-webkit | epiphany-browser | galeon | midbrowser
  | moblin-web-browser | xulrunner | xulrunner-1.9 | konqueror | chromium-browser | midori | google-chrome
The following NEW packages will be installed:
  binutils gsfonstools gsfonstools-x11 java-common libfontenc1 libxfont1 oracle-java8-installer oracle-java8-set-default
  x11-common xfonts-encodings xfonts-utils
0 upgraded, 11 newly installed, 0 to remove and 5 not upgraded.
Need to get 8,021 kB of archives.
After this operation, 30.5 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Fig. 4.10 Command to install Java

```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
Package configuration
Configuring oracle-java8-installer
Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX
You MUST agree to the license available in http://java.com/license if you want to use Oracle JDK.
<Ok>
```

Fig. 4.11 Oracle JDK license agreement

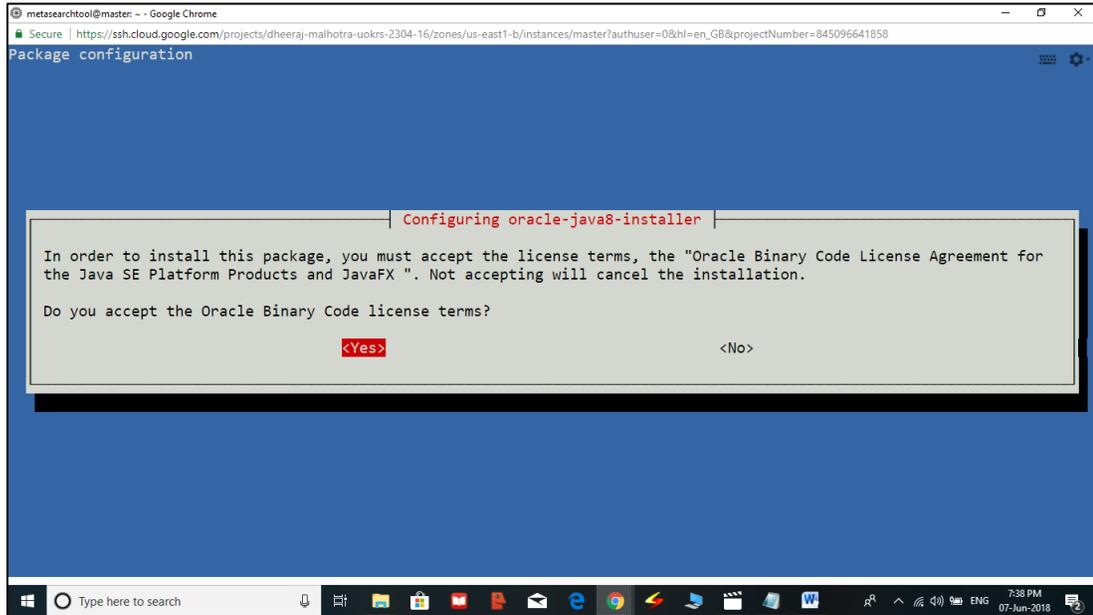


Fig. 4.12 Oracle binary code license agreement

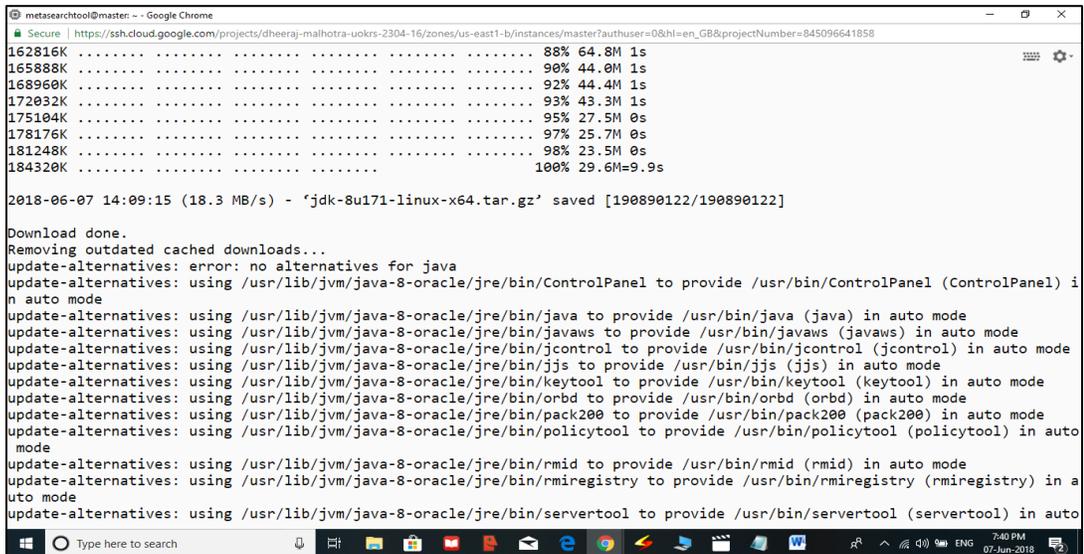
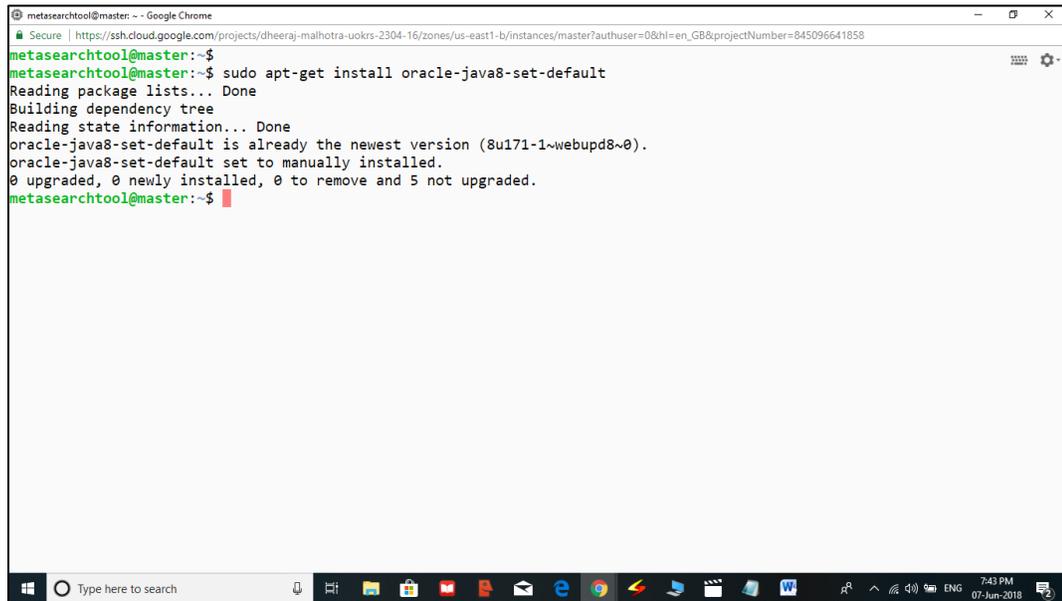
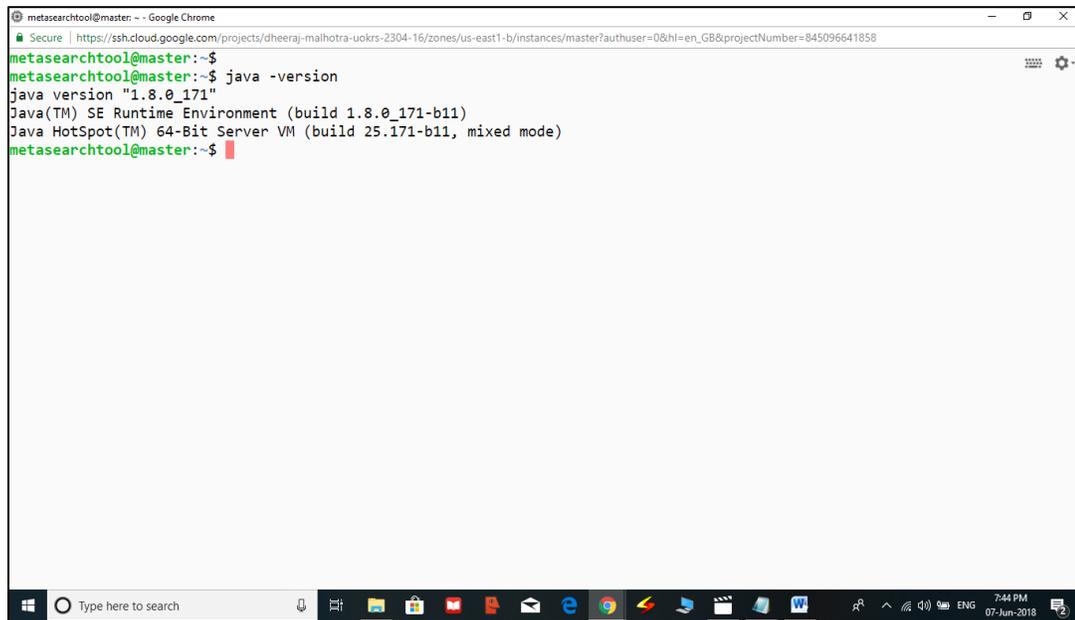


Fig. 4.13 Downloading Java in .tar format



```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$
metasearchtool@master:~$ sudo apt-get install oracle-java8-set-default
Reading package lists... Done
Building dependency tree
Reading state information... Done
oracle-java8-set-default is already the newest version (8u171-1~webupd8~0).
oracle-java8-set-default set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
metasearchtool@master:~$
```

Fig. 4.14 Setting Java as default

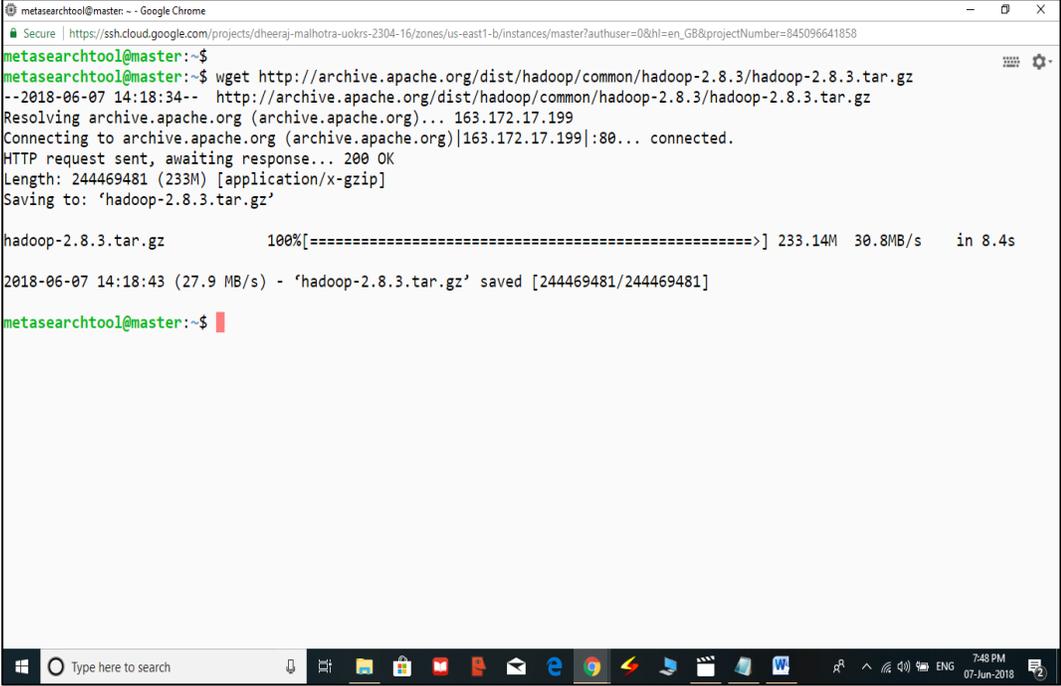


```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$
metasearchtool@master:~$ java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
metasearchtool@master:~$
```

Fig. 4.15 Checking installed version of Java

4.3.1.2. Hadoop Installation

The next step is to install Hadoop from Apache- Hadoop website. The binary link of the most stable version of Hadoop is required to be checked as available on the site. If the desired release is not available, then the archive links may be considered for installation. The tar version of Hadoop-2.8.3 can be downloaded using *wget* command as shown in Fig. 4.16. The tar/.gz version can then be extracted using *tar* command via superuser privilege as shown in Fig. 4.17. The extracted files are required to be copied to */usr/local/hadoop* to ensure that the same path could be used for configurations as shown in Fig. 4.18. The *ls* command can be used to check various files copied to the desired location, i.e., */usr/local/Hadoop*.

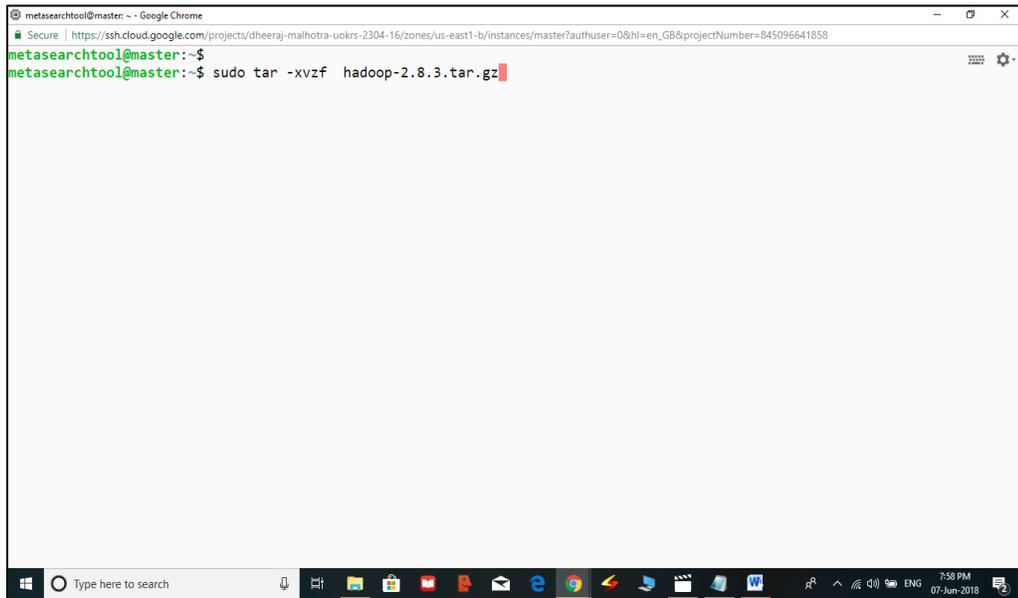


```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$
metasearchtool@master:~$ wget http://archive.apache.org/dist/hadoop/common/hadoop-2.8.3/hadoop-2.8.3.tar.gz
--2018-06-07 14:18:34-- http://archive.apache.org/dist/hadoop/common/hadoop-2.8.3/hadoop-2.8.3.tar.gz
Resolving archive.apache.org (archive.apache.org)... 163.172.17.199
Connecting to archive.apache.org (archive.apache.org)|163.172.17.199|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 244469481 (233M) [application/x-gzip]
Saving to: 'hadoop-2.8.3.tar.gz'

hadoop-2.8.3.tar.gz      100%[=====] 233.14M  30.8MB/s   in 8.4s
2018-06-07 14:18:43 (27.9 MB/s) - 'hadoop-2.8.3.tar.gz' saved [244469481/244469481]

metasearchtool@master:~$
```

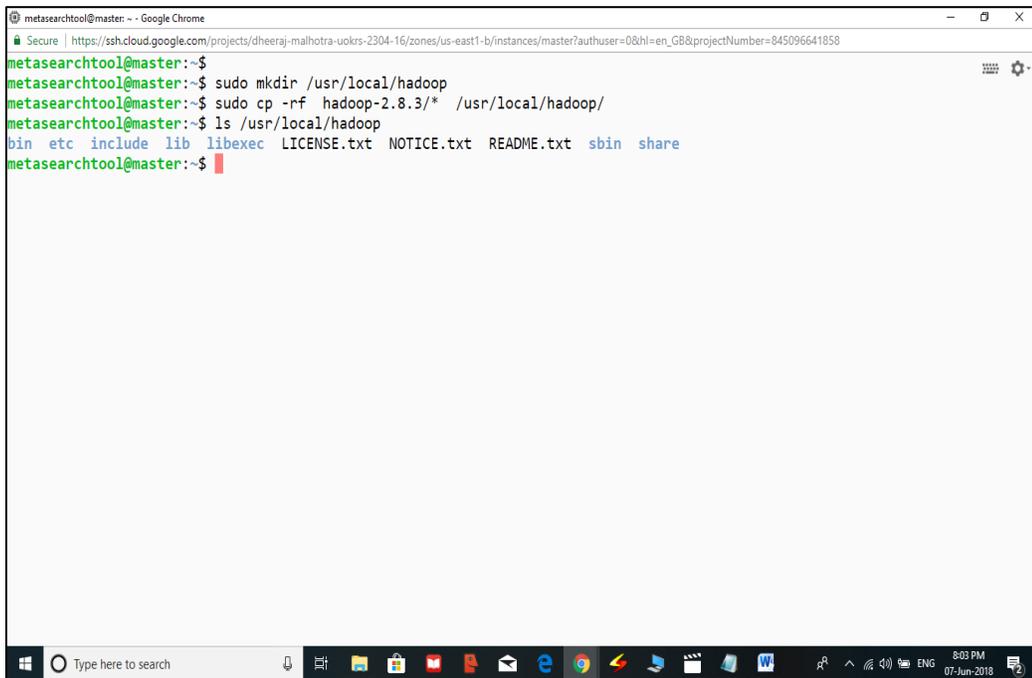
Fig. 4.16 Downloading Hadoop-2.8.3



A terminal window titled "metasearchtool@master" is shown. The terminal prompt is "metasearchtool@master:~\$". The user has entered the command "sudo tar -xvzf hadoop-2.8.3.tar.gz". The terminal output is empty, indicating the extraction process is either in progress or has just finished. The window title bar shows "metasearchtool@master - Google Chrome" and the address bar shows "Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858". The taskbar at the bottom shows various application icons and the system tray with the time "7:58 PM" and date "07-Jun-2018".

```
metasearchtool@master:~$ sudo tar -xvzf hadoop-2.8.3.tar.gz
```

Fig. 4.17 Extracting Hadoop



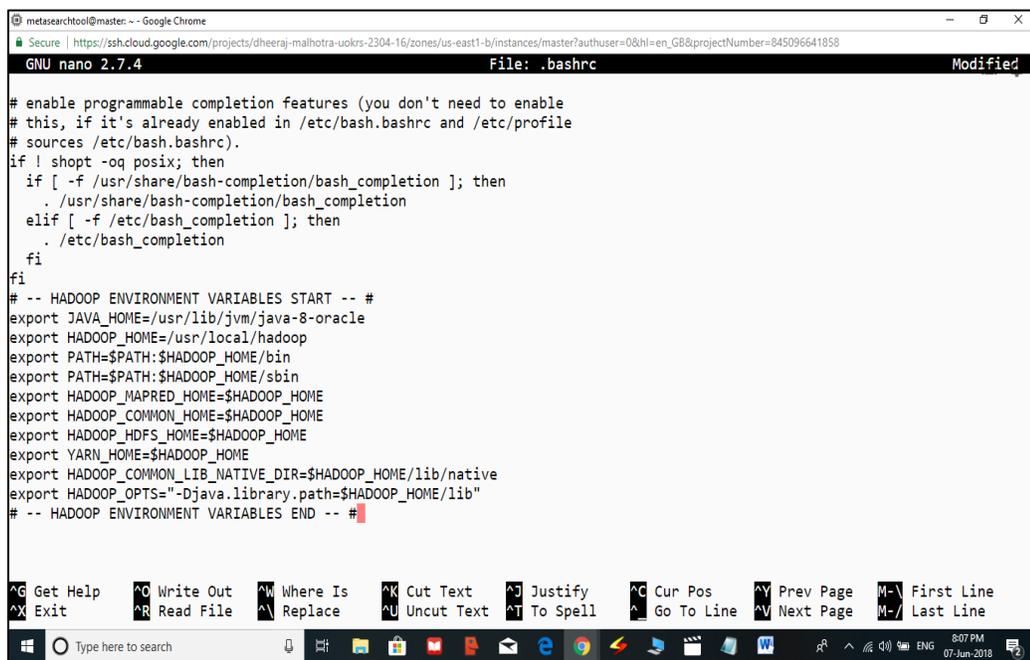
A terminal window titled "metasearchtool@master" is shown. The terminal prompt is "metasearchtool@master:~\$". The user has entered the following commands: "sudo mkdir /usr/local/hadoop", "sudo cp -rf hadoop-2.8.3/* /usr/local/hadoop/", and "ls /usr/local/hadoop". The terminal output shows the directory listing: "bin etc include lib libexec LICENSE.txt NOTICE.txt README.txt sbin share". The window title bar shows "metasearchtool@master - Google Chrome" and the address bar shows "Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858". The taskbar at the bottom shows various application icons and the system tray with the time "8:03 PM" and date "07-Jun-2018".

```
metasearchtool@master:~$ sudo mkdir /usr/local/hadoop
metasearchtool@master:~$ sudo cp -rf hadoop-2.8.3/* /usr/local/hadoop/
metasearchtool@master:~$ ls /usr/local/hadoop
bin  etc  include  lib  libexec  LICENSE.txt  NOTICE.txt  README.txt  sbin  share
metasearchtool@master:~$
```

Fig. 4.18 Copying Hadoop to /usr/local/hadoop

4.3.1.3. Configuring Environment Variables

After Java and Hadoop installation, the next step is to configure various Hadoop environment variables. This process of configuration starts by copying different environment variables like HADOOP_HOME, JAVA_HOME, YARN_HOME, etc. with their appropriate path at the bottom of the .bashrc file as shown in Fig. 4.19. The path is shown for JAVA_HOME in Fig. 4.19 is required to be copied and to be overwritten into *hadoop-env.sh* file using superuser privilege as demonstrated in Fig. 4.20. This process of copying and overwriting is necessary so that Hadoop can determine the location of Java and can access it for execution.



```
metasearchtool@master: -- Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096541858
GNU nano 2.7.4 File: .bashrc Modified

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
# -- HADOOP ENVIRONMENT VARIABLES START -- #
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
# -- HADOOP ENVIRONMENT VARIABLES END -- #

^G Get Help ^O Write Out ^W Where Is ^X Cut Text ^J Justify ^C Cur Pos ^Y Prev Page ^M First Line
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line ^V Next Page ^- Last Line
Type here to search 8:07 PM 07-Jun-2018
```

Fig. 4.19 Copying environment variables to .bashrc file

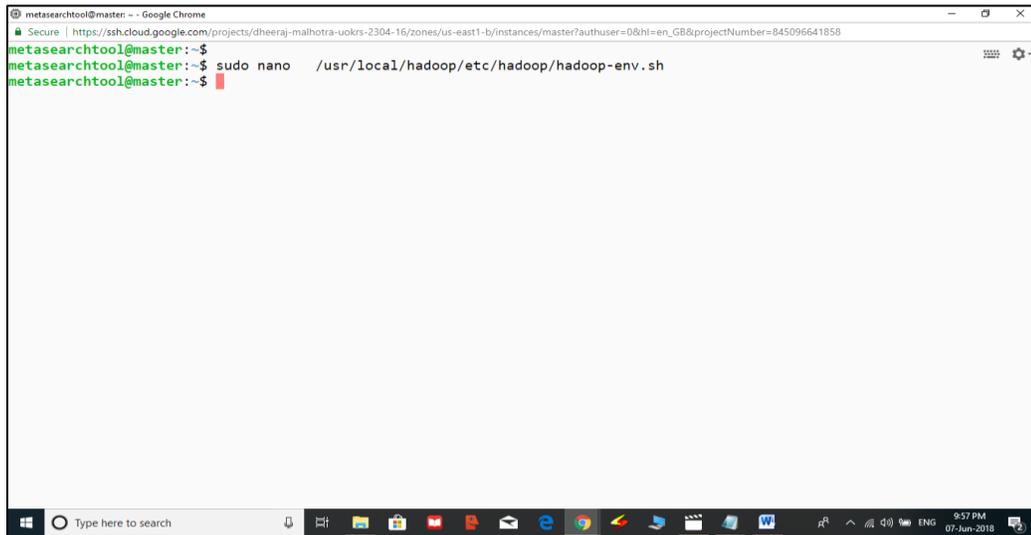
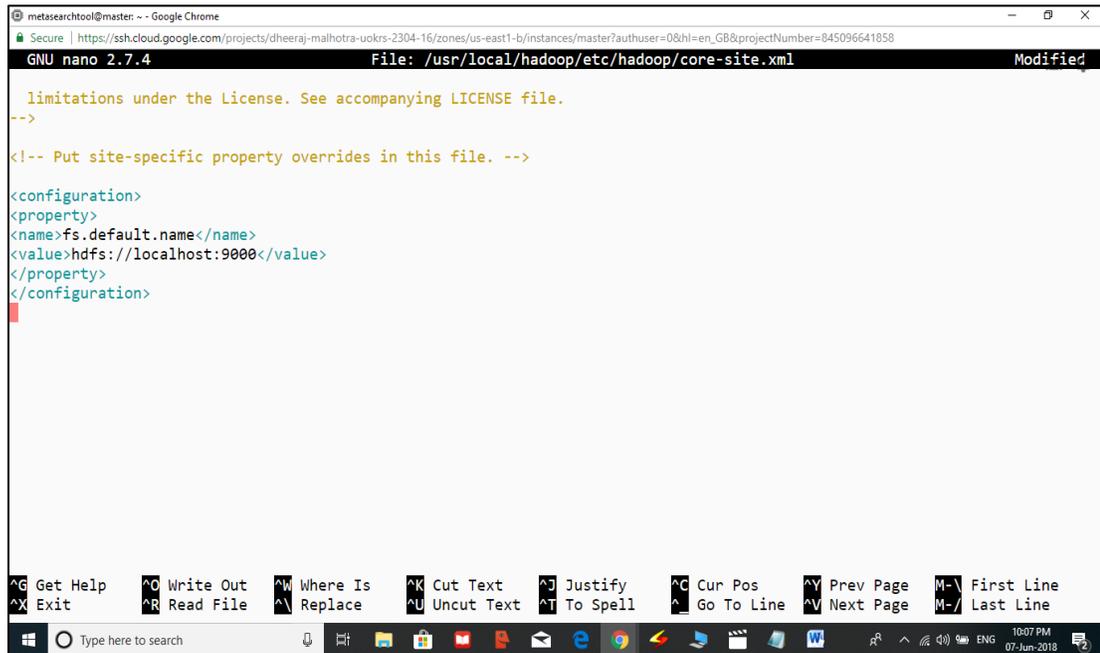


Fig. 4.20 Editing JAVA_HOME variable in hadoop-env.sh file

4.3.1.4. Configuring .XML Files

After configuring environment variables, the next step is to configure various .XML files with superuser privilege. The files such as *mapred-site.xml*, *hdfs-site.xml*, *core-site.xml*, and *yarn-site.xml* are required to be configured. The *core-site.xml* is required to update <configuration> tag with a <property> tag to specify the port number of localhost: 9000 as shown in Fig. 4.21. The *hdfs-site.xml* is required to be updated with <property> tag specifying replication factor and path of the NameNode and DataNode as shown in Fig. 4.22. The *yarn-site.xml* is required to be updated with a <property> tag specifying necessary information for big data processing via the MapReduce framework in HDFS environment as highlighted in Fig. 4.23. The *mapred-site.xml* is not available within /usr/local/hadoop path. Instead, a template file is available. So, it is first required to generate required .XML file from the template file as shown in Fig. 4.24. This process of generation of the file is followed by adding a <property> tag within newly generated *mapred-site.xml* for assigning yarn value to the map-reduce framework as shown in Fig. 4.25.



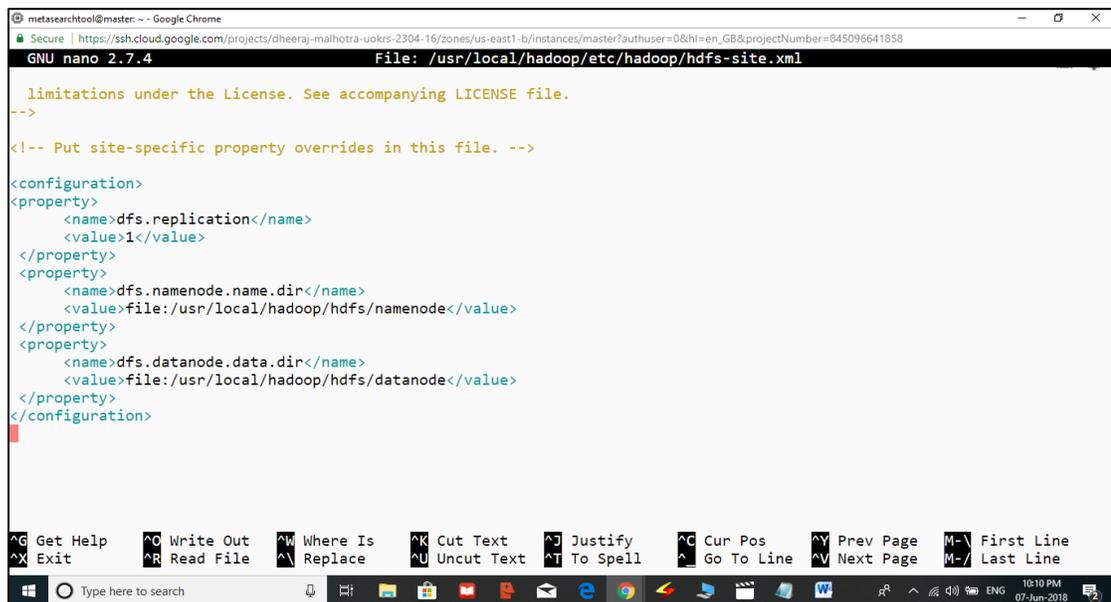
The screenshot shows the nano 2.7.4 editor in a terminal window. The file being edited is /usr/local/hadoop/etc/hadoop/core-site.xml. The content of the file is as follows:

```
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

Fig. 4.21 Configuring core-site.xml



The screenshot shows the nano 2.7.4 editor in a terminal window. The file being edited is /usr/local/hadoop/etc/hadoop/hdfs-site.xml. The content of the file is as follows:

```
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:/usr/local/hadoop/hdfs/namenode</value>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:/usr/local/hadoop/hdfs/datanode</value>
</property>
</configuration>
```

Fig. 4.22 Configuring hdfs-site.xml

```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
GNU nano 2.7.4 File: /usr/local/hadoop/etc/hadoop/yarn-site.xml

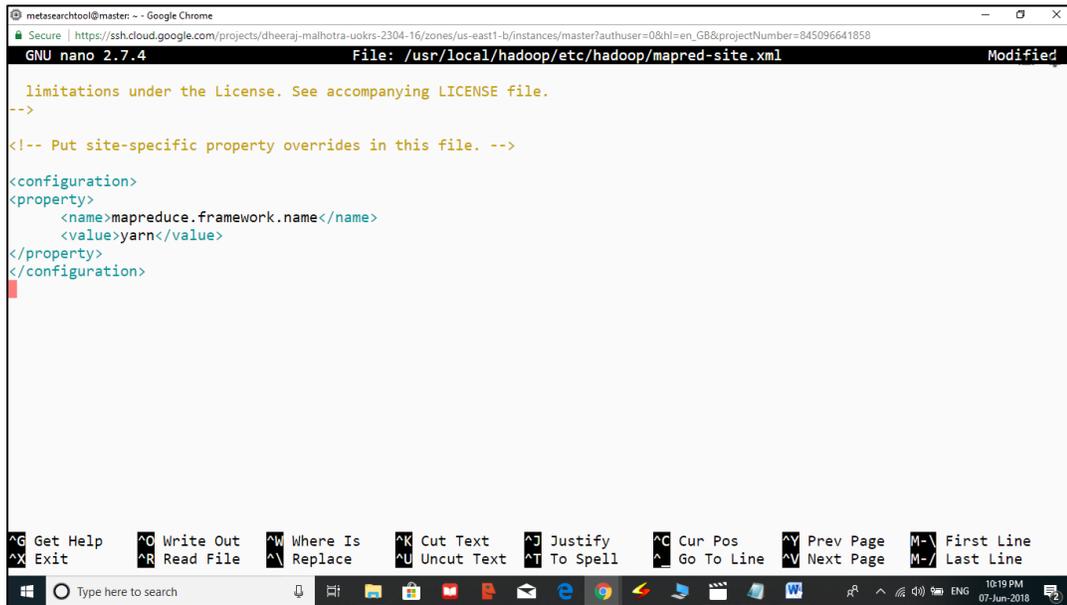
-->
<configuration>

<!-- Site specific YARN configuration properties -->
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
```

Fig. 4.23 Configuring yarn-site.xml

```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$
metasearchtool@master:~$ sudo cp -rf /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop
/mapred-site.xml
metasearchtool@master:~$ sudo nano /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

Fig. 4.24 Generating mapred-site.xml



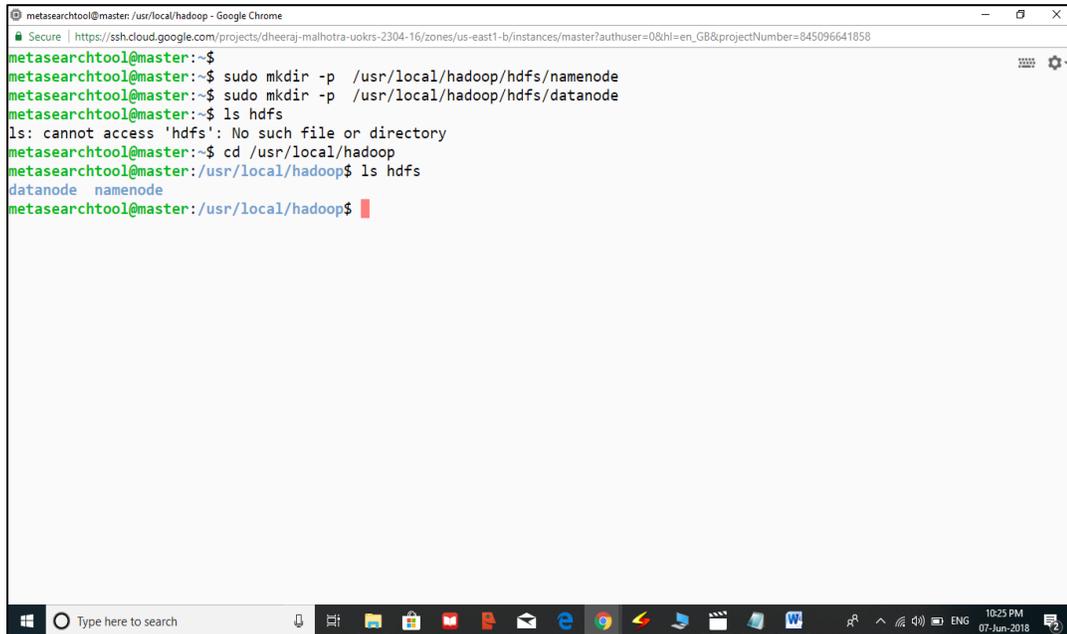
```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
GNU nano 2.7.4 File: /usr/local/hadoop/etc/hadoop/mapred-site.xml Modified

  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
</configuration>
```

Fig. 4.25 Configuring mapred-site.xml



```
metasearchtool@master: /usr/local/hadoop - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$
metasearchtool@master:~$ sudo mkdir -p /usr/local/hadoop/hdfs/namenode
metasearchtool@master:~$ sudo mkdir -p /usr/local/hadoop/hdfs/datanode
metasearchtool@master:~$ ls hdfs
ls: cannot access 'hdfs': No such file or directory
metasearchtool@master:~$ cd /usr/local/hadoop
metasearchtool@master:~/usr/local/hadoop$ ls hdfs
datanode namenode
metasearchtool@master:~/usr/local/hadoop$
```

Fig. 4.26 Creation of NameNode and DataNode directory

4.3.1.5. Creating Directories and Changing Ownership

The NameNode and DataNode directories are required to be created for setting various configurations to continue single node setup. The existence of these directories can be verified by using *ls hdfs* command within a specified path */usr/local/hadoop* as shown in Fig. 4.26.

It may be noted that all directories are accessed via '*root*' using superuser privilege as illustrated in Fig. 4.27. However, as shown in Fig. 4.28, change in ownership to metasearch tool from the root is required so that the Hadoop directory can be locally accessed without the intervention of root. The *chown-R* command is used to change ownership, and the *ls-l* command is used to verify the owner of Hadoop subdirectory within */usr/local* path.

4.3.1.6. Rebooting

After performing all the above five steps, there is a need to reboot the machine so that all configuration changes can take place effectively. This process can be accomplished through rebooting as demonstrated in Fig. 4.29. It may be noted that rebooting is mandatory to execute various commands on the Hadoop platform such as *hdfs*, *hadoop*, etc.

```

metasearchtool@master: /usr/local/hadoop - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master: /usr/local/hadoop$ ls -l
total 152
drwxr-sr-x 2 root staff 4096 Jun  7 14:32 bin
drwxr-sr-x 3 root staff 4096 Jun  7 14:32 etc
drwxr-sr-x 4 root staff 4096 Jun  7 16:53 hdfs
drwxr-sr-x 2 root staff 4096 Jun  7 14:32 include
drwxr-sr-x 3 root staff 4096 Jun  7 14:32 lib
drwxr-sr-x 2 root staff 4096 Jun  7 14:32 libexec
-rw-r--r-- 1 root staff 99253 Jun  7 14:32 LICENSE.txt
-rw-r--r-- 1 root staff 15915 Jun  7 14:32 NOTICE.txt
-rw-r--r-- 1 root staff 1366 Jun  7 14:32 README.txt
drwxr-sr-x 2 root staff 4096 Jun  7 14:32 sbin
drwxr-sr-x 4 root staff 4096 Jun  7 14:32 share
metasearchtool@master: /usr/local/hadoop$

```

Fig. 4.27 Root ownership to subdirectories within /usr/local/Hadoop

```

metasearchtool@master: /usr/local - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master: /usr/local/hadoop$ cd ..
metasearchtool@master: /usr/local$ sudo chown -R metasearchtool /usr/local/hadoop/
metasearchtool@master: /usr/local$ ls -l
total 36
drwxrwsr-x 2 root      staff 4096 May 10 23:26 bin
drwxrwsr-x 2 root      staff 4096 May 10 23:26 etc
drwxrwsr-x 2 root      staff 4096 May 10 23:26 games
drwxr-sr-x 10 metasearchtool staff 4096 Jun  7 16:53 hadoop
drwxrwsr-x 2 root      staff 4096 May 10 23:26 include
drwxrwsr-x 4 root      staff 4096 May 10 23:28 lib
lrwxrwxrwx 1 root      staff  9 May 10 23:26 man -> share/man
drwxrwsr-x 2 root      staff 4096 May 10 23:29 sbin
drwxrwsr-x 4 root      staff 4096 May 10 23:28 share
drwxrwsr-x 2 root      staff 4096 May 10 23:26 src
metasearchtool@master: /usr/local$

```

Fig. 4.28 Ownership change from root to metasearch tool

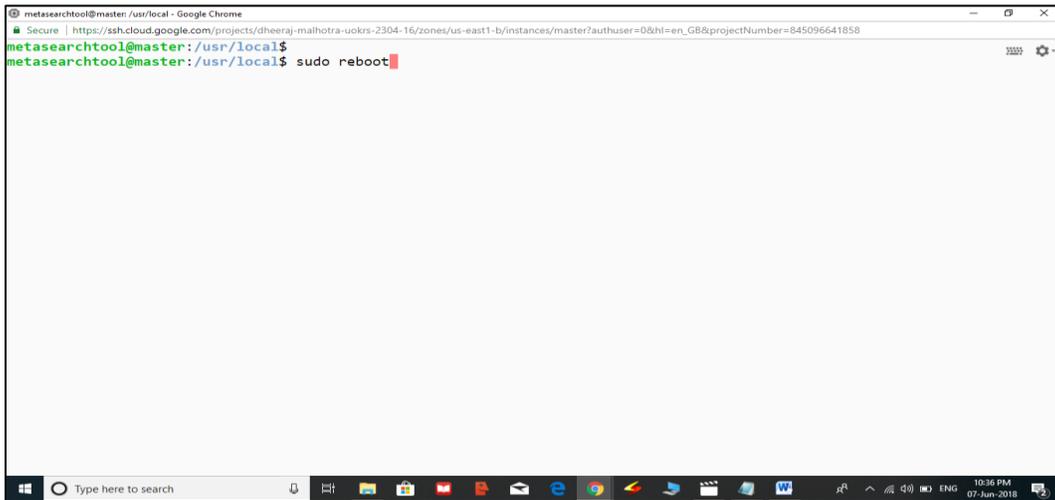


Fig. 4.29 Command to reboot the machine

4.3.1.7. RSA Key Generation and Authorization

To authorize SSH access, the *RSA* key is required to be generated followed by authorization on the local machine. The commands for generation and authorization of keys is as shown below in Fig. 4.30 and Fig. 4.31 respectively. The random art image of the generated key will be shown as the acknowledgment after the execution of the command. The public key will be saved in `/home/metasearchtool/.ssh/id_rsa` path.

```
metasearchtool@master:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/metasearchtool/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/metasearchtool/.ssh/id_rsa.
Your public key has been saved in /home/metasearchtool/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:mwbIZtnMJ7p9tAQqTJXFrLWxMjYUi/Jw+22TQ6mA/2M metasearchtool@master
The key's randomart image is:
+---[RSA 2048]---+
  .*.
  .+. =
  o oo.o +
  *.o**+.
  .o+.*=O.S
  .o.= =oo
  ..+*o+.
  .E+ +o
  .o...
+-----[SHA256]-----+
metasearchtool@master:~$
```

Fig. 4.30 RSA key generation

```
metasearchtool@master:~$ sudo cat .ssh/id_rsa.pub >> .ssh/authorized_keys
metasearchtool@master:~$
```

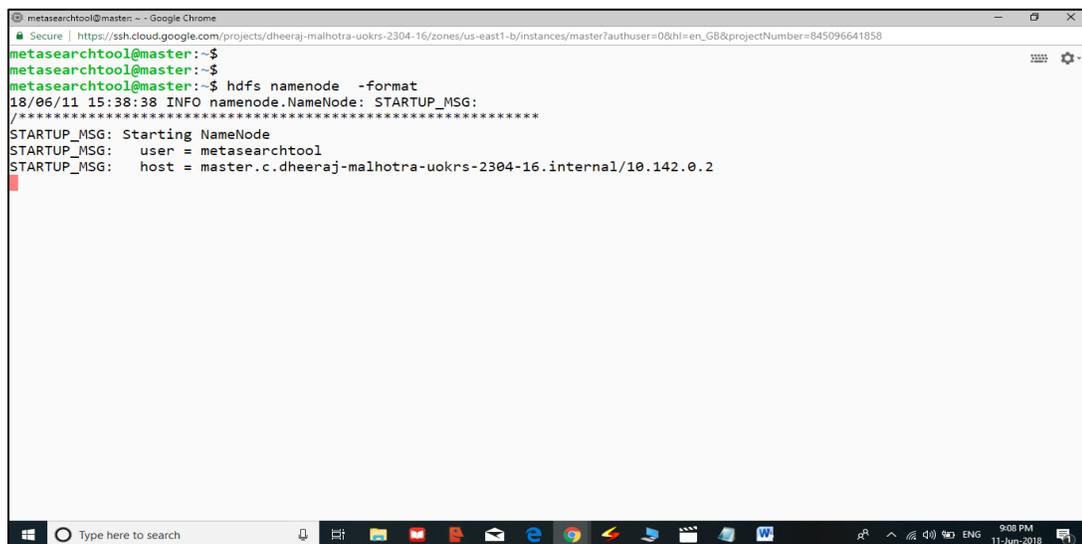
Fig. 4.31 RSA key authorization

4.3.1.8. NameNode Cleaning and Service Verification

To access the DFS, NameNode is required to be formatted using the command as shown in Fig. 4.32. After formatting, various services like `dfs.sh`, `yarn.sh`, etc. may be initiated to start the services like NameNode, DataNode, Secondary NameNode, etc. The command `start-all.sh` may be used to start all services by a single command. The `jps` command may be used to verify the status of running services as highlighted in Fig. 4.33.

4.3.1.9. Firewall Rule Settings

To add firewall rules, edit tab on the top of the Google Cloud Platform (GCP) is required to be accessed as shown in Fig. 4.34. The presented interface then may be used to edit various fields to set the name of the firewall; targets, i.e., all instances in the network; source IP, i.e., 0.0.0.0/0; protocol and ports, i.e., allow all, etc.



```
metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$
metasearchtool@master:~$
metasearchtool@master:~$ hdfs namenode -format
18/06/11 15:38:38 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: user = metasearchtool
STARTUP_MSG: host = master.c.dheeraj-malhotra-uokrs-2304-16.internal/10.142.0.2
```

Fig. 4.32 NameNode formatting

```
metasearchtool@master:~$ start-dfs.sh
18/06/11 15:47:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java c
lasses where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-metasearchtool-namenode-master.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-metasearchtool-datanode-master.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-metasearchtool-secondarynamenode-master.out
18/06/11 15:48:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java c
lasses where applicable
metasearchtool@master:~$ jps
1584 SecondaryNameNode
1714 Jps
1396 DataNode
1301 NameNode
metasearchtool@master:~$
```

Fig. 4.33 Commands to start and verify services

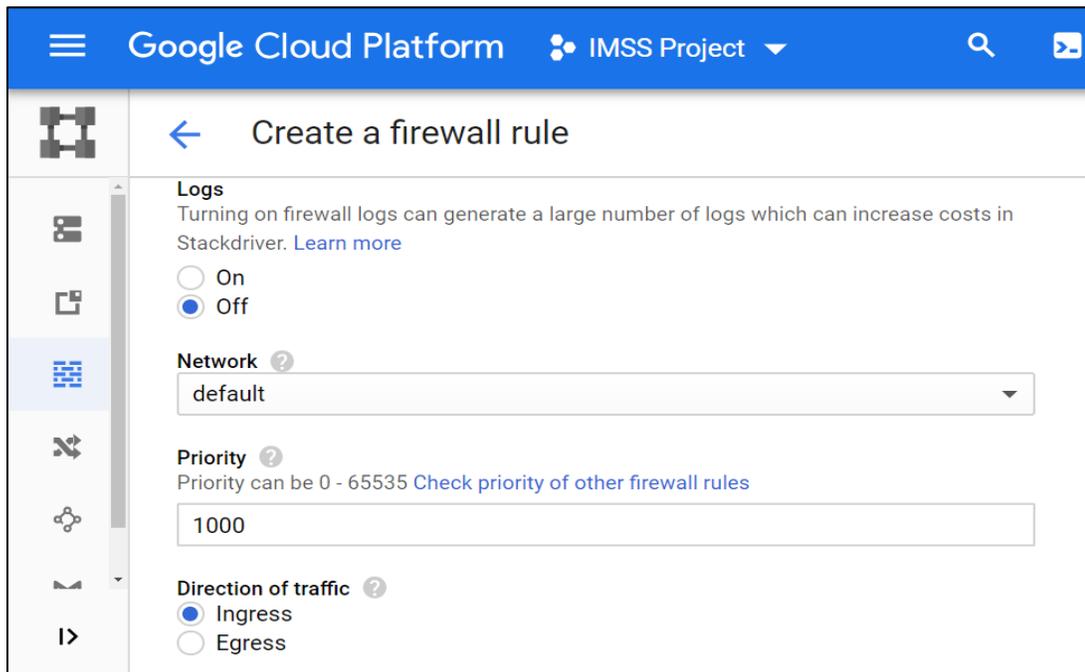
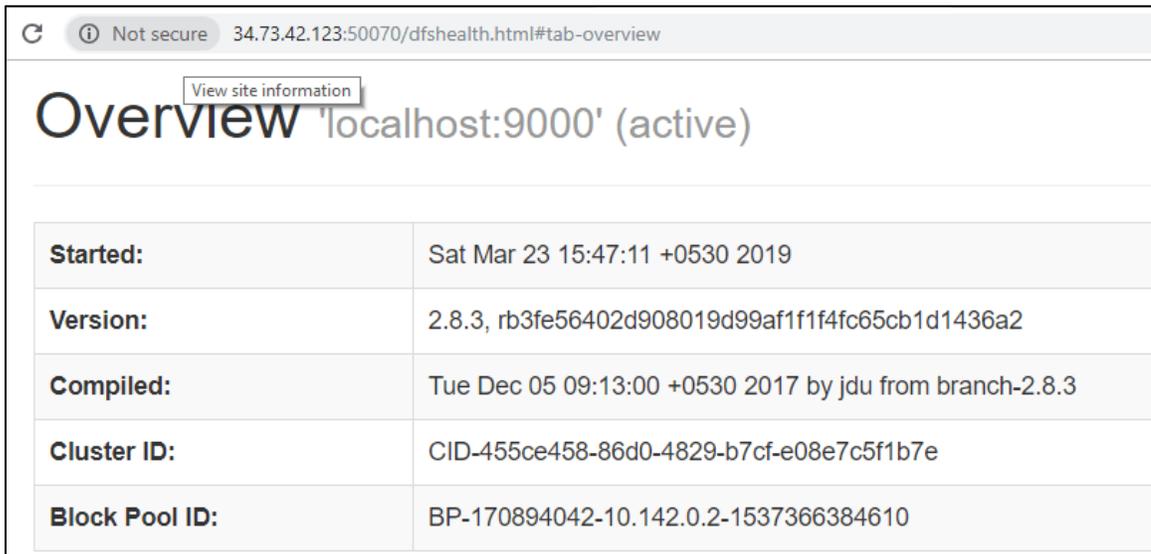


Fig. 4.34 Adding firewall rules

4.3.1.10. DFS Health Checkup

The external IP address of the instance is required to be copied to the browser address bar followed by a colon and then port no, for example like 35.231.192.17:50070/. The procedure to access the master node DFS health or NameNode information is shown in Fig. 4.35 and Fig. 4.36.



Started:	Sat Mar 23 15:47:11 +0530 2019
Version:	2.8.3, rb3fe56402d908019d99af1f1f4fc65cb1d1436a2
Compiled:	Tue Dec 05 09:13:00 +0530 2017 by jdu from branch-2.8.3
Cluster ID:	CID-455ce458-86d0-4829-b7cf-e08e7c5f1b7e
Block Pool ID:	BP-170894042-10.142.0.2-1537366384610

Fig. 4.35 NameNode information

4.3.2 Multi-Node Configuration

To set up a multi-node cluster, the steps as mentioned for a master node within section 4.3.1 are required to be repeated for each of the slave nodes of the cluster. The master and slave instances are required to be set up a single node instance. However, the additional steps required for a multi-node cluster are discussed as follows:

- (i) Networking and SSH Syncing
- (ii) Editing Masters and Slaves File
- (iii) Property Modification in .XML Files

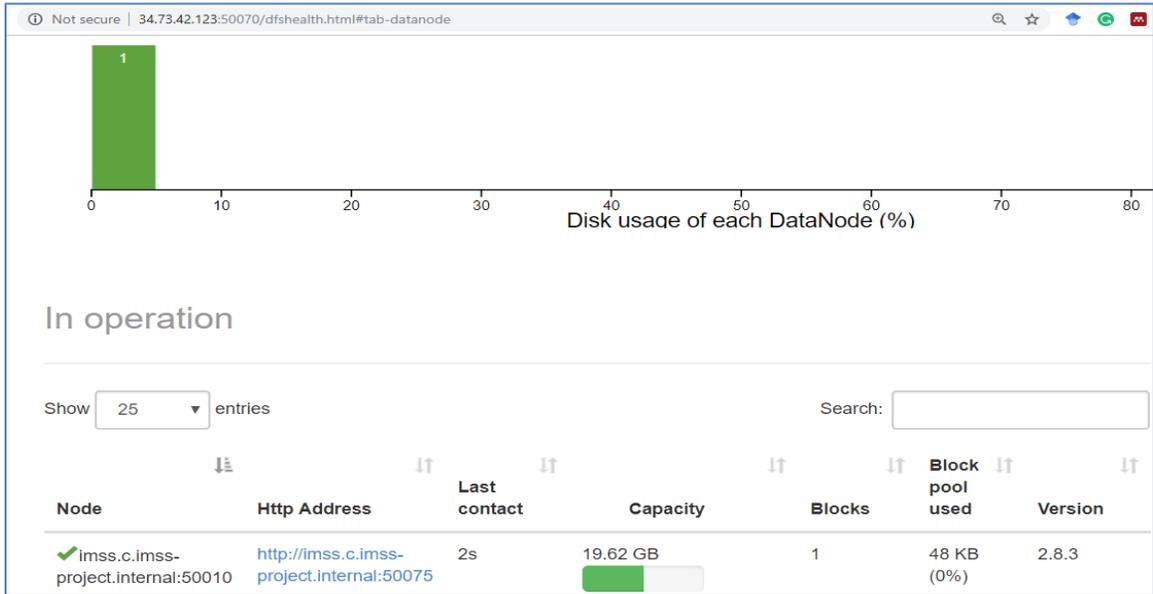


Fig. 4.36 Master node DFS

```

metasearchtool@master: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$ rm -rf /usr/local/hadoop/hdfs/namenode
metasearchtool@master:~$ rm -rf /usr/local/hadoop/hdfs/datanode
metasearchtool@master:~$ mkdir -p /usr/local/hadoop/hdfs/namenode
metasearchtool@master:~$ mkdir -p /usr/local/hadoop/hdfs/datanode
metasearchtool@master:~$ ls -l /usr/local/hadoop/hdfs/
total 8
drwxr-sr-x 2 metasearchtool staff 4096 Jun 11 16:02 datanode
drwxr-sr-x 2 metasearchtool staff 4096 Jun 11 16:02 namenode
metasearchtool@master:~$

```

Fig. 4.37 NameNode and DataNode recreation

The NameNode and DataNode directories are first required to be deleted and then recreated. This recreation is necessary to avoid any possibility for junk data within these directories and hence to facilitate multi-node configuration and can be accomplished through the execution of commands as shown in Fig. 4.37.

4.3.2.1. Networking and SSH Syncing

After completing the process of single node setup on all the three instances, i.e., Master, Slave1 and Slave2 instances, networking is required between all the three instances as all of them are working on different IP addresses as shown in Fig. 4.38. This networking can be accomplished in two sub-steps (i) IP address copying of all instances within */etc/hosts* file of the master as well as slave instances as shown in Fig. 4.39 (ii) Copying key generated at master instance within all slave instances as depicted in Fig. 4.40 and Fig. 4.41. After performing networking, connections between master and slave can be synced using *ssh* command. This command will allow accessing any of the slaves from a master window without requiring to open a dedicated SSH window of each of the slave as shown in Fig. 4.42

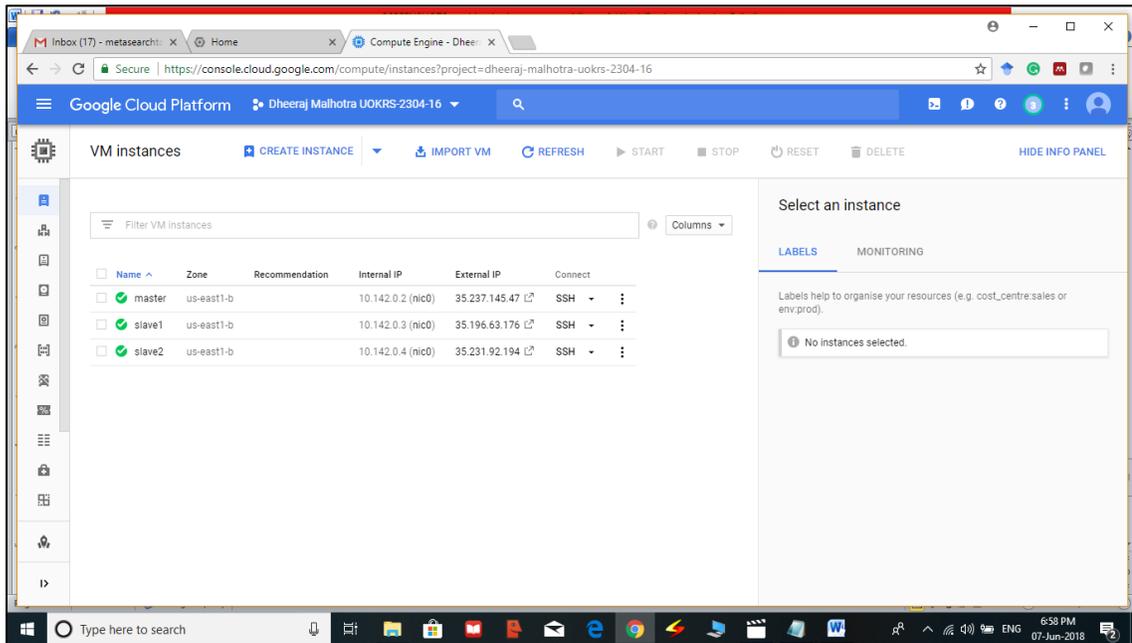


Fig. 4.38 External IP address information of all three instances

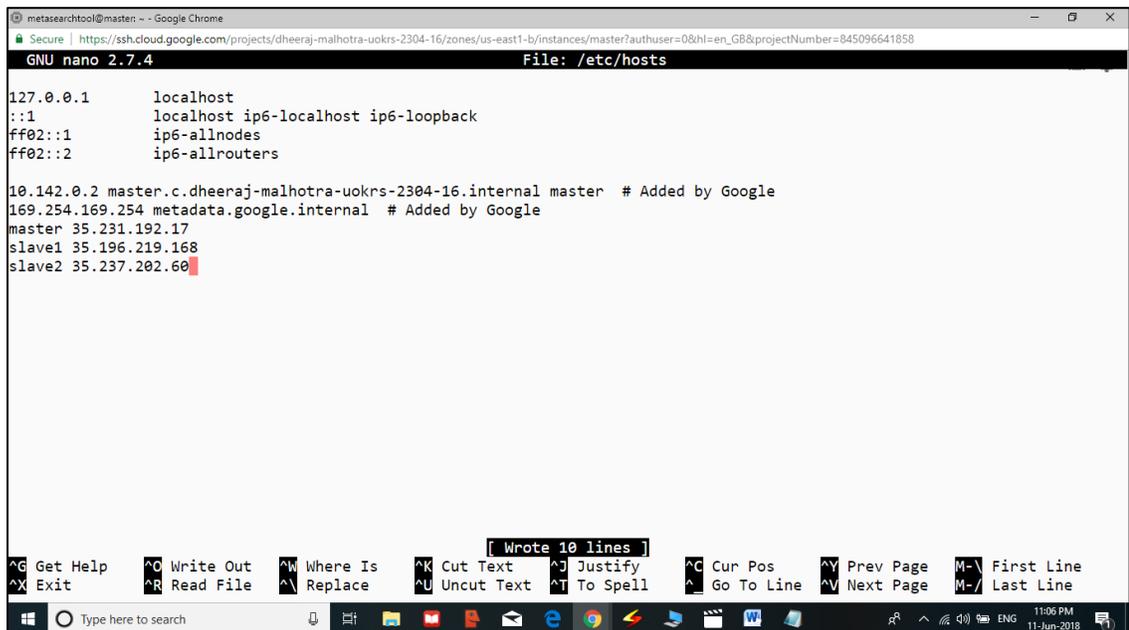
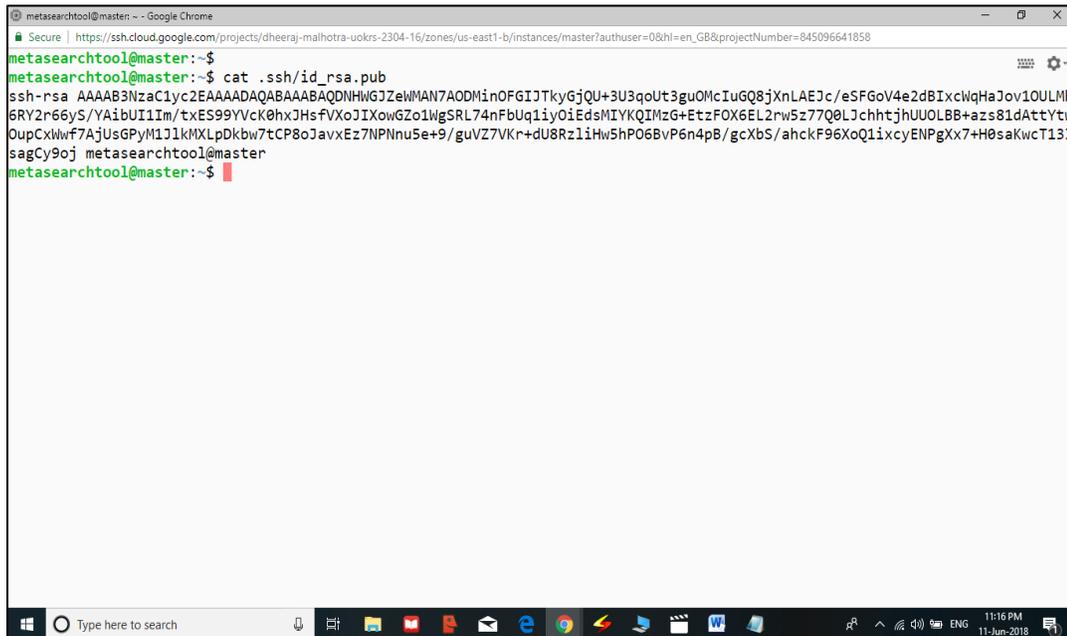


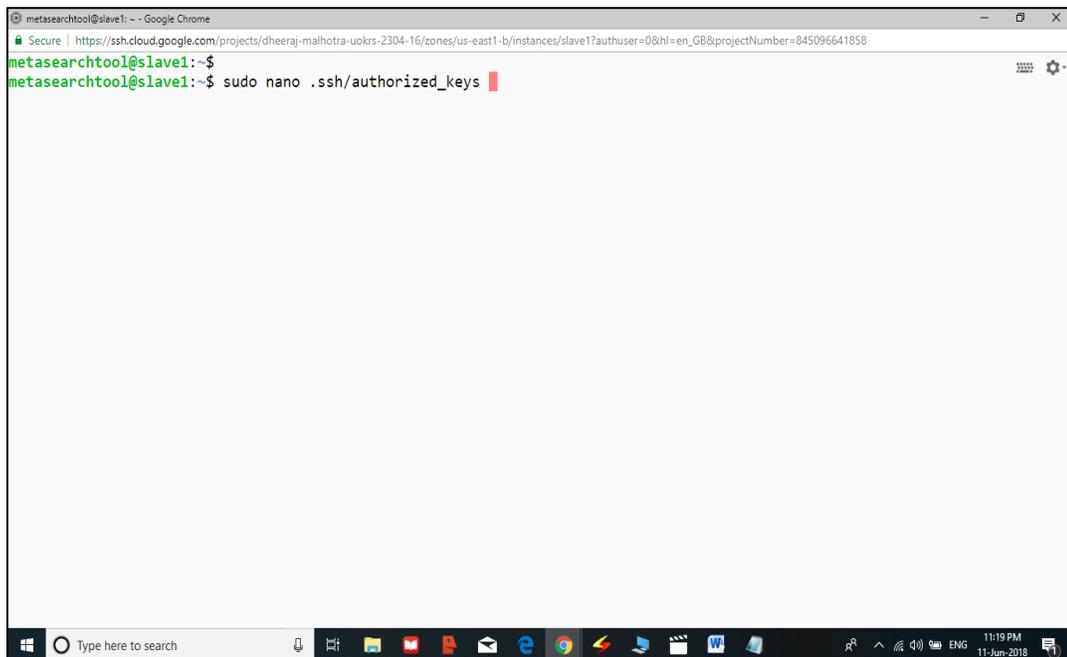
Fig. 4.39 Copying IP address information in /etc/hosts



A terminal window titled 'metasearchtool@master' showing the execution of 'cat .ssh/id_rsa.pub' and the output of an SSH public key. The key is a long string of characters starting with 'ssh-rsa AAAAAB3NzaC1yc2EAAAADAQABAAQDNHwGJZeWMAN7AODMinOFGIJTkyGjQU+3U3goUt3guOMcIuGQ8jXnLAEJc/eSFGoV4e2dBIxcWqHaJov10ULMh6RY2r66yS/YAibUIIIm/txES99YVcK0hxJHsfVxoJIXowGZo1WgSRL74nFbUq1iyOiEdsMIYKQIMzG+EtzFOX6EL2rw5z77Q0LjchhtjhUUOLBB+azs81dAttYtwOupCxWwF7AjUsGPYMIJ1kMXLpDkbw7tCP8oJavxEz7NPNnu5e+9/guVZ7VKr+dU8Rz1iHw5HP06BvP6n4pB/gcXbS/ahckF96XoQ1ixcyENPgXx7+H0saKwct13IsagCy9oj metasearchtool@master'.

```
metasearchtool@master:~$ cat .ssh/id_rsa.pub
ssh-rsa AAAAAB3NzaC1yc2EAAAADAQABAAQDNHwGJZeWMAN7AODMinOFGIJTkyGjQU+3U3goUt3guOMcIuGQ8jXnLAEJc/eSFGoV4e2dBIxcWqHaJov10ULMh6RY2r66yS/YAibUIIIm/txES99YVcK0hxJHsfVxoJIXowGZo1WgSRL74nFbUq1iyOiEdsMIYKQIMzG+EtzFOX6EL2rw5z77Q0LjchhtjhUUOLBB+azs81dAttYtwOupCxWwF7AjUsGPYMIJ1kMXLpDkbw7tCP8oJavxEz7NPNnu5e+9/guVZ7VKr+dU8Rz1iHw5HP06BvP6n4pB/gcXbS/ahckF96XoQ1ixcyENPgXx7+H0saKwct13IsagCy9oj metasearchtool@master
```

Fig. 4.40 Key generation at the master instance



A terminal window titled 'metasearchtool@slave1' showing the command 'sudo nano .ssh/authorized_keys' being entered.

```
metasearchtool@slave1:~$ sudo nano .ssh/authorized_keys
```

Fig. 4.41 Command to copy a master key within slave instance

```
metasearchtool@slave2 -- Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
metasearchtool@master:~$ ssh slave2
Linux slave2 4.9.0-6-amd64 #1 SMP Debian 4.9.88-1+deb9u1 (2018-05-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jun 12 06:52:25 2018 from 74.125.41.104
metasearchtool@slave2:~$
```

Fig. 4.42 SSH syncing between master and slave2 instance

4.3.2.2. Editing Masters and Slaves file

Before, starting with configuration of .XML files, one need to update the masters and slaves file. This update is necessary as master instance will act both as NameNode and DataNode. However, a slave instance will work like DataNode only. The editing of masters and slaves file is shown in Fig. 4.43, 4.44 and 4.45.

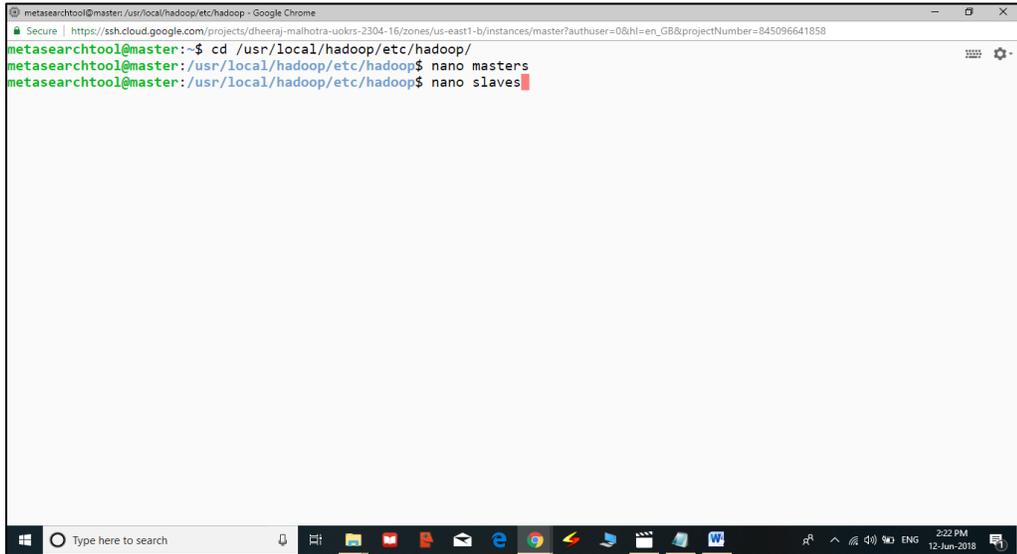


Fig. 4.43 Commands to edit *masters* and *slaves* file

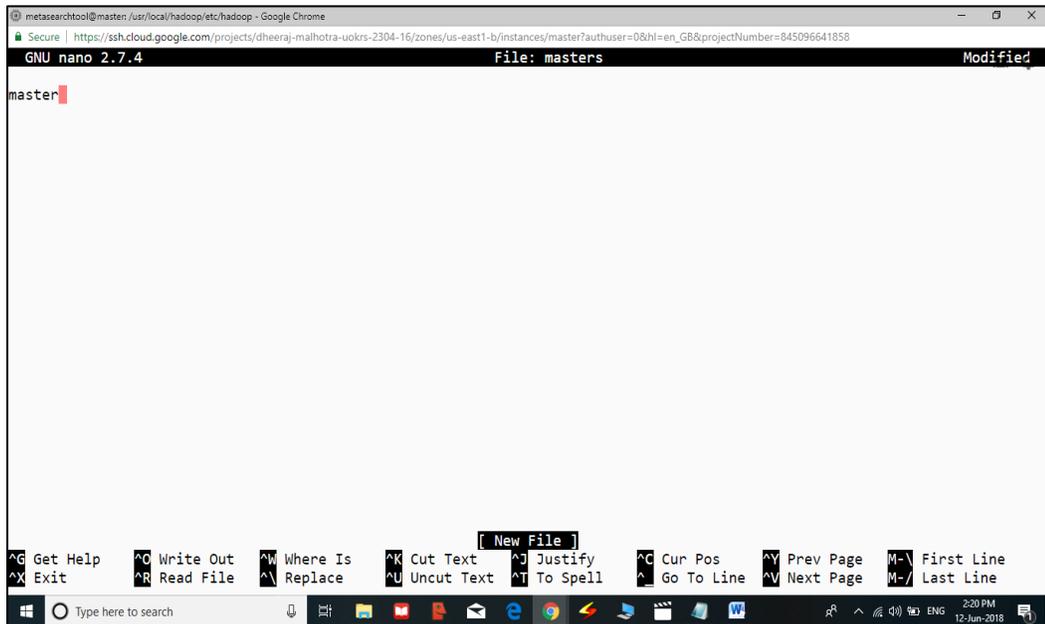


Fig. 4.44 Editing *masters* file

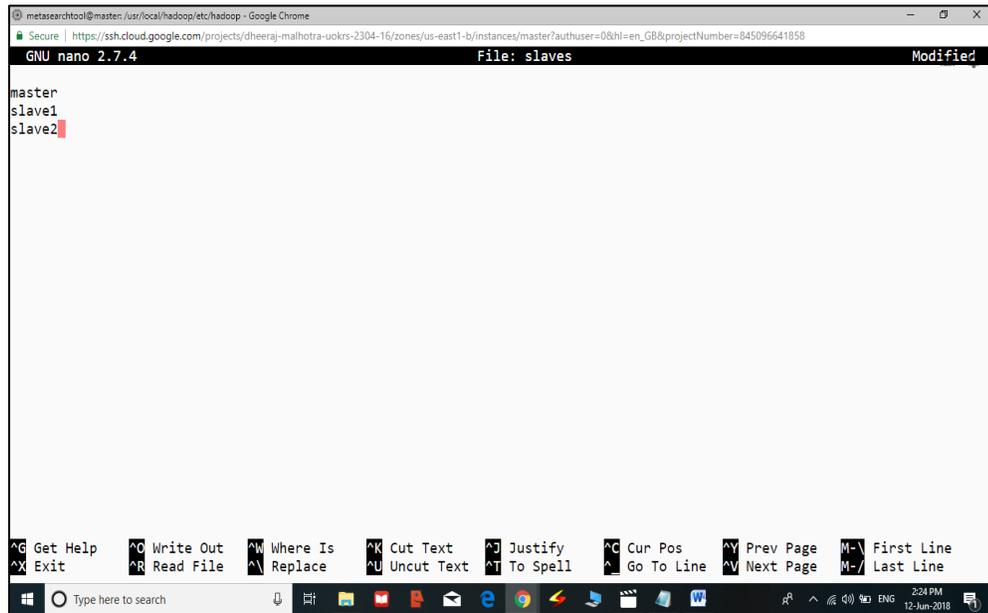


Fig. 4.45 Editing *slaves* file

4.3.2.3. Property Modification in .XML Files

To complete the process of multi-node cluster setup, there is a need to edit property tags within various .XML files at master and all slave instances. The property tag of core-site.xml is modified in all instances to specify master: 9000 instead of localhost: 9000 as shown in Fig. 4.46.

Similarly, there is a need to update replication factor from 1 to 3 in hdfs-site.xml of all the instances as we have a total of 3 data nodes including one master node and two slave nodes as shown in Fig. 4.47. Moreover, an additional property tag specifying the *mapred.job.tracker* information is required to be appended within mapred-site.xml of all the three instances as shown in Fig. 4.48. However, yarn-site.xml does not require any modification in any of its property.

The screenshot shows the nano 2.7.4 editor in a terminal window. The file being edited is core-site.xml. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://master:9000</value>
</property>
</configuration>
```

The terminal window also shows the nano editor's command palette at the bottom, including options like 'Get Help', 'Write Out', 'Where Is', 'Cut Text', 'Justify', 'Cur Pos', 'Prev Page', 'First Line', 'Exit', 'Read File', 'Replace', 'Uncut Text', 'To Spell', 'Go To Line', 'Next Page', and 'Last Line'. The system tray at the bottom right shows the time as 2:39 PM on 12-Jun-2018.

Fig. 4.46 Property tag modification in core-site.xml

The screenshot shows the nano 2.7.4 editor in a terminal window. The file being edited is hdfs-site.xml. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>dfs.replication</name>
  <value>3</value>
</property>
</configuration>
```

The terminal window also shows the nano editor's command palette at the bottom, including options like 'Get Help', 'Write Out', 'Where Is', 'Cut Text', 'Justify', 'Cur Pos', 'Prev Page', 'First Line', 'Exit', 'Read File', 'Replace', 'Uncut Text', 'To Spell', 'Go To Line', 'Next Page', and 'Last Line'. The system tray at the bottom right shows the time as 2:48 PM on 12-Jun-2018.

Fig. 4.47 Replication factor modification in hdfs-site.xml

```
metasearchtool@master:/usr/local/hadoop/etc/hadoop - Google Chrome
Secure | https://ssh.cloud.google.com/projects/dheeraj-malhotra-uokrs-2304-16/zones/us-east1-b/instances/master?authuser=0&hl=en_GB&projectNumber=845096641858
GNU nano 2.7.4 File: mapred-site.xml Modified

  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
<property>
<name>mapred.job.tracker</name>
  <value>master:54311</value>
</property>
</configuration>
```

Fig. 4.48 Appending additional property tag in mapred-site.xml

The process of adding firewall rules and starting and verifying DFS and yarn services is similar to that of a single node cluster setup. The NameNode information and DFS health can be retrieved for various instances like master and slave instances as already discussed in section 4.3.1.8, 4.3.1.9 and 4.3.1.10.

4.4. CHAPTER SUMMARY

This chapter discusses the various features provided by the Google Cloud Platform (GCP). These features can be used to set up a single node and multi-node cluster for big data analytics. The multi-node cluster can be implemented in HDFS and Map-Reduce environment to perform personalized page ranking as required by deployed metasearch application, i.e., IMSS tool.

CHAPTER 5

IMSS: SYSTEM DESIGN OF INTELLIGENT META SEARCH SYSTEM

5.1. INTRODUCTION

This chapter discusses the detailed three-phase implementation design of the Intelligent Meta Search System (IMSS). The first two phases describe best match prediction and search query disambiguation followed by the third phase for personalized web page ranking using the innovative Advanced Cluster Vector Page Ranking (ACVPR) algorithm. The interface of the IMSS tool deployed using ACVPR algorithm for web search personalization is also discussed in detail.

5.2. SYSTEM DESIGN

The present research work addresses the problem of personalized web page search and ranking using intelligent analytics based on big data and machine learning. The system design of the deployed IMSS tool is shown in Fig. 5.1. The prescribed system design is capable of predicting best match user ID using machine learning followed by query disambiguation or query expansion. The query disambiguation is followed by re-ranking of results retrieved from various search engines like Google, Bing, and Qwant using Intelligent Meta Search System (IMSS). Our contribution is a new Advanced Cluster Vector Page Ranking (ACVPR) algorithm used to implement the IMSS tool. The detailed discussion about ACVPR algorithm is in section 5.2.3. The deployed system design consists of the following three phases:

- Phase 1: Best match prediction using a machine learning model

- Phase 2: Query disambiguation and web page retrieval
- Phase 3: Web page ranking using IMSS tool and ACVPR algorithm

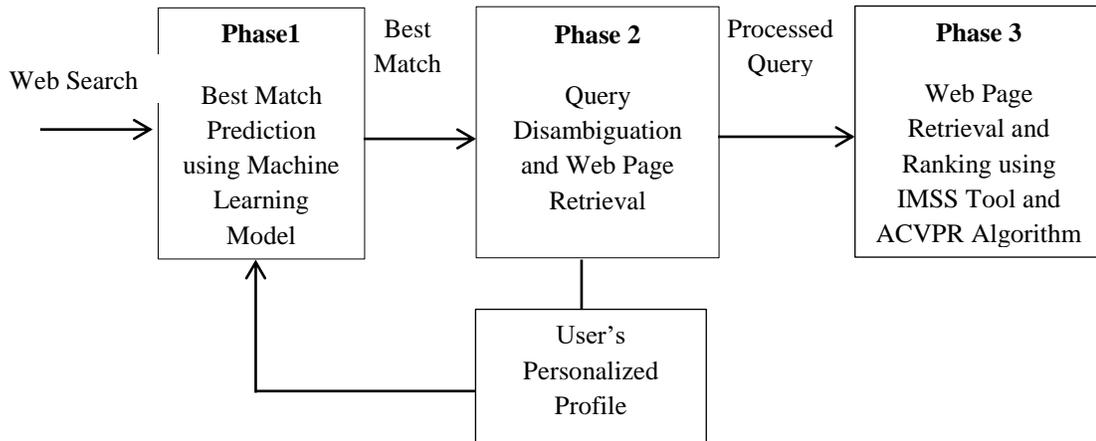


Fig. 5.1 System design of IMSS tool

5.2.1. Phase 1: Best Match Prediction using Machine Learning Model

To determine the personalized preferences of a user by predicting feedback for a web link and best match existing user ID, phase 1 implements here two different machine learning models (i) logistic regression and (ii) collaborative filtering. The logistic regression based model is used to predict the feedback for a given web-link to determine its ranking in the final output corresponding to a search query of the user. The collaborative filtering based model is used to predict information about the best match existing User ID of the system. This information assists in suitable expansion and disambiguation of the search queries by suggesting previously searched queries, sharing similar keywords with the current query, and are made by existing best match users of the system. The latest browsing history of the best match user is considered for predicting preferences of the current user. The data is required to be in the .csv format as required by R statistical tool (Malhotra & Rishi, 2018b). The .csv format file will consist of the data about the following five variables:

- Feedback represents the relevancy response by the user for the previous web link in his browsing history and can take either of two values, i.e., Yes or No
- Loading represents the web page loading experience of the user and can take either of two values, i.e., Good or Bad
- Response represents the response time experience of the user and can take either of two values, i.e., Good or Bad
- Security represents the security protocol feature provided by the candidate web page and can take either of two values, i.e., Yes or No
- Personalized represents the usage of the *Search Recommendation* feature on the interface of the tool, i.e., personalized expansion of the query by the user as available on the tool interface and can take either of two values, i.e., Yes or No

As the response variable, i.e., feedback is binomial in the first type of machine learning model so that we will use family = binomial (link = "logit") while creating the personalized search model. This syntax can be easily understood in mathematical terms as discussed below:

The natural logarithm of the odds ratio may be expressed as

$$\ln(\text{odds ratio}) = \ln [P / (1 - P)] \quad \dots\dots (1)$$

where, P = Probability of success or probability of response, i.e., Feedback = Yes

$$\text{logit}(P) = \ln [P(\text{Feedback} = \text{Yes}) / P(\text{Feedback} = \text{No})] = C_0 + C_1 \times \text{Loading} + C_2 \times \text{Response} + C_3 \times \text{Security} + C_4 \times \text{Personalized} \quad \dots\dots (2)$$

To improve the prediction accuracy of the response variable, i.e., feedback by predicting the natural logarithm of the odds ratio, the probability of accurately predicting the feedback response of the web user in the recommended model may be calculated as follows:

$$\text{Probability of true feedback} = \text{predicted odds ratio} / (1 + \text{predicted odds ratio}) \dots (3)$$

5.2.1.1. Steps for Generating and Testing Machine Learning Model

- Reading feedback.txt file to retrieve search data
- Model generation using various search parameters
- Plotting diagnostic curves for the generated model
- Recasting model by removing non-significant search parameter
- Deviance calculation between original and recast model
- Testing for multicollinearity and over-dispersion
- Plotting diagnostic curves for recast model

The regression and collaborative filtering based machine learning models used within the present research work are discussed in detail within chapter 7.

5.2.2. Phase 2: Query Disambiguation and Web Page Retrieval

In today's era of big data, even a state of art search engine is fetching output links on the top that may not be relevant to the user. Moreover, if the search query is ambiguous or incomplete, then even a popular search engine is not likely to produce the appropriate result as conventional search engines tend to return the result by interpreting all possible meanings of the query. Hence, query disambiguation is one of the critical characteristics of the IMSS tool for personalized information retrieval. As shown in Fig. 5.1, the web search query is processed in phase2 to determine the keywords of the search query. This step is followed by the determination of usage of similar keywords by other users in their queries possessing best match or similar profile to that of the current user of the tool. The browsing history of the best match user is retrieved to determine the most appropriate search queries to be recommended to the current user. For instance, if the previous user with best match ID already searched for queries like "Apple iPhone" or "Apple iPhone XS". Moreover if the current user inputs a partial or incomplete query like "Apple," then the previous user's search queries will be shown as recommendations to the current user because of the matching profile of two users as reported by machine learning based

recommendation module and sharing keyword, i.e., "Apple" between the search queries of the two users. The expanded or disambiguated personalized queries are then presented to the user. The user may select any one of the suggested query or may continue to search his or her original unexpanded query. The final query chosen by the user is further passed to all three background search engines, i.e., Google, Qwant, and Bing to retrieve web links. The IMSS tool can thus possess a good recall characteristic by extracting massive volume of the web because of the involvement of three popular and giant search engines in its background, i.e., Google, Qwant and Bing. However, as already discussed, search pages retrieved may be initially ranked in a biased or non-relevant order by these search engines to support paid or advertised web links. Hence top ranked links from each of the background search engines are passed to phase-3 for further shortlisting. The re-ranking is accomplished by predicting feedback of each of the shortlisted web link considering recent browsing history of the previous user possessing best match profile through regression based machine learning model as discussed in detail within section 7.3.1 of chapter 7 to better satisfy the personalized search needs of the current user. The number of web links chosen from each of background search engine for further processing and shortlisting in phase-3 is dependent upon the number of running Virtual Machine (VM) instances to satisfy the real-time response requirement of the user. The number of VM instances may be dynamically activated depending upon the current load of the system. Hence deployed IMSS tool possesses elastic scaling through the implementation of Hadoop 2 based analytics framework.

5.2.3. Phase 3: Web Page Ranking using Advanced Cluster Vector Page Ranking (ACVPR) Algorithm

Our contribution is design and development of a new Advanced Cluster Vector Page Ranking (ACVPR) algorithm to assist the user to frame unambiguous search query and hence to retrieve a relevant web page. The ACVPR algorithm is implemented in the form of Intelligent Meta Search System (IMSS) tool for personalized query recommendation

and web page ranking. The ACVPR is an improvement of our previous published Relevancy Vector (RV) algorithm (Malhotra & Rishi, 2018 a, b). ACVPR is an advanced version of RV algorithm due to two main reasons (i) ACVPR is a generic page ranking algorithm, while RV algorithm is adapted for the page search and ranking of E-Commerce websites only (ii) ACVPR algorithm unlike RV algorithm incorporate machine learning based regression model and collaborative filtering model

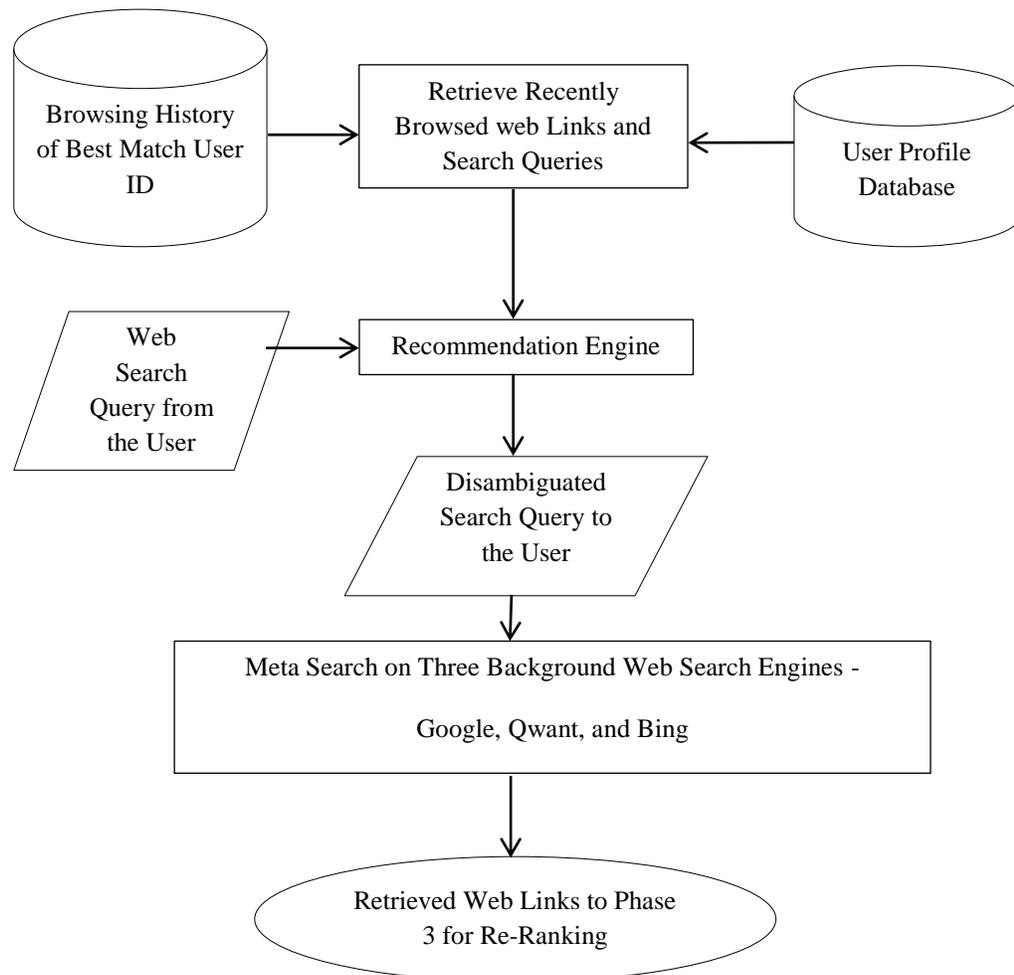


Fig. 5.2 Phase 2: Query disambiguation & web page retrieval

The pioneered *ACVPR algorithm* is step by step discussed as follows:

- Start

- Accept a web search query from a user
- Split the query into various keywords W_1, W_2, \dots, W_n .
- Determine the minimum (min) and maximum (max) length of each of the keyword as follows:
 - Set $\text{min} = \text{strlen}(W_1)$, $\text{max} = \text{strlen}(W_1)$
 - Set $c = 2$
 - While ($c < n$) do
 - If $\text{MIN} > W_c$ then
 - $\text{MIN} = \text{strlen}(W_c)$
 - EndIf
 - If $\text{MAX} < W_c$ then
 - $\text{MAX} = \text{strlen}(W_c)$
 - EndIf
 - EndWhile
- Execute a personalized search query on all or selected backend search engines
- Preprocess each web page in the form of a *web dictionary* consisting of keywords having a length between min and max as determined in step no.4
- Predict relevancy of each of the returned web page from each of the search engines by generating machine learning model using $\text{glm}()$ function
- Check for null deviance, residual deviance, fisher scoring and possible errors like multi-collinearity and over-dispersion to determine the model effectiveness in ascertaining best user match and hence feedback of a prospective web page
- Remove all those model generation parameters for which predictability value is more than 0.05 and recast the model if it satisfies any of the following condition:
 1. Residual deviance is more than null deviance
 2. Fisher scoring iterations are more than 8 or over-dispersion is more than 0.05

3. Search parameters have a substantial value of the standard error
 4. Variance inflation factor, $vif()$ is more than 5
- Determine user navigation session, which can be accomplished by comparing the user's query with each of the past search queries present in the user's profile database using the Longest Common Subsequence (LCS). The LCS is used to determine the proximity between various users' preferences and store the same in SRV [ID] to represent the similarity matrix to represent the personalized similarity between multiple users of the system
 - Calculate timestamp T_s of creation and average time spent by past user T_p to calculate Time Relevance Vector, $TRV [ID] = (T_s + T_p) / 2$
 - Calculate CRV [ID] as follows:
 - For $x=1$ to n do *// n refer to the total number of websites*
 - Calculate the frequency of each keyword using a *web dictionary* retrieved from step no.6
 - Call **Map** (*SEngine_ID, Web_Log*)
 - Call **Reduce** (*KL, count*)
 - Calculate the average frequency from the frequency of individual keywords
 - Store average frequency in CRV [ID]
 - End For
 - For $x=1$ to r do *// r refer to remaining websites*
 - Calculate Privacy vector, $PV[ID]=0$;
If (linkprivacy = privacy (website (ID))) then
set $PV[ID]=1$
 - Calculate Accessibility Vector, $AV[ID]=0$;
If (Cloud = Public) then set

AV[ID] =1

- Calculate Reply Time Vector, Set RTV[ID]=0

If (linkresponse > ReplyTime (website(ID)) then

RTV [ID] = strresponse – ReplyTime (website (ID))

End If

End For

- Eliminate all websites with either RTV[ID]= 0, PV[ID]=0 or AV[ID]=0
- Determine Feedback Relevancy Vector, i.e., FRV[ID] to analyze online reviews and categorize them into negative, positive and neutral reviews:

// Well satisfied user with feedback rating = 5 or 4

Set Count= 0

If (Review is Positive) then

Count = Count + 2

//Hesitant user with feedback rating= 3 or 2

Else If (Review is Neutral) then

Count = Count -1

// unsatisfied user with feedback rating= 1

Else If (Review is Negative)

Count = Count – 2

End IF

- Set FRV [ID] = Count
- Calculate Rank (website (ID)):= AV [ID]*((SRV [ID]* W1+ CRV [ID]*W2 + TRV [ID]*W3 + FRV [ID]*W4 +PV [ID]*W5+ RTV [ID]*W6)
- Accept feedback from the user about the shown ranking order and update

user profile database with a new value of weights, W1 to W6

- Stop

The Advanced Cluster Vector Page Ranking (ACVPR) algorithm determines the relevancy of a website for a specific user using the calculation of various relevancy vectors such as *Content Relevancy Vector (CRV)*, *Similarity Relevancy Vector (SRV)*, *Reply Time Vector (RTV)*, *Feedback Relevancy Vector (FRV)*, and *Privacy Vector (PV)*. The algorithm starts with a personalized expansion and disambiguation of the search query as discussed in section 5.2.2. The ACVPR algorithm will calculate the minimum and maximum length of each of the keyword of the search string. The SRV is determined using the Longest Common Subsequence (LCS) and similarity matrix calculation using collaborative filtering based machine learning model is discussed in detail within chapter-7. The CRV is determined using Map and Reduce functions as discussed in section 5.2.3.1. The algorithm will remove all those websites from final output with Reply Time Vector =0, Accessibility Vector =0, Privacy Vector =0. Moreover, search parameters which do not satisfy various criteria of regression-based learning and hence personalization like predictability, Fisher scoring, standard error, vif value, over-dispersion and multi-collinearity as imposed by the machine learning model are also required to be removed. The previous step is further followed by calculation of Feedback Relevancy Vector (FRV) depending on the experience of the past user. In last, the rank of a website is calculated by weighted summation of various relevancy vectors. The user feedback about the ranking of listed sites in recent web searches is considered for altering weight values for different vectors to incorporate recent changes in the user's personalized preferences. The weights to be assigned to various relevance vectors lie between 0 and 1 and are determined by long term and short term personalized preferences of the user as retrieved from his or her profile, browsing or search query history, and relevance feedback of the best match existing user of the system. The decision regarding the exact value of weights to be assigned to various relevance vectors is based on the prediction of user feedback for prospective web links and best match existing user of the

system as determined through collaborative filtering and regression-based recommendation modules discussed in detail within chapter7. The detailed flowchart of innovative ACVPR algorithm is shown in Fig. 5.3.

5.2.3.1. Map () and Reduce () Methods

Map method will accept a key as search engine ID for each retrieved web links cluster from various background search engines and the second argument is weblog to tokenize each of the entry of link entry in the weblog for counting frequency of each of the keyword in the web search query. *Insert ()* method is used to generate elements in the list by inserting numeric one corresponding to each occurrence of a keyword as we token. However, Reduce method is implemented to cumulate over all the occurrence of each keyword as indicated by Map () function through the insertion of numeric 1(one) to determine the frequency of the keyword in each of the web document. The map () and reduce () methods assist in the calculation of the individual frequency of various keywords of the disambiguated search query. The average frequency of different keywords is used to determine the Content Relevancy Vectors (CRV). The weighted contribution of CRV is used to determine the overall rank of a web page as discussed in ACVPR algorithm within section 5.2.3.

Map (SEngine_ID : Integer, Web_Log : String)

/ Web Log Cluster Processing */*

```
{
List<String> TL: = Tokenize (Web_Log)           // TL- Token List
  While (Web_Token in TL)
    {
      Insert ((String) KL, (Integer) 1)         // KL- Keyword List
    }
}
```

Reduce (KL: String, count: List <Integer>)

// Frequency calculation

```
{
Integer Freq = 0
While(KL)
  {
    Freq = Freq + 1
  }
Insert ((String) Web_Token, (Integer) Freq)
}
```

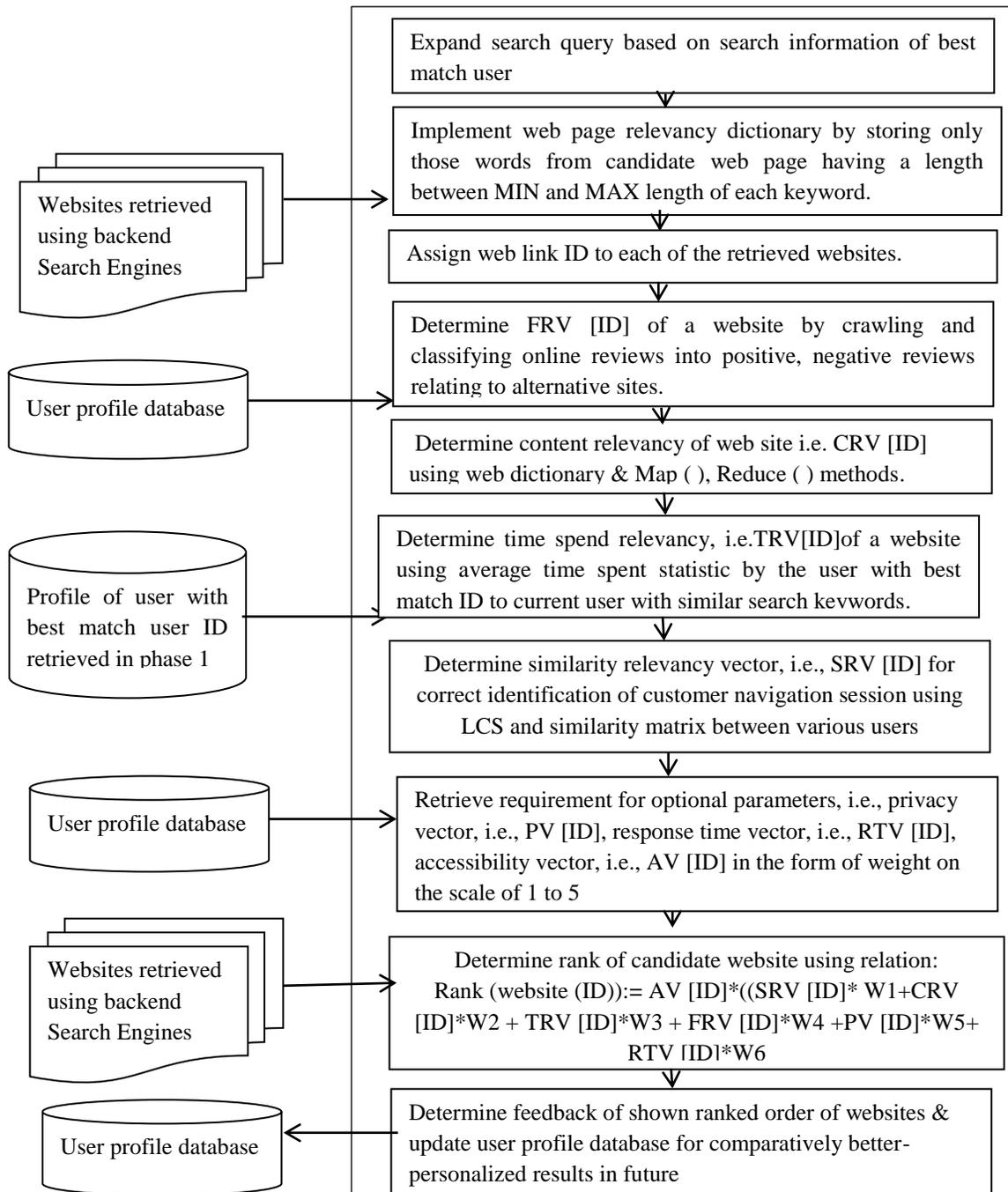


Fig. 5.3 Flowchart of ACVPR algorithm

5.3. INTELLIGENT META SEARCH SYSTEM (IMSS) TOOL

In order to check the personalized search effectiveness and efficiency of the pioneered ACVPR algorithm, current research work discusses its implementation in the form of Intelligent Meta Search System (IMSS). This tool is a machine learning enabled metasearch engine which uses a Python interpreter on the server side for implementation of the data analysis. The results of the analysis are processed using PHP, HTML 5, and CSS 3 and finally displayed on a user browser. This tool is deployed on Google cloud engine with Hadoop features enabled for computation of user similarity, rating based recommendations. The tool uses a MySQL engine for search query management. This system employs logistic regression and collaborative filtering based machine learning techniques for personalized search recommendations by predicting feedback of a user about prospective web link and by predicting the best match existing user of the system. The system requires the user to sign up and answer a few questions only for the first time within the *Add Skills* section for emotional and behavior analysis. By similarity score calculation based on behavior analysis between various users, the IMSS system will determine an existing best match user ID for a new user searching the web. The user similarity is determined by using the concept of Euclidean distance. The detailed information about best match along with similarity information will be shown to the user. If the search query of a new user includes keywords of the search query of previous users with best match ID or same profession type, then the system will recommend the search queries sharing keywords made by the best match user to the current user. All web links in the result presented to the user will allow the user to rate the rank and relevance of the output web link. The rating provided by the user will be used to alter priority or rank of the output link when a user is searching a query including keywords of previously searched queries by other existing users of the system with a similar profile with preference given to the best match user as recommended by the system. The system can work in two modes, *personalized search mode*, and *advanced search mode*. The advanced search mode let the user select search engines among Google, Bing, and Qwant

to be used as background search engines for the implemented metasearch tool, i.e., IMSS. Moreover, a user can also sort the output links in the order of page loading speed. These features are not available within personalized search mode. The detailed description and interface of the tool are shown in the subsections from 5.3.1 to 5.3.7.

5.3.1. IMSS Tool- Sign Up and Sign In

The user intended to perform a personalized search on the web needs to access the Intelligent Meta Search System (IMSS) through the *Sign In* interface as shown in Fig. 5.4. The user needs to fill his or her contact details along with profession details during *Sign Up* as shown in Fig. 5.5.

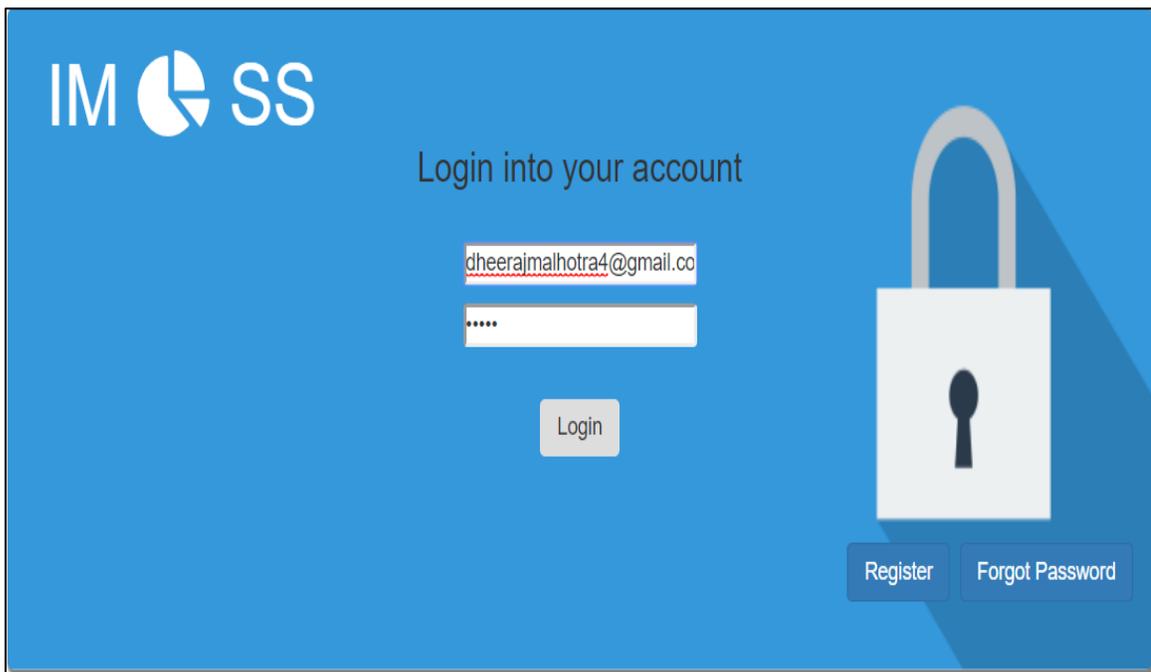


Fig. 5.4 Sign In interface of the IMSS tool

IM SS

Registration Form

dheeraj malhotra

9811949964

dheerajmalhotra4@gmail.co

.....

Select Profession ▼

Select Profession

Doctor

Teacher

Login

Fig. 5.5 Sign up-User registration

5.3.2. Behavior Analysis

As soon as the user fills the registration form, he or she will be prompted to add skills for behavior analysis. The user will be presented with a set of a few multiple choice questions. The answers given by the user will be used to determine the similarity of the current user with other existing users of the same profession already registered within the tool. This analysis, in turn, helps to identify search recommendations through collaborative filtering based machine learning technique. It may be noted that the user will be prompted to add skills only for the first time during the sign-up process. However, during sign in, questions are not repeatedly asked. The interface of the tool to add skills is shown in Fig. 5.6.

User ID: dheeraj.malhotra@vips.edu	Home	Change Password	Logout
Skill Query			
<h2>Skills</h2>			
Q1) I am a good decision maker.			
<input type="radio"/> Most Likely	<input type="radio"/> Somewhat Likely	<input type="radio"/> Least Likely	
<input type="submit" value="Submit"/>			
Q2) My decisions are usually accepted as good by the person affected.			
<input type="radio"/> Most Likely	<input type="radio"/> Somewhat Likely	<input type="radio"/> Least Likely	
<input type="submit" value="Submit"/>			
Q3) When faced with an important decision, I am not overly anxious about making a wrong choice.			

Fig. 5.6 Adding user skills

5.3.3. Personalized Search Mode and Tracking Recent Changes in the User Preferences

The proposed and implemented IMSS tool can work within two search modes (i) Personalized Search Mode (ii) Advanced Search Mode. The interface of personalized search mode is comparatively more user-friendly than advanced search mode. The interface allows the user to enter the search string. The personalized search mode will automatically select all the three background search engines, i.e., Google, Qwant and Bing for searching the query on the web. There are two types of buttons available beneath search string box, i.e., *Fast Forward* and *Search Recommendation*. The *Fast Forward* button will let the user to directly search using three-parent search engines without providing any personalized search recommendations. However, *Search Recommendation* button will provide search recommendations based on the search history of other users

possessing the best match or similar profiles. Here, the priority or rank of the links is also altered for ratings provided by the previous user. The user interfaces showing personalized search recommendations and web page ranking are shown in Fig. 5.7 and Fig. 5.8.

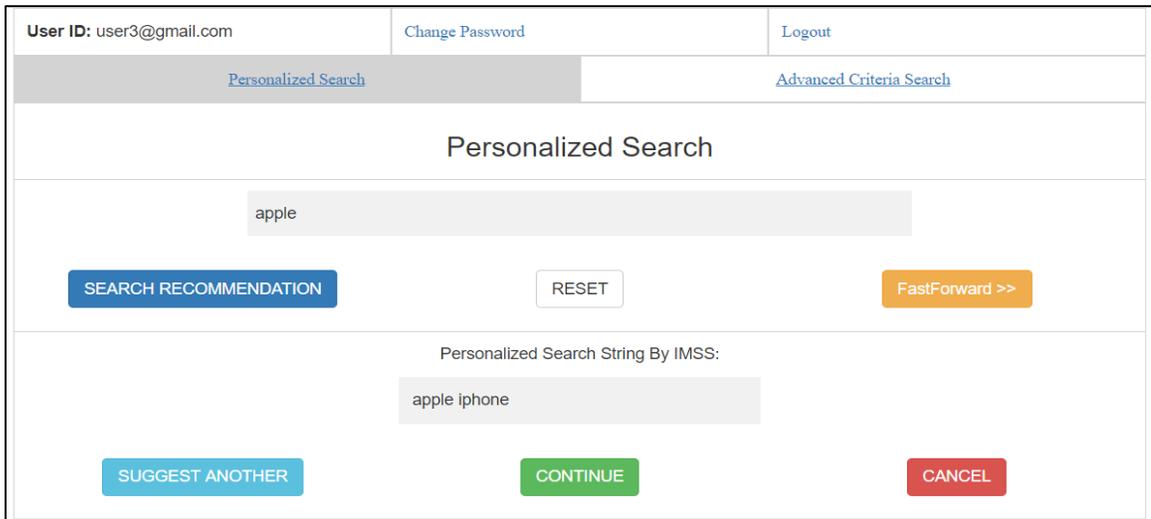


Fig. 5.7 Search recommendations in personalized search mode

Priority / Rank	Web Link (Search Engine)	Response Time	Security	Loading speed	Rating
1	https://www.youtube.com/watch?v=PRML7k_ve_I (G)	1.26 ms	Secure	1.29 ms	1 ▾
2	https://samsungvr.com/ (G,Q)	1.26 ms	Secure	1.28 ms	1 ▾
3	https://www.bestbuy.com/site/samsung-store/samsung-gear-vr/pcmcat381400050004.c?id=pcmcat381400050004 (G)	1.26 ms	Secure	1.27 ms	1 ▾
4	https://www.youtube.com/watch?v=LX7iruyrM (G,Q)	1.26 ms	Secure	1.28 ms	1 ▾
5	https://en.wikipedia.org/wiki/Samsung_Gear_VR (G,Q)	1.26 ms	Secure	1.27 ms	1 ▾
6	https://www.amazon.com/d/Pc-Virtual-Reality-Headsets/Samsung-Gear-Controller-Discontinued-Manufacturer/B06XJJ7CRQ (G)	1.26 ms	Secure	1.26 ms	1 ▾ 1 2 3 4 5
7	https://www.samsung.com/us/mobile/virtual-reality/gear-vr/gear-vr-with-controller-sm-r324nzaaxar/ (G,Q)	1.26 ms	Secure	1.26 ms	1 ▾
8	https://www.oculus.com/gear-vr/ (G)	1.26 ms	Secure	1.26 ms	1 ▾

Fig. 5.8. Web page ranking using IMSS tool

The user search preferences are prone to change over some time. The deployed IMSS tool can easily incorporate recent changes in personalized preferences of the users via tracking user ratings for recent searches performed on the web. For instance, high rating 5 to some of the web links in the recent searches indicate current preference bend of the user. The IMSS tool uses new preferences while re-ranking web pages to satisfy better-personalized search needs of the user rather than just following the first time created profile of the user during sign up in the system. Thus, along with query disambiguation and expansion using profile of the best match user, the implemented IMSS tool can also keep track of new preferences by referring recent browsing history and rankings given to the listed web links in last web search sessions. The changes in user preferences are automatically incorporated to update the user profile and hence automatic determination of new best match user of the system without requiring the explicit intervention of the user unlike conventional personalized search systems discussed in the literature.

5.3.4. Advanced Search Mode

The advanced search mode allows the user to select one, two or all the three search engines to be used by the deployed metasearch tool. Here, also a *fast-forward* button is used to search on the web without providing personalized search recommendations. However, the *Search* button will provide recommendations based on the previous search performed by the best match or other similar profile users. The advanced search mode allows the user to sort the results in the order of page loading speed. The user interface within the advanced search mode is shown in Fig. 5.9 and Fig. 5.10.

User ID: user1@gmail.com

[Change Password](#)

[Logout](#)

[Personalized Search](#)
[Advanced Criteria Search](#)

Advanced Search Criteria

Google

Qwant

Bing

Sort by Loading Speed

SEARCH

RESET

FastForward >>

Fig. 5.9 Advanced search mode – search engine selection

1	https://www.researchgate.net/publication/310361044_IMSS-E_An_Intelligent_Approach_to_Design_of_Adaptive_Meta_Search_System_for_E_Commerce_Website_Ranking (Q,G)	2.35 ms	Secure	2.37 ms	1 ▼
2	https://www.amazon.in/Data-Structures-Program-Design-Using/dp/1683922077 (Q)	2.35 ms	Secure	2.36 ms	1 ▼
3	https://www.sciencedirect.com/science/article/pii/S1319157818307869 (Q,G)	2.35 ms	Secure	2.36 ms	1 ▼
4	http://iranarze.ir/wp-content/uploads/2018/12/E10300-IranArze.pdf (Q)	2.35 ms	Not secure	2.36 ms	1 ▼
5	https://rd.springer.com/chapter/10.1007/978-981-10-2750-5_20 (Q)	2.35 ms	Secure	2.35 ms	1 ▼

Fig. 5.10 Page loading speed based web page ordering in advanced search mode

5.3.5. Machine Learning Statistics

The interface of the IMSS tool shows the accuracy of personalized search recommendations. The tool implements a collaborative filtering based prediction. The evaluation metrics displayed on the tool interface include information about Root Mean

Squared Error (RMSE), Mean Absolute Error (MAE). The machine learning summary provides information regarding the number of users in the system, number of questions, accuracy matrix, prediction matrix, user similarity matrix, and best match user ID. The detailed description of various evaluation metrics and matrices is discussed within the chapter number 7. The accuracy metrics shown on the interface of the IMSS tool is shown in Fig. 5.11.

5.3.6. Security-Personalized Privacy Protection

The personalized search systems provide information regarding search queries made by other existing users of the system to the current user to improve or expand their search queries. The user feels uncomfortable when someone else can explore the type of search queries made by them. However, the IMSS tool protects the personal information of the users from one another. The machine learning statistics on the interface of the IMSS tool hides the user's personal information such as name, email ID, mobile number, etc. The best match is existing user ID for the current user, and system generated ID is shown at the bottom of the machine learning summary as shown in Fig. 5.11. The available statistics displayed on the interface of the tool are useful for research purpose and to easily track improvements in the machine learning capabilities of the system. However, at the same time user's personalized information is kept secured and is not accessible to other users of the system. Thus, unlike conventional personalized page ranking systems as discussed in the literature, users do not get hesitant in providing personal information and personalized search preferences while exploring web through deployed IMSS tool.

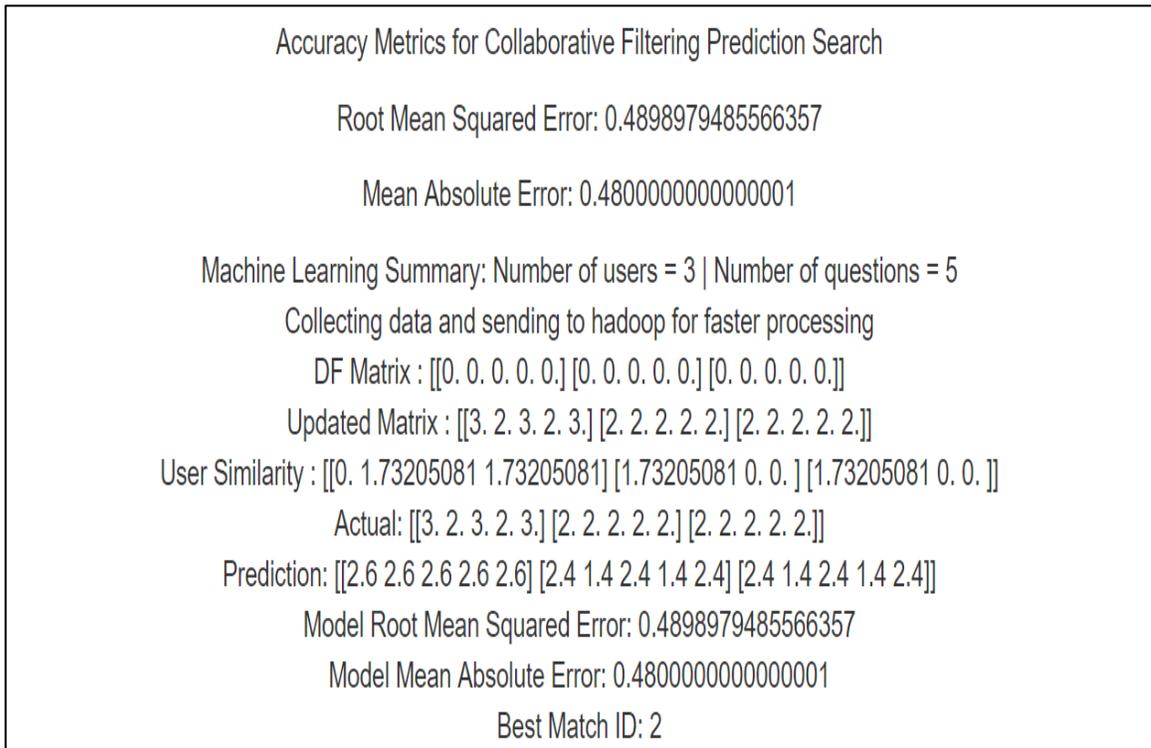


Fig. 5.11 Machine learning statistics on IMSS interface

5.3.7. Page Ranking and User Rating

The output window of recommended metasearch tool shows web links and information with headers priority or rank calculated using ratings provided by the previous user and default rank offered by the search engine, Web Link with search engine stamp, G- Google, B- Bing and Q- Qwant to show the search engine used to retrieve the web link. The tool interface in Fig. 5.12 shows information regarding response time, security and page loading speed. The feedback field allows the user to give the rating between 1 to 5 to depict the relevance of the web link to satisfy the personalized search needs of the user. The developer console may be used to track the flow of execution as shown in Fig. 5.13. Further, the web link selection and page visit are illustrated in Fig. 5.14 and Fig. 5.15.

6	https://www.samsung.com/uk/laundry/washing-machine/ (G)	1.18 ms	Secure	1.19 ms	5 ▾ Rating is saved as 5
7	https://www.youtube.com/watch?v=Ofx5JLN9ZEs (G)	1.18 ms	Secure	1.20 ms	1 ▾
1	https://www.wisegeek.com/what-is-an-high-efficiency-washing-machine.htm (B)	2.03 ms	Secure	2.14 ms	1 ▾
2	http://vcusedappliances.com/ (B)	2.03 ms	Not secure	2.12 ms	1 ▾
3	https://www.washingtonpost.com/news/the-switch/wp/2016/11/04/samsung-officially-recalls-top-loading-washers-over-a-month-after-government-warning/ (B)	2.03 ms	Secure	2.10 ms	1 ▾

Fig. 5.12 Page ranking and user rating

http://link.springer.com/content/pdf/bfm%3A978-981-10-2750-5%2F1.pdf (G)	1.64 ms Not secure 1.66 ms 1 ▾
http://vips.edu/wp-content/uploads/2016/09/Special-Issue-VJR-conference-2018.pdf (G)	2 1.64 ms Not secure 1.66 ms 1 ▾

Developer Console (Elements, Console, Sources):

```

<!doctype html>
<html lang="zxx" class="gr_34_73_42_123"> == $0
  <head>...</head>
  <body style="padding:5px;margin-bottom:10px;" data-gr-c-s-loaded="true">
    <header name="Access-Control-Allow-Origin" value="*"...>
  </body>
  <span class="gr_tooltip">...</span>
</html>

```

html_gr_34_73_42_123 body

Styles Event Listeners DOM Breakpoints Properties Accessibility

- html.gr_34_73_42_123
- HTMLHtmlElement
- HTMLElement

Fig. 5.13 Search results and developer console

8	https://dl.acm.org/citation.cfm?id=2979782 (G)	1.64 ms	Secure	1.64 ms	1 ▼
9	https://www.sciencedirect.com/science/article/pii/S1319157818307869 (G)	1.64 ms	Secure	1.64 ms	1 ▼
1	https://www.researchgate.net/publication/310361044_IMSS-E_An_Intelligent_Approach_to_Design_of_Adaptive_Meta_Search_System_for_E_Commerce_Website_Ranking (Q,G)	2.35 ms	Secure	2.37 ms	1 ▼
2	https://www.amazon.in/Data-Structures-Program-Design-Using/dp/1683922077 (Q)	2.35 ms	Secure	2.36 ms	1 ▼
3	https://www.sciencedirect.com/science/article/pii/S1319157818307869 (Q,G)	2.35 ms	Secure	2.36 ms	1 ▼

Fig. 5.14 Web link selection and page attributes listing

→ <https://www.sciencedirect.com/science/article/pii/S1319157818307869>

Outline Download Share Export

**Journal of King Saud University -
Computer and Information Sciences**

Available online 27 November 2018

open access

In Press, Corrected Proof

IMSS-P: An intelligent approach to design & development of personalized meta search & page ranking system

Dheeraj Malhotra , O.P. Rishi

Show more

<https://doi.org/10.1016/j.jksuci.2018.11.013> [Get rights and content](#)

Fig. 5.15 Web page visit via IMSS tool

5.3.8. Various Tables in IMSS

The deployed IMSS tool uses MySQL engine to maintain multiple tables. There are eight different tables including *Profession*, *Query*, *User*, *User_Query*, *User_Rating*, *Question*, *Response*, and *Similarity Metrics*. The detailed structural description and various record instances within multiple tables of the IMSS tool are shown from Fig. 5.16 to Fig. 5.23.

```
MariaDB [(none)]> use imss;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [imss]> show tables;
+-----+
| Tables_in_imss |
+-----+
| profession      |
| query           |
| question        |
| response        |
| similaritymetrics |
| user            |
| user_query      |
| user_rating     |
+-----+
8 rows in set (0.00 sec)

MariaDB [imss]> |
```

Fig. 5.16 Tables in IMSS

```

MariaDB [imss]> select * from user;
+-----+-----+-----+-----+-----+-----+
| id | name | email | mobile | password | encrypt_password |
| otp | type | profession |
+-----+-----+-----+-----+
| 1 | ADMIN | ADMIN | NULL | ADMIN | NULL |
| NULL | ADMIN | NULL |
| 2 | user1 | user1@gmail.com | 9654785857 | user1 | 24c9e15e52afc47c225b757e7bee1f9d |
ED | 49429 | USER | 1 |
| 3 | user2 | user2@gmail.com | 9874585214 | user2 | 7e58d63b60197ceb55a1c487989a3720 |
ED | 00528 | USER | 1 |
| 4 | user3 | user3@gmail.com | 9854741253 | user3 | 92877af70a45fd6a2ed7fe81e1236b78 |
ED | 38168 | USER | 1 |
| 5 | user4 | user4@gmail.com | 9658745852 | user4 | 3f02ebe3d7929b091e3d8ccfde2f3bc6 |
ED | 51037 | USER | 1 |
| 6 | Dheeraj Malhotra | dheerajmalhotra4@gmail.com | 9811949964 | dheeraj | d9c59c1aa00818c948423d8d9f141f30 |
ED | 31596 | USER | 1 |
| 7 | Dr OP Rishi | omprakashrishi@yahoo.com | 9414258030 | oprishi | 062692095832bf671ab6cfacfce447f6 |
ED | 27450 | USER | 1 |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

```

Fig. 5.17 Structural description about user table

```

imsscloud@imss: ~ - Google Chrome
https://ssh.cloud.google.com/projects/imss-project/zones/us-east1-b/instances/imss?authuser=0&hl=en_US&projectNumber=85666484857
21 | dheerajmalhotra4@gmail.com | 1 | 1 | 3 | 6 |
22 | dheerajmalhotra4@gmail.com | 1 | 2 | 3 | 6 |
23 | dheerajmalhotra4@gmail.com | 1 | 2 | 3 | 6 |
24 | dheerajmalhotra4@gmail.com | 1 | 3 | 3 | 6 |
25 | dheerajmalhotra4@gmail.com | 1 | 4 | 3 | 6 |
26 | dheerajmalhotra4@gmail.com | 1 | 5 | 3 | 6 |
27 | omprakashrishi@yahoo.com | 1 | 1 | 3 | 7 |
28 | omprakashrishi@yahoo.com | 1 | 2 | 3 | 7 |
29 | omprakashrishi@yahoo.com | 1 | 3 | 3 | 7 |
30 | omprakashrishi@yahoo.com | 1 | 4 | 3 | 7 |
31 | omprakashrishi@yahoo.com | 1 | 5 | 3 | 7 |
+-----+-----+-----+-----+
31 rows in set (0.00 sec)

MariaDB [imss]> desc similaritymetrics;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| userId | text | YES | | NULL | |
| searchString | text | YES | | NULL | |
| searchId | text | YES | | NULL | |
| searchEngineValue | text | YES | | NULL | |
| term | text | YES | | NULL | |
| titlecount | text | YES | | NULL | |
| summaryCount | text | YES | | NULL | |
| linkCount | text | YES | | NULL | |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

```

Fig. 5.18 Structural description about similarity metrics table

```

imsscloud@imss: ~ - Google Chrome
https://ssh.cloud.google.com/projects/imss-project/zones/us-east1-b/instances/imss?authuser=0&hl=en_US&projectNumber=85666484857
1494 rows in set (0.00 sec)

MariaDB [imss]> desc user_query;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | NO   | PRI | NULL     | auto_increment |
| qid        | text      | YES  |     | NULL     |              |
| feedback   | text      | YES  |     | NULL     |              |
| feedbackvalue | text      | YES  |     | NULL     |              |
| useremail  | text      | YES  |     | NULL     |              |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

MariaDB [imss]> select * from user_query;
+-----+-----+-----+-----+-----+
| id | qid | feedback | feedbackvalue | useremail |
+-----+-----+-----+-----+-----+
| 1  | 13  | 5         | 5              | user1@gmail.com |
| 2  | 36  | 5         | 5              | user1@gmail.com |
| 3  | 36  | 5         | 5              | user2@gmail.com |
| 4  | 36.0 | 5         | 5              | user3@gmail.com |
| 5  | 120 | 5         | 5              | user2@gmail.com |
| 6  | 187 | 5         | 5              | user3@gmail.com |
| 7  | 254.0 | 5         | 5              | dheerajmalhotra4@gmail.com |
| 8  | 808.0 | 5         | 5              | dheerajmalhotra4@gmail.com |
| 9  | 806.0 | 5         | 5              | dheerajmalhotra4@gmail.com |
| 10 | 805.0 | 5         | 5              | dheerajmalhotra4@gmail.com |
| 11 | 542.0 | 5         | 5              | dheerajmalhotra4@gmail.com |
+-----+-----+-----+-----+-----+

```

Fig. 5.19 Structural description about user_query table

```

MariaDB [imss]> desc user_rating;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | NO   | PRI | NULL     | auto_increment |
| profession_id | int(11)   | YES  |     | NULL     |              |
| uid        | int(11)   | YES  |     | NULL     |              |
| query_id   | int(11)   | YES  |     | NULL     |              |
| query_title | text      | YES  |     | NULL     |              |
| result_id  | int(11)   | YES  |     | NULL     |              |
| result_title | text      | YES  |     | NULL     |              |
| result_link | text      | YES  |     | NULL     |              |
| rating     | text      | YES  |     | NULL     |              |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

```

Fig. 5.20 Structural description about user_rating table

```

MariaDB [imss]> desc profession;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)| NO   | PRI | NULL    | auto_increment|
| name  | text   | YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.04 sec)

MariaDB [imss]> select * from profession;
+----+-----+
| id | name  |
+----+-----+
| 1  | Teacher |
| 2  | Doctor |
+----+-----+
2 rows in set (0.00 sec)

MariaDB [imss]> █

```

Fig. 5.21 Structural description about profession table

```

https://ssh.cloud.google.com/projects/imss-project/zones/us-east1-b/instances/imss?authuser=0&hl=en_US&projectNumber=85666484857
MariaDB [imss]> desc question;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int(11)| NO   | PRI | NULL    | auto_increment|
| profession_id  | int(11)| YES  |     | NULL    |               |
| profession_name| text   | YES  |     | NULL    |               |
| question       | text   | YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

MariaDB [imss]> select * from question;
+----+-----+-----+-----+
| id | profession_id | profession_name | question |
+----+-----+-----+-----+
| 1  | 1 | Teacher | I am a good decision maker. |
| 2  | 1 | Teacher | My decisions are usually accepted as good by the person affected. |
| 3  | 1 | Teacher | When faced with an important decision, I am not overly anxious about making a wrong choice. |
| 4  | 1 | Teacher | My friends and co-workers ask me for help in making important decisions. |
| 5  | 1 | Teacher | I make a decision and act rather than worrying about the alternatives and becoming tense. |
+----+-----+-----+-----+

```

Fig. 5.22 Structural description about question table

```

imsscloud@imss: ~ - Google Chrome
https://ssh.cloud.google.com/projects/imss-project/zones/us-east1-b/instances/imss7authuser=0&hl=en_US&projectNumber=85666484857
-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
MariaDB [imss]> select * from response;
-----+-----+-----+-----+-----+-----+
| id | user_id | profession_id | question_id | question_score | uid |
-----+-----+-----+-----+-----+-----+
| 1 | user1@gmail.com | 1 | 1 | 3 | 2 |
| 2 | user1@gmail.com | 1 | 2 | 3 | 2 |
| 3 | user1@gmail.com | 1 | 3 | 3 | 2 |
| 4 | user1@gmail.com | 1 | 4 | 3 | 2 |
| 5 | user1@gmail.com | 1 | 5 | 3 | 2 |
| 6 | user2@gmail.com | 1 | 1 | 3 | 3 |
| 7 | user2@gmail.com | 1 | 2 | 2 | 3 |
| 8 | user2@gmail.com | 1 | 3 | 2 | 3 |
| 9 | user2@gmail.com | 1 | 4 | 2 | 3 |
| 10 | user2@gmail.com | 1 | 5 | 1 | 3 |
| 11 | user3@gmail.com | 1 | 1 | 3 | 4 |
| 12 | user3@gmail.com | 1 | 2 | 3 | 4 |
| 13 | user3@gmail.com | 1 | 3 | 3 | 4 |
| 14 | user3@gmail.com | 1 | 4 | 3 | 4 |
| 15 | user3@gmail.com | 1 | 5 | 3 | 4 |
| 16 | user4@gmail.com | 1 | 1 | 3 | 5 |
| 17 | user4@gmail.com | 1 | 2 | 3 | 5 |
| 18 | user4@gmail.com | 1 | 3 | 2 | 5 |
| 19 | user4@gmail.com | 1 | 4 | 3 | 5 |
| 20 | user4@gmail.com | 1 | 5 | 3 | 5 |
| 21 | dheerajmalhotra4@gmail.com | 1 | 1 | 3 | 6 |
| 22 | dheerajmalhotra4@gmail.com | 1 | 2 | 3 | 6 |
-----+-----+-----+-----+-----+-----+

```

Fig. 5.23 Structural description about response table

5.4. CHAPTER SUMMARY

This chapter discusses three phases of the deployed system design. Phase 1 addresses best match prediction using logistic regression or collaborative filtering based machine learning model. Phase 2 is about query disambiguation and is helpful when the user query is incomplete or ambiguous. Phase 3 discusses website re-ranking process in detail using ACVPR algorithm and flowchart. Moreover, map and reduce methods for keywords frequency calculation to determine CRV is also elaborated. The interface of the IMSS tool deployed to assess the effectiveness of the pioneered ACVPR algorithm is discussed in detail with the help of various screenshots. At last, the chapter discusses the MYSQL engine based database design and structural aspects of various relations used in the deployment of IMSS tool.

CHAPTER 6

PREDICTIVE ANALYTICS

6.1. INTRODUCTION

This chapter discusses the fundamentals of machine learning aspects employed in the current research work. The pioneered Advanced Cluster Vector Page Ranking (ACVPR) algorithm and Intelligent Meta Search System (IMSS) tool for web search personalization employs machine learning techniques like logistic regression and collaborative filtering using deployment platforms like R and Python in the present research work. These machine learning techniques assist in finding the existing best match user within the IMSS system. This information, in turn, is useful in predicting incomplete or erroneous search queries by referring search history of the matching user as predicted by the ACVPR algorithm. This chapter further discusses various features of the Python language used to support machine learning framework implemented within current research work. The details about logistic regression and collaborative filtering based machine learning models and the calculation of various relevant evaluation metrics are discussed in chapter 7.

6.2. PYTHON FOR IMSS TOOL DEPLOYMENT

Python is a high level, object-oriented and scalable programming language. Python can deliver both (i) Robustness similar to conventional compiled languages and, (ii) Ease of use identical to interpreted and scripting languages. There are several reasons to use Python to implement machine learning capabilities in the implementation of Intelligent Meta Search System (IMSS) tool within current research work:

- Python can efficiently use Hadoop platform to code and implement map and reduce functions to effectively and efficiently handle big data retrieved in terms of a massive number of web links, web title, and summary by all three background search engines, i.e., Google, Qwant and Bing. These vast data is returned for analysis by the metasearch tool, i.e., IMSS within current research work
- The mathematical and intelligent models such as clustering based collaborative filtering model used to identify the best match existing User ID in the implemented IMSS system for query disambiguation can be easily applied using open source *Scikit-Learn* library well supported by *Scipy* and *Numpy* libraries available in Python
- Python's capability to easily connect with MySQL databases allow processing of data in the tabular format with relative ease. The implementation of innovative ACVPR algorithm and IMSS tool in the current research work requires various databases to be maintained using MySQL and Python
- Python provides excellent data analysis capabilities. It gives a quite powerful data analysis package, i.e., *Pandas* which in turn reduces the time gap between the starting and completion of processing by an analytics tool like IMSS within current research work
- Python provides *the matplotlib* library to quickly and effectively visualize data and results. The experimental analysis of the current research work leads to the generation of various plots using Python to find a user's correlation for search query recommendation and behavior analysis for finding existing best match user ID of IMSS tool

6.3. LEVELS OF ANALYTICS

The analytics to process user's search data is an essential component of the proposed and implemented the meta-search system. Here, in the current research work, predictive analytics is deployed to personalize search results. The process of analytics may be viewed to exhibit the following levels:

- Level 1: Descriptive Analytics
- Level 2: Diagnostic Analytics
- Level 3: Predictive Analytics
- Level 4: Prescriptive Analytics

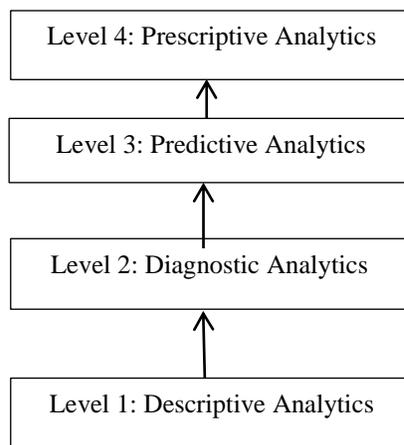


Fig. 6.1 Levels of analytics

6.3.1. Level 1- Descriptive Analytics

The various levels of analytics are shown in Fig. 6.1. The focus of descriptive analytics is to describe “*What has happened?*” Any procedure or activity that may lead to the conversion of raw facts or data into a meaningful and user interpretable form is termed as *descriptive analytics*. For instance, *statistics* fall under this category as it employs various mathematical functions like sum, count, mean, etc. to summarize data.

6.3.2. Level 2- Diagnostic Analytics

The focus of diagnostic analytics is to describe “*Why did it happen?*” Any procedure or activity that lead to drill down the root cause of observation is termed as *diagnostic analytics*. For instance, *data mining* falls under this category as it employs various techniques to determine the cause of an observation. There are multiple tools to implement diagnostic analytics, for instance, Excel, Spot fire, etc.

6.3.3. Level 3- Predictive Analytics

The focus of predictive analytics is to describe “*What will happen?*” Any procedure or activity that analyzes data to predict what will happen in the future is termed as *diagnostic analytics*. For instance, *machine learning* falls under this category as it employs various techniques to analyze the data and to predict the probable outcome in the future. The quality of prediction is dependent on multiple factors including the quality of the input data, and hence predictive analytics can't predict the future with 100% accuracy. The deployed metasearch system, i.e., IMSS implements predictive analytics for the following purposes:

- Prediction of a best match User ID for search query expansion or disambiguation using collaborative filtering based machine learning model
- Prediction of response to various questions used for behavior analysis by pioneered ACVPR algorithm and IMSS tool using collaborative filtering based machine learning model
- Prediction of feedback for a prospective web link listed corresponding to user's search query and hence in predicting correct personalized rank of a specific web link in the output of the implemented metasearch tool, i.e., IMSS using logistic regression based machine learning model within current research work

6.3.4. Level 4- Prescriptive Analytics

The focus of prescriptive analytics is to ensure “*How do you make the best will happen?*” Any procedure or activity that provides the optimized course of action for solving a given

problem is termed as *prescriptive analytics*. For instance, *deep learning* falls under this category as it monitors all performance oriented key metrics to ensure the best outcome.

6.4. MACHINE LEARNING

Machine learning is a field of computer science that deals with the development of techniques and algorithms that possess the capability to learn from data and can make predictions. The machine learning enabled system can improve their effectiveness and efficiency in prediction over time and repeated usage.

Machine learning assist in determining a response variable from a dependent variable, for instance, consider a linear regression model to represent a straight line,

$$Y = mX + C$$

Here, X is a dependent variable and Y is a response variable.

6.4.1. Types of Machine Learning

The present research work uses Supervised and Unsupervised Machine Learning techniques for predictive analytics as depicted in Fig. 6.2 and discussed below:

6.4.1.1. Supervised Machine Learning

The overall goal of the machine learning algorithm is to learn the patterns from datasets with a large set of input and the corresponding output. These datasets are prepared under the supervision of a subject expert. Hence, learning here is termed as supervised learning. The deployed metasearch system within current research work uses supervised machine learning for the implementation of logistic regression based model to predict user feedback about a prospective web link listed corresponding to the search query of the user and hence assist in predicting correct rank of the web link to satisfy the personalized search needs of the user. The details regarding generation, training, and testing of the

supervised regression based machine learning model used by the pioneered IMSS tool is discussed in section 7.3.1.

The supervised algorithm after learning the patterns from input dataset can map the real-time input into a corresponding class of events or outputs. The overall objective is to derive the function:

$$Q = F(P)$$

Where P is the input vector and Q is the required output. The mathematical equation or function is commonly referred to as a Model.

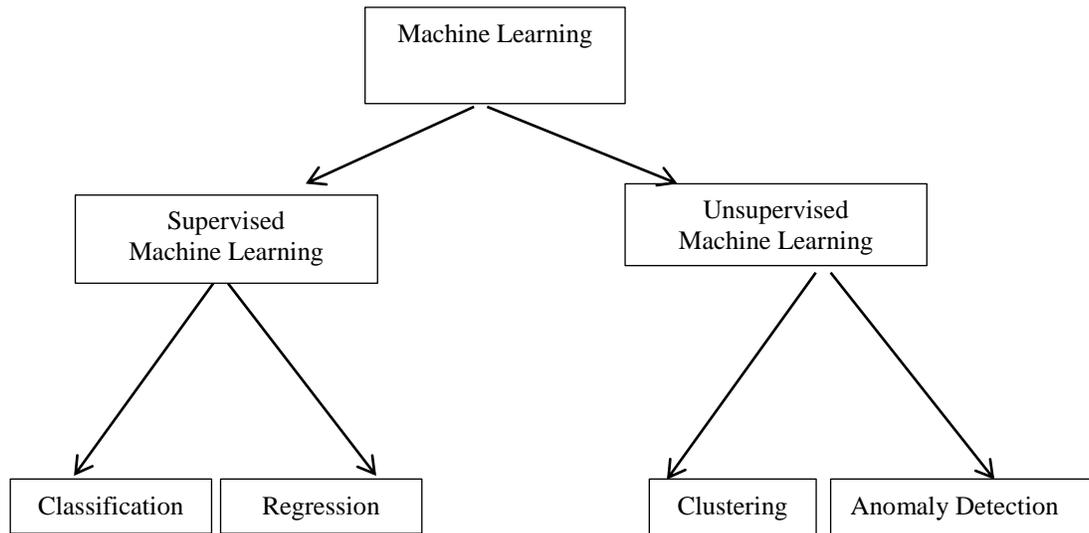


Fig. 6.2 Machine learning techniques

The supervised learning may be further categorized into two subtypes:

(i) *Regression*

(ii) *Classification*

(i) Regression

If the derived function, $Q = F(P)$ has continuous output, for instance, prediction of some car parking required for a new shopping mall or prediction of sales by an E-Commerce website.

(ii) Classification

If the derived function, $Q = F(P)$ has discrete output, i.e., the number of classes that may be predicted can be two or sometimes more than two. For instance, spam filtering by an email server, a mail may be classified into any of the following two categories, i.e., Spam or Non-Spam. Similarly, listed web link in the output of the IMSS tool is determined as relevant or not relevant using classification based supervised learning.

The supervised learning process may be summarized via the following three steps:

(i) Training of Algorithm

The supervised algorithm will be provided with a massive number of input datasets and their corresponding outputs. These datasets are prepared under the supervision of a subject expert. The algorithm will learn the patterns from the input dataset and express the same using a statistical function of the form, $Q = F(P)$. The derived function is commonly referred to as a model.

(ii) Testing of Model

The derived model will be validated by using a portion of input datasets that were not used in the training phase. The effectiveness of the model in predicting the correct class or output from a given input may be validated using the same.

(iii) Prediction by the Model

The trained and tested model may now be used for the prediction of the

outcomes on real-time datasets.

The current research work uses datasets feedback.csv for training, and testing of deployed supervised model as discussed in chapter 7.

6.4.1.2. Unsupervised Machine Learning

In unsupervised learning, the corresponding output for a given input within the historical datasets is unknown. The overall goal of an unsupervised algorithm is to study and learn the exciting patterns within the input datasets. However, these algorithms do not require any assistance of a subject expert and hence are termed as unsupervised. These algorithms enable the machine to learn more complex problems as compared to supervised learning.

Unsupervised learning may be further categorized into two subtypes:

(i) Clustering

The process of partitioning input datasets into subsets known as clusters where items within one cluster are similar and are dissimilar from items within other clusters. It may be noted that output classes or clusters in unsupervised learning are not known in advance. For instance, a grouping of customers in a retail store by their profile can be accomplished via clustering.

In the present research work, a popular technique, K means is used to find clusters of similar users. Here, the objective is to assure high intra-cluster similarity between various users of the implemented metasearch system and low inter-cluster similarity between them. The user is required to answer a few questions based on decision-making skills during the registration process. The answers given by a specific user of a given profile is used to determine similar users belonging to the same cluster. In the first step, the centroid score is assumed for each cluster, and various users who are having centroid score

near to the centroid are then assigned to the specific cluster. The centroid score is then recalculated as mean or average of all the user scores assigned to that particular cluster and then again various users are assigned to multiple clusters. The process is repeated until the convergence of all clusters is observed. The distance between centroid score and user score is expressed as Euclidean distance and is calculated as shown in equation (1):

$$d = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad \dots\dots (1)$$

(ii) *Anomaly Detection*

It is commonly termed as *outlier detection*. It is used to identify observations which do not conform to the usual behavior of items within a given dataset. It has applicability in a large number of domains such as fraud detection etc.

Note: In the pioneered ACVPR algorithm and IMSS tool within the current research work, unsupervised machine learning based on the clustering is deployed to group various users of the metasearch tool by the similarity in their behavior analysis and decision making skills as retrieved from their profiles. The keywords within the search strings of similar users are used to identify search recommendations for the new user. The similarity between various users is calculated by using collaborative filtering based recommender system. The detail of the collaborative filtering based recommender system is included within the next section.

6.5. RECOMMENDER SYSTEMS

Recommender systems are information filtering systems. They are used to predict user preferences to satisfy the personalized user requirements. For instance, E-Commerce companies like *Amazon* usually recommend products to the customer based on the recent purchase or items within the cart of the customer. There are two types of Recommender Systems:

- Content-Based Filtering
- Collaborative Filtering

The content filtering based recommender systems emphasize recommendations based on a similar category of the items. For instance, if a user purchases one item from a specific group, then another item from the same group will be recommended to the user. The idea of content-based filtering is shown in Fig. 6.3.

The collaborative filtering based recommender systems focus on user-based similarity. For instance, if a user likes or purchases some item, then another similar user will be recommended a similar item. The idea of collaborative filtering is shown in Fig. 6.4.

There are two types of collaborative filtering techniques (<http://www10.org/> [155]):

- Memory-based collaborative filtering
- Model-based collaborative filtering

The memory-based collaborative filtering is based on creating a ranked list of items preferred by similar users. These items can be sorted and are to be shown to the similar user.

The model-based collaborative filtering is established on matrix factorization.

In the present research work, memory-based collaborative filtering is deployed to determine the ranked list of web links to be displayed to the new user of the pioneered

IMSS tool. The database of the tool maintains the various rankings given by other similar user and give preference to high ranked links by the previous user to display the improved order of web page ranking.

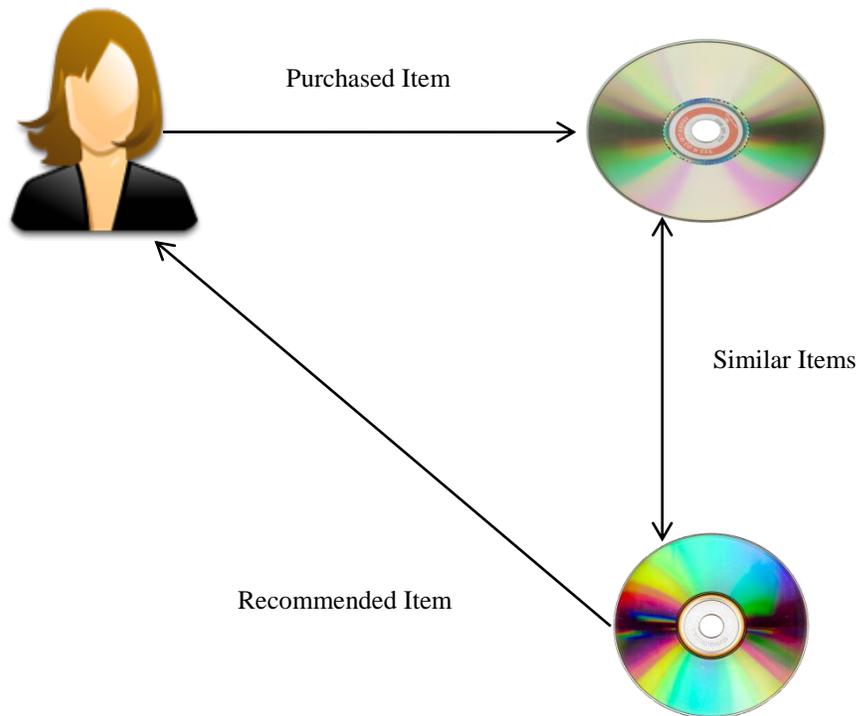


Fig. 6.3 Content-based filtering

6.6. KNOWLEDGE DISCOVERY PROCESS

The process of retrieving information from data is often known as ‘Knowledge Discovery Process’ (KDP). This process comprises various steps as highlighted in Fig. 6.5. First of all, data is collected through implicit or explicit ways. In the current research work, user information during the signup process is gathered explicitly. However, on the other hand, time spent by a user on a website to determine relevancy is implicit data collection and is used to determine Time Relevancy Vector (TRV) used by innovative ACVPR algorithm within current research work. Selection process means out of multiple columns of data;

not every column is meaningful for knowledge discovery. Hence, suitable columns are selected that can determine the response variable. For instance, here in the current research work, recast regression based machine learning model is deployed by selectively removing response time column as discussed in detail within section 7.3.1.

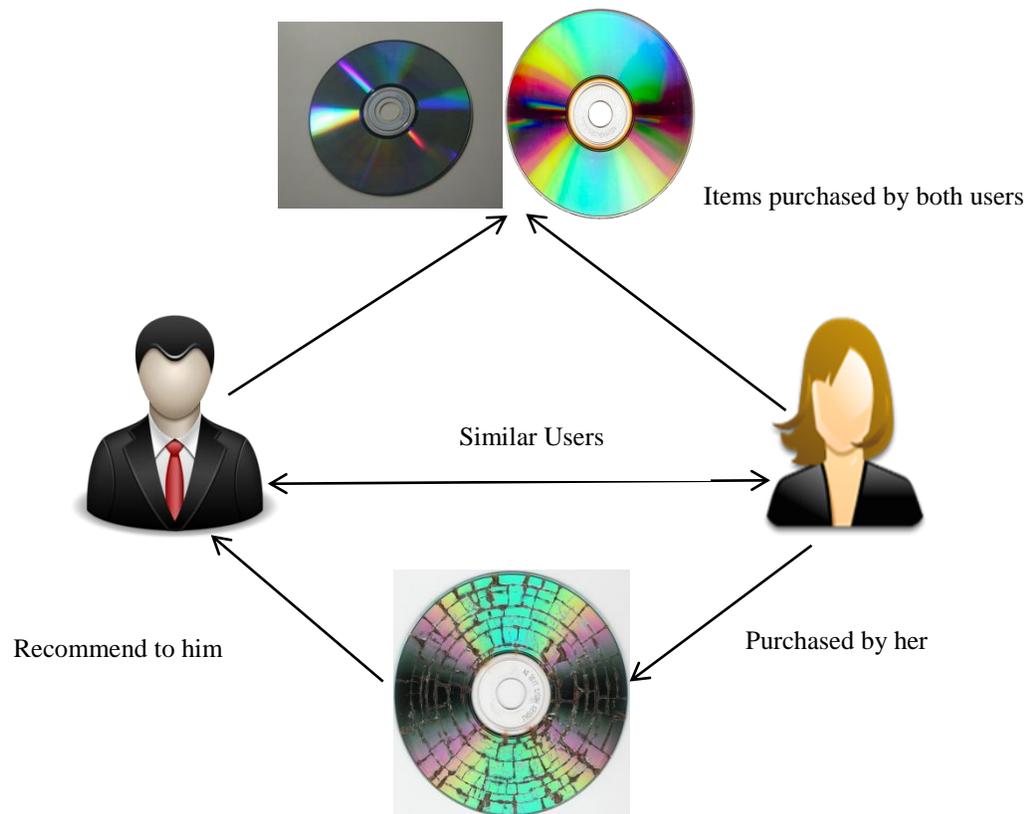


Fig. 6.4 Collaborative filtering

The data will be put into the data frames using Pandas package of Python, and then the correlation matrix may be calculated to assess the accuracy of the selection process. Here in the present research work correlation matrix is calculated and plotted for various users of the system to represent the relationship between actual and predicted responses for different questions used for behavior analysis by the pioneered metasearch tool, i.e.,

IMSS during the signup process. Preprocessing, refer to cleaning or determination of missing data. Moreover, one can't perform machine learning on the textual data. The data is first needed to be converted into the numbers as machine learning includes mathematical equations that can accept and process numbers only.

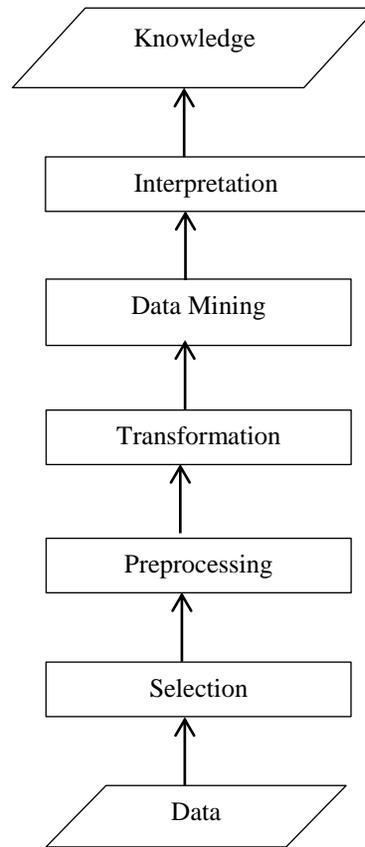


Fig. 6.5 Knowledge discovery process

Transformation means standardization or normalization of the data as there may be one column. The data with a considerable variation may be represented into a limited number of forms, e.g., in digital logic, a single value may describe a range of voltage, e.g., -3V to -5 V is 0 while +3 V to +5 V is 1. The explicit steps of preprocessing and transformation

are not required by the IMSS tool as feedback data collected from the user of the deployed tool is already in numeric and normalized form

Data mining sub-step uses an algorithm to determine the response variable from the dependent variable by calculating coefficients etc. Here, in the current research work, big data analytics using Hadoop2 framework is deployed as the number of web links returned and to be processed with real-time response requirement are quite huge. The three background search engines, i.e., Google, Qwant, and Bing are used for searching and returning a vast number of web links to metasearch tool, i.e., IMSS to improve the overall recall metric of the deployed system.

In the interpretation step, various plots are drawn to learn the difference between actual and predicted values to resolve errors like Mean Absolute Error, Root Mean Square Error, etc. The error calculation is done within the interpretation and evaluation phase, and finally, the low value of errors in prediction represent useful and significant knowledge discovery. The current research work uses *the matplotlib* library of Python and R statistical tool to plot Receiver Operating Characteristics (ROC) curves, diagnostic plots and various plots to represent evaluation metrics of both types of deployed machine learning models as discussed in detail within chapter7.

6.7. MACHINE LEARNING FRAMEWORK

The features provided by the Python to implement various steps of Knowledge Discovery Process (KDP) as mentioned in section 6.6 are as follows:

- *Storage*: Data is required to be stored in a structured or unstructured format
- *Reading*: Data may be read into the data frames using Pandas package of the Python
- *Selection*: Response variable and dependent variables are required to be selected, and independent variables will be ignored. There are two main methods to

determine various types of variables, i.e. (i) Visual method using correlation matrix (ii) Mathematical method using correlation coefficients such as the Pearson correlation coefficient. It may be noted that correlation greater than +0.8 (Positive correlation or directly proportional) and lesser than -0.8 (Negative correlation or inversely proportional) is considered as significant

- *Preprocessing*: To perform preprocessing on data, following methods may be used (a) `delna ()` function of Python may be used to delete null values (b) `fillna ()` function of Python may be used to fill various null values using mean, median, etc.
- *Conversion*: The conversion of textual data into a numerical format to enable machine learning model may be carried out using

(a) *Python: Label Binarizer* for conversion to a binary category

(b) *Python: Label Encoding* for manual conversion to more than two numerical category

(c) *Python: One Hot Encoding*: It automatically creates a matrix with a large number of 0's and 1's where the category is available and is usually good while dealing with a large number of categories

- *Transformation*: Perform the transformation on data to map large values into small values to visualize them on the graphs easily. The sub-steps of transformation are: (i) Standardization (ii) Normalization. There are various functions available in the Python for calculation of both of these metrics via Pandas object using which one can easily calculate standardization and normalization. The syntax to use various Python functions for calculation of both of these metrics as per their formulae is as follows:

```
Pmean = pd ['column name'].mean ()
```

Standard Deviation = pd ['column name'].std ()

Pmin = pd ['column name'].min ()

Pmax = pd ['column name'].max ()

Standardization = (P – Pmean) / Standard Deviation

Normalization = (P – Pmean) / (Pmax – Pmin)

- *Data Mining*: The data mining algorithms may be deployed on the existing frameworks, e.g., *linear regression* framework and the object *lm* can be used to predict response variable *Y* from dependent variable *X*, while performing mining operation, *scipy* library is used for variable equations available in it. However, *scipy* require a mathematical library *numpy* to convert these equations into polynomials
- *Interpretation/Evaluation*: This phase may be implemented in various formats, such as (a) Mathematical where one can calculate accuracy, precision, recall, MAE, RMSE, etc. (b) Visual Interpretation, i.e., Graphs by generating plots using the *matplotlib* library in Python

6.8. CHAPTER SUMMARY

This chapter discusses the basics of machine learning frameworks deployed within the present research work. The detailed discussion about various forms of analytics, recommender systems, types of machine learning is also included in this chapter. This chapter further discusses the Knowledge Discovery Process (KDP) and multiple features of Python language to support machine learning framework implemented within the pioneered ACVPR algorithm and IMSS tool.

CHAPTER 7

EXPERIMENTAL EVALUATION AND GRAPHICAL ANALYSIS

7.1. INTRODUCTION

This chapter discusses the evaluation of the effectiveness and efficiency of the pioneered Advanced Cluster Vector Page Ranking (ACVPR) algorithm and Intelligent Meta Search System (IMSS) tool for web search personalization within current research work. The implementation and comparison of two deployed machine learning models based on logistic Regression and collaborative filtering and calculation of their corresponding evaluation metrics are also discussed in detail. The chapter also discusses the comparison of the pioneered approach with baselines, comparison of IMSS tool with popular recommendation approaches and professional metasearch engines.

7.2. DATASETS

The current research work uses two types of datasets for experimental evaluation (i) Google Zeitgeist (ii) Feedback.CSV. The source and relevance of used datasets are discussed in detail as follows.

7.2.1. Google Zeitgeist or Google Trends 2017

In present research work, volunteers need to search for queries to compare the personalized search precision of pioneered Intelligent Meta Search System (IMSS) tool with popular search engines like Google, Qwant, Bing, and Dogpile. The Google Zeitgeist is public and is the most widely accepted query set by the community. The

Google Trends is published annually and is a repository of most popular search queries. The Google Trends is chosen because of two reasons (i) The available queries are searched by real users (ii) These are most popular queries, and hence the personalization implemented by search engines like Google on these queries is most assured. Therefore, the personalized search precision of the recommended approach by the current research work can be easily compared with professional and popular search engines.

Table 7.1 shows the chosen top 10 and a total of 90 search queries of 9 different categories as available on Google Trends for the calendar year 2017 in India for experiments within current research work. The Google Trends from India is preferred because of the familiarity of volunteers to judge the personalized precision of page ranking of their favorite search category as shown in response by deployed IMSS tool and professional search engines under consideration. Fig. 7.1 and Fig. 7.2 show top 5 search queries of each of the two sample categories of “How to...” and “What is...” on Google Trends 2017 search query databases in India.

Table 7.1 Categories of search queries from Google Trends 2017 in India

S. No.	Category	Examples of Queries	Number of Queries
1	News	UP Election Results, CBSE Results	10
2	Sports	Indian Premier League, Pro Kabaddi	10
3	Movies	Dangal, Munna Michael	10
4	Songs	Raabta, Hawa Hawa	10
5	Entertainers	Sunil Grover, Arshi Khan	10
6	Near Me	Coffee Shops Near Me, Post Offices Near Me	10

7	What Is	What is Bitcoin, What is Jallikattu	10
8	How To	How to buy Bitcoin, How to book Jio Phone	10
9	Overall	Bahubali 2, Live Cricket Score	10

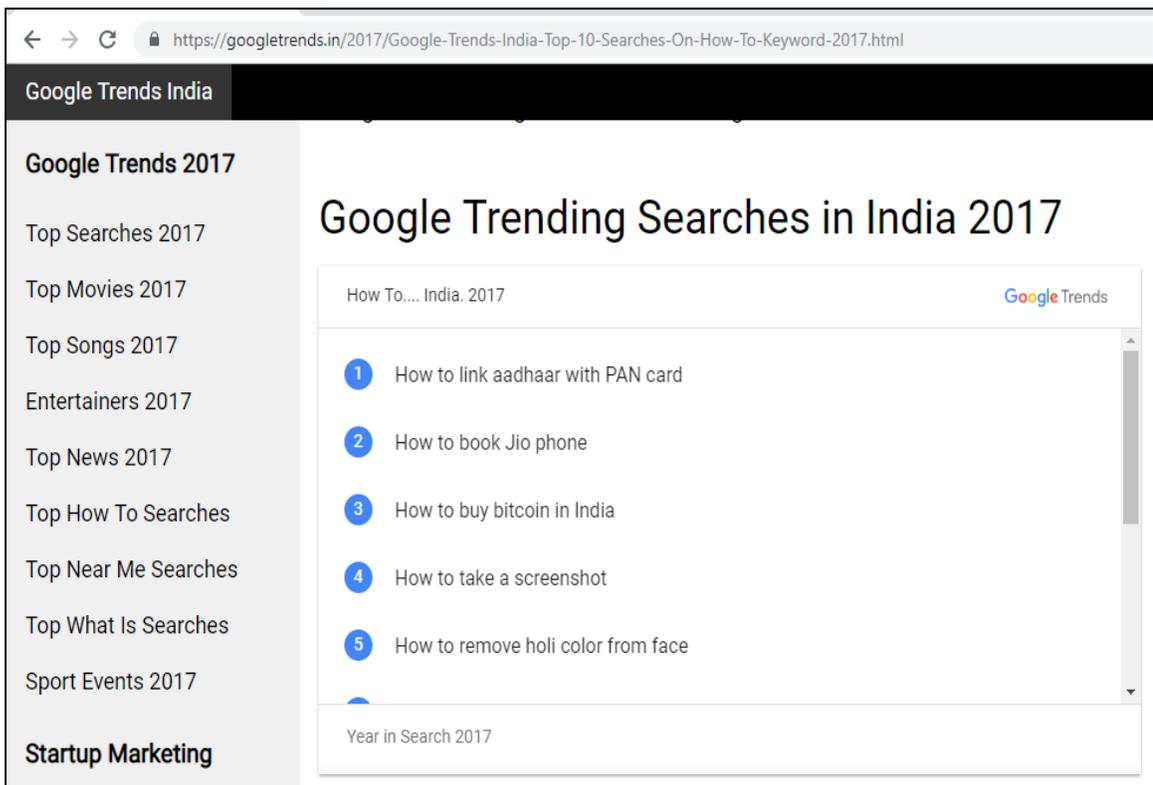


Fig 7.1 Top search queries in the category of “How to” on Google Trends

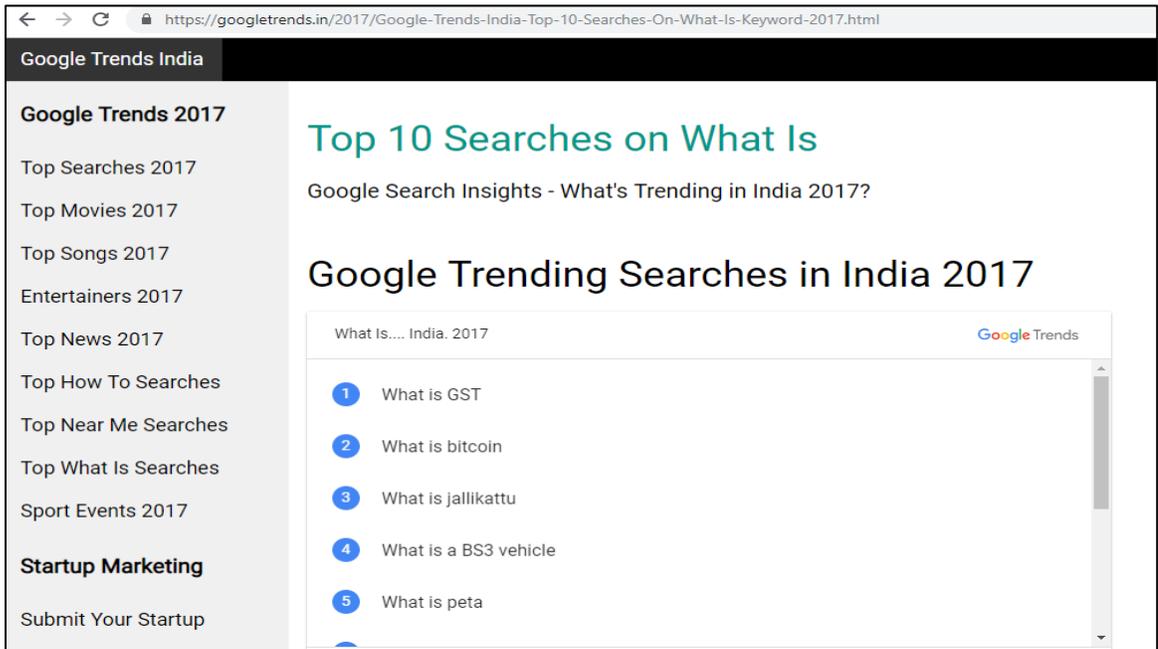


Fig 7.2 Top search queries in the category of “What is” on Google Trends

7.2.2. Feedback.CSV

In order to predict the personalized preferences of the user, one of the machine learning models deployed by the current research work is based on logistic regression and require user’s feedback.csv file to get evaluated and implemented using R statistical tool. The feedback.csv file is generated by the help of volunteers as discussed in section 7.6.2. The .csv format file consists of the real search data of 110 volunteers with a total of 270 observations as volunteers were asked to choose at least two search queries of different categories from Google Trends 2017 database as shown in table 7.1. These datasets of 110 volunteers are then replicated by the help of a Java replication program to generate three files of 989 MB, 1.5 GB, and 2.24 GB to test for Hadoop2 based analytics capabilities. The replicated data is made from the real datasets of human volunteers used for experimental study as personalized search precision should be evaluated from real search datasets and should be judged by humans. The replicated real-time datasets are

used due to two main reasons (i) professional search engines does not provide actual user browsing history data due to their security policies. (ii) To check capabilities for big data analytics by the implemented tool, a large volume of datasets is required, which is generated using a replication program. However, there is no deviation between real search datasets and synthesized/replicated datasets used within current research work except for the volume. The feedback.csv consists of information regarding user_id, profession_id, query_id, query_title, result_id, result_title, result_link, and various binomial rating columns with two possible values as discussed below (Malhotra & Rishi, 2018b):

- *Feedback* represents the relevancy response by the user for the previous web link in his browsing history and can take either of two values, i.e., Yes or No
- *Loading* represents the web page loading experience of the user and can take either of two values, i.e., Good or Bad
- *Response* represents the response time experience of the user and can take either of two values, i.e., Good or Bad
- *Security* represents the security protocol feature provided by the candidate web page and can take either of two values, i.e., Yes or No
- *Personalized* represents the usage of the feature, i.e., personalized expansion of the query by the user as available on the tool interface and can take either of two values, i.e., Yes or No

7.3. MACHINE LEARNING MODELS

Machine learning capabilities are required to predict the best matching user ID with similar web search tastes to the current user of the IMSS system. The matching user ID and his or her search history assist to correctly expand or suggest personalized search queries to the current user by suggesting the queries sharing the keywords as searched by the matching user. The present research work employs two types of machine learning

models (i) Logistic regression (ii) Collaborative filtering to impart machine learning capabilities. The machine learning performance of the recommended models is evaluated by various relevant metrics, for instance, the logistic regression model is evaluated using accuracy, specificity, sensitivity, precision, and recall metrics while collaborative filtering model is evaluated using MAE, RMSE metrics. These statistics, in turn, assists in the accurate evaluation of the web search personalization capabilities of the IMSS system. These two machine learning models are discussed in detail as follows.

7.3.1. Machine Learning Using Logistic Regression

To predict the preference of a user for a specific web page, we have developed here a machine learning model based on logistic regression. Here, the response variable to be predicted is *feedback* representing the relevancy of listed web link in the output of the metasearch tool or a search engine for the user. The data is required to be in the .csv format as necessitated by R statistical tool (Malhotra & Rishi, 2018b). The Intelligent Meta Search System (IMSS) provide search recommendations to the new user based on feedback given by the previous users, predicted as best match users by the machine learning model as shown in Fig. 7.3.

As the response variable, i.e., feedback is binomial, so family = binomial (link = "logit") will be used while creating the personalized search model within current research work. This syntax can be easily understood in mathematical terms as shown in equation (1) and (2):

$\ln(\text{odds ratio}) = \ln [P / (1 - P)] \quad \dots\dots\dots(1)$ <p>Where, P = Probability of success or probability of response, i.e., Feedback = Yes</p>

$\text{Logit (P)} = \ln [P(\text{Feedback} = \text{Yes}) / P(\text{Feedback} = \text{No})] = C_0 + C_1 \times \text{Loading} + C_2 \times \text{Response} + C_3 \times \text{Security} + C_4 \times \text{Personalized} \quad \dots\dots\dots(2)$

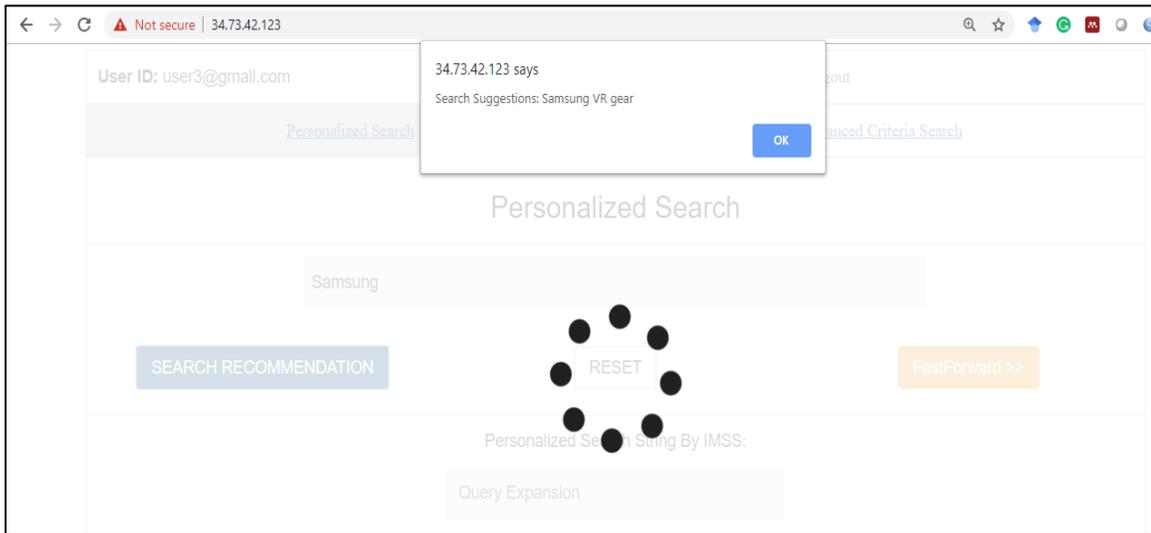


Fig. 7.3 Search suggestions by IMSS tool

To improve the prediction accuracy of the response variable, i.e., feedback by predicting the natural logarithm of the odds ratio, the probability of accurately predicting the feedback response of the web user in the deployed model may be calculated as shown in mathematical equation (3):

$$\text{Probability of true feedback} = \text{predicted odds ratio} / (1 + \text{predicted odds ratio}) \quad \dots(3)$$

7.3.1.1. Steps for Generating Model

- Reading feedback.txt file to retrieve search data
- Model generation using various search parameters
- Plotting diagnostic curves for the generated model
- Recasting model by removing non-significant search parameter
- Deviance calculation between original and recast model
- Testing for multicollinearity and over-dispersion to determine the accuracy of prediction
- Plotting diagnostic curves for recast model

The detailed description of the various steps is discussed below.

Reading feedback.txt file

In order to generate a logistic regression based machine learning model for predicting web user personalized search preferences and hence feedback for a specific web link, there is a need to consider his or her previous search and corresponding feedback data to determine latest preferences of the user. This feedback data is maintained in a text file within the .csv format. The feedback file can be read using read.csv() function as discussed below:

```
feedback_data <- read.csv ("D://DheerajUOK//machinelearning//feedback.txt", header =  
TRUE, sep = ",")
```

As shown above, read.csv () function accepts three parameters, i.e.,(i) path of the file containing user feedback data about previous searches on the tool, (ii) header information, i.e. whether header or column captions are there in the feedback file and (iii) separator information, which is comma in the case of.csv file. The feedback.txt information is stored within feedback_data, and that can be summarized to show consolidated details on various columns within the feedback file.

Model generation using various search parameters

In order to generate the regression model for feedback prediction of prospective web link, glm () function in R (Malhotra & Rishi, 2018b) with the following syntax is used and is also shown in Fig. 7.4.

```
feedback_model = glm (feedback ~ loading + response + security + personalized, data =  
feedback_data, family = binomial (link = "logit") summary (feedback_model)
```

The summary of the generated model is also shown in table 7.2, and table 7.3 represents information regarding reference value taken for each parameter, estimated contribution, standard error, and Pr, i.e., predictability value calculated using glm() function. Here, the first argument of glm() function is a response variable, i.e., feedback and is required to be predicted concerning the remaining parameters, i.e., loading, response, security and personalized. These variables are so chosen for feedback prediction as they are most important to determine the overall search experience of the user (Malhotra & Rishi, 2018b). If a user, search for a query and the web pages are ranked based on the personalized preferences of the user, and hence there is a high probability of positive feedback for a web page. However, just personalized ranking is not enough, if attributes like rapid page loading, less response time, secure browsing are missing, then there are high chances that user might not be able to give feedback as he might even not able to open the web page for browsing. The binomial regression model is derived from feedback.csv datasets obtained from the volunteers through real-time search.

Moreover, the interface of the IMSS tool is designed in such a way that volunteers can quickly determine various attributes like page loading speed, response time and security without requiring visiting the web page listed in the output for his or her recent search query as statistics are available on the interface of the tool. The estimated contribution, Std. Error and Z values are shown in table 1. If $Pr > 0.05$ for a specific parameter then the particular parameter is not considered as significant for deriving the regression model, for instance, as shown in table 7.2, the response variable is having $Pr = 0.755470$. Hence the response variable is not significant and may be removed by recasting the model.

Further, information regarding null deviance, residual deviance, and Fisher Scoring iterations is also available within the summary of the model. A small value of Residual deviance as compared to Null deviance shows a good model. Moreover, a Fisher scoring of fewer than eight iterations, here it is four iterations also strengthens the fact that

proposed and deployed model is a useful model (Malhotra & Rishi, 2018b) and can efficiently and correctly predict the dependence of user's feedback value over other search parameters such as response time, page loading speed, security, and personalized relevancy.

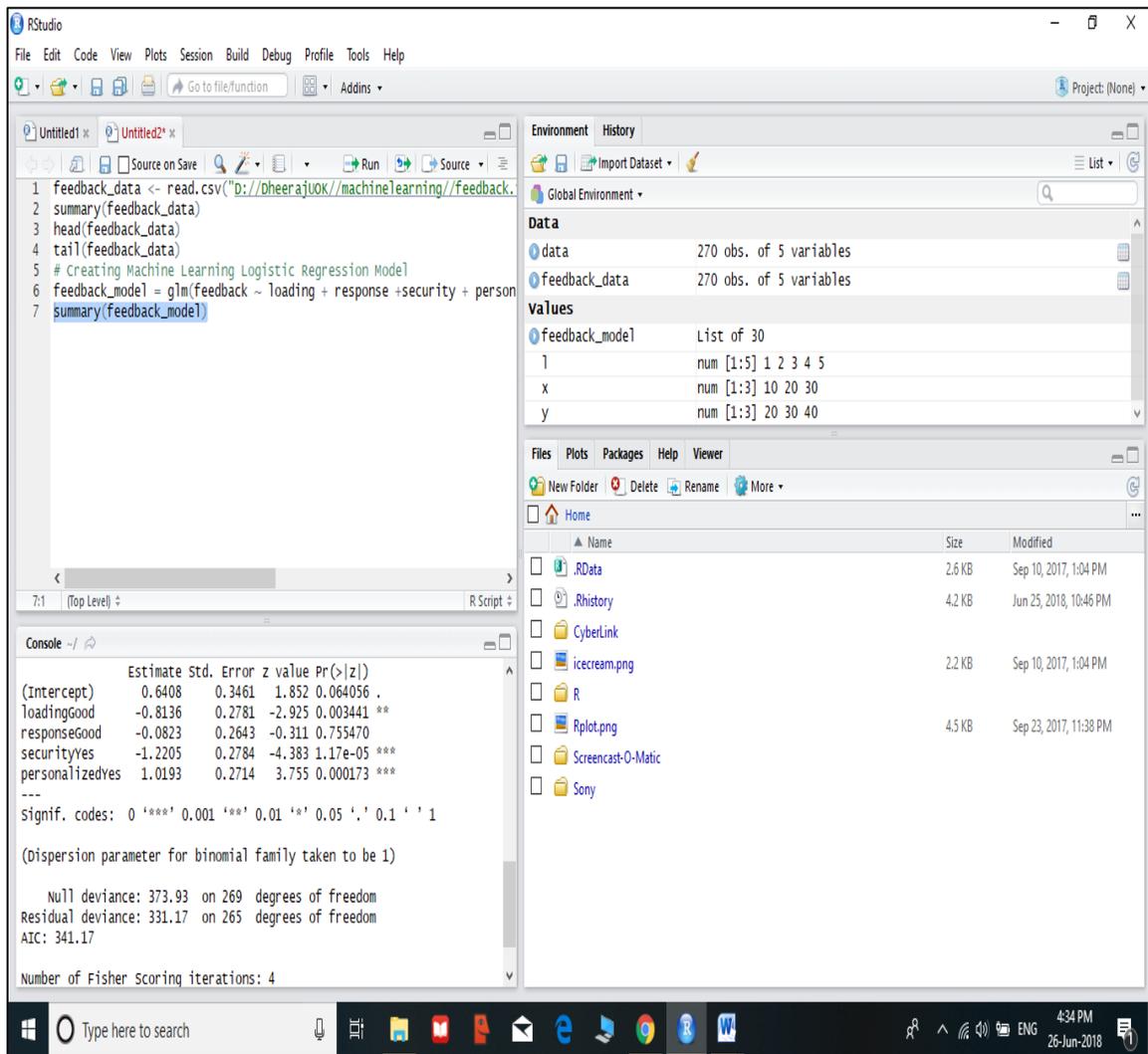


Fig. 7.4 Generalized linear model generation for IMSS tool

Table 7.2 Statistics of various search parameters as calculated by the generalized linear model

Search Parameter	Reference Value	Estimate	Std. Error	Z Value	Predictability Value
Page Loading	Good	-0.8136	0.2781	-2.925	0.003441
Response Time	Good	-0.0823	0.2643	-0.311	0.755470
Security	Yes	-1.2205	0.2784	-4.383	1.17e-05
Personalized Query	Yes	1.0193	0.2714	3.755	0.000173

Table 7.3 Model deviance statistics

Null Deviance	373.93
Residual Deviance	331.17
Degrees of Freedom	269 (null deviance) and 265 (residual deviance)
Fisher scoring Iterations	4

Plotting Diagnostic curves for the generated model

There are four diagnostic curves plotted for the generated model, feedback_model as shown from Fig. 7.5 to Fig. 7.8 (Malhotra & Rishi, 2018b). The detailed interpretation of various diagnostic plots is being discussed in this chapter after recasting feedback_model and while plotting diagnostic curves in fig. 7.9 to 7.12 for recast feedback_model2.

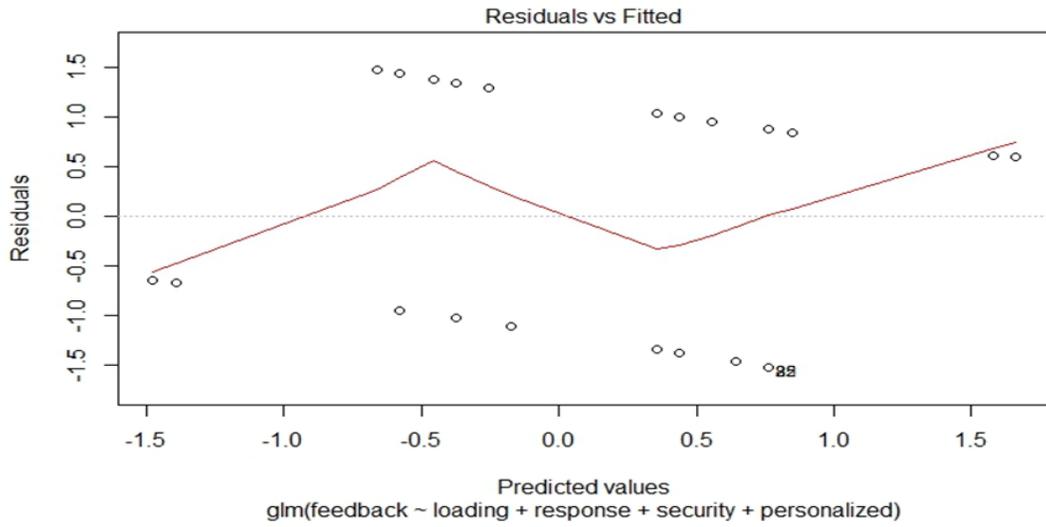


Fig. 7.5 Feedback_Model diagnostic plot- Residuals vs. Fitted

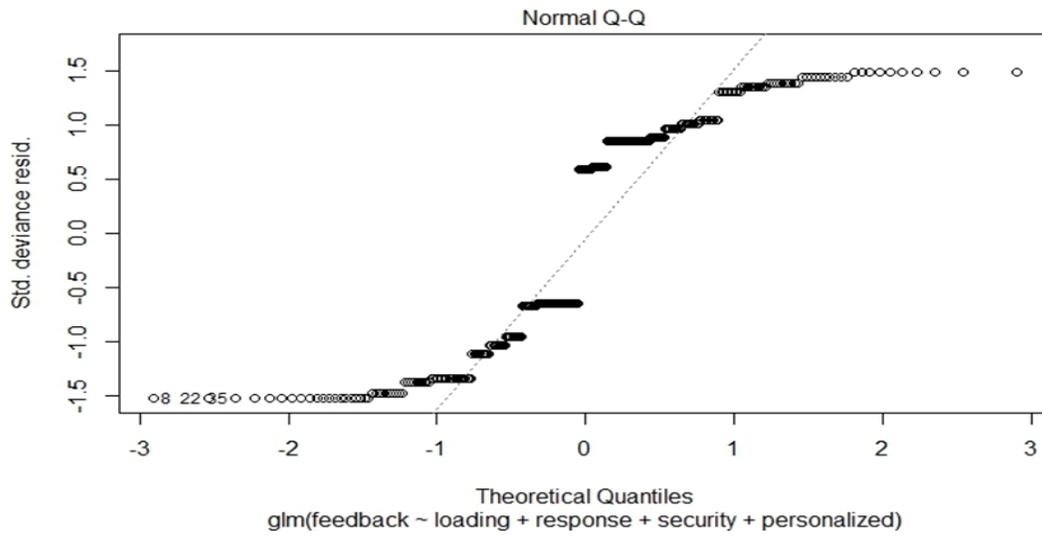


Fig.7.6 Feedback_Model diagnostic plot- Normal Q-Q

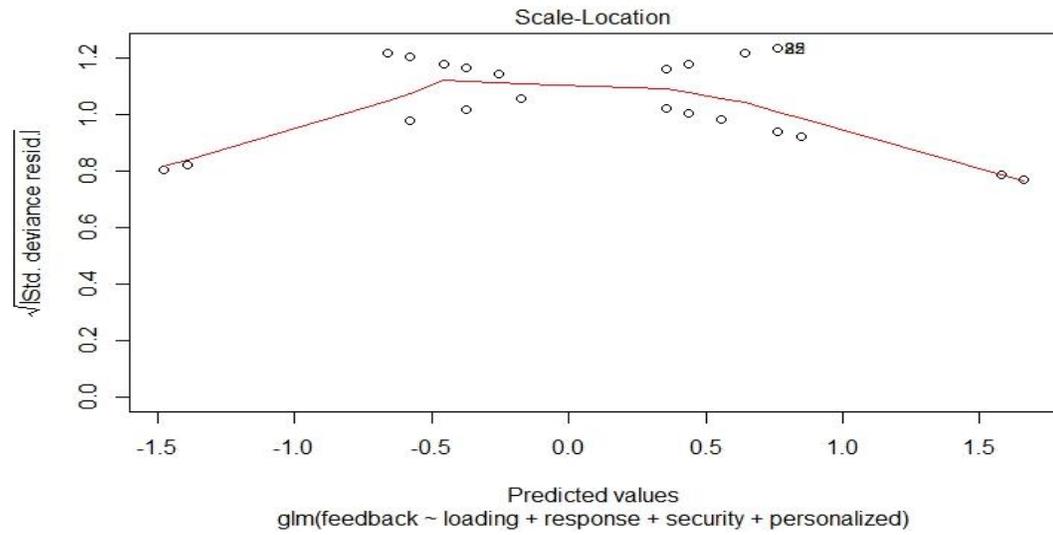


Fig.7.7 Feedback_Model diagnostic plot- Scale-Location

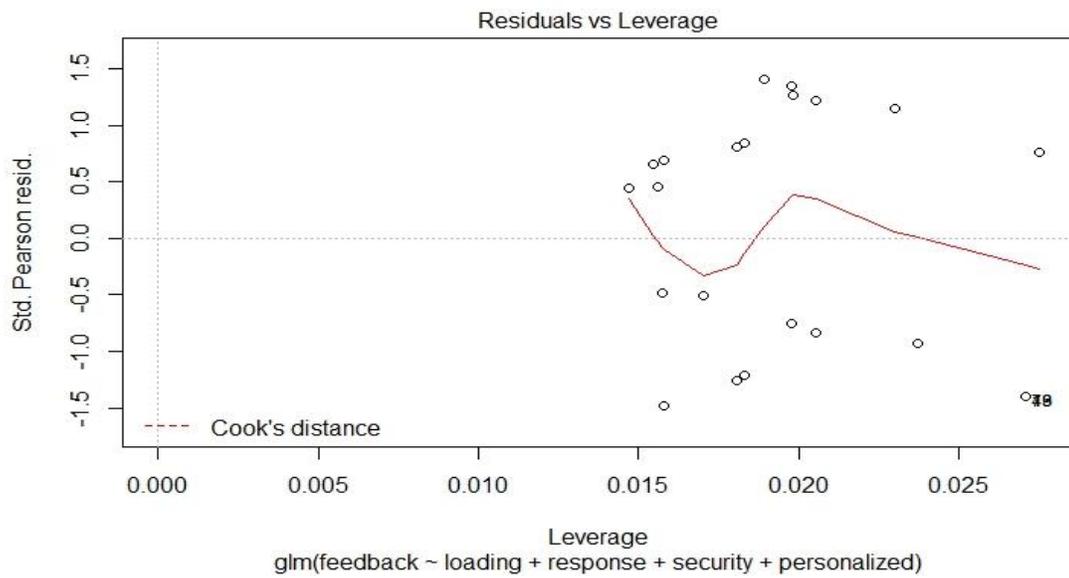


Fig. 7.8 Feedback_Model diagnostic plot- Residuals vs. Leverage

Recasting model by removing non-significant search parameter

As evident from the summary and diagnostic curves, the generated model may be further improved by recasting. The recast model may be generated by eliminating search parameters having P value more than 0.05, i.e., after removal of the response parameter. The command to generate new feedback_model2 without response parameter is as follows (Malhotra & Rishi, 2018b):

```
feedback_model2 = glm (feedback ~ loading + security + personalized, data =  
feedback_data, family = binomial (link = "logit")  
  
summary (feedback_model2)
```

After recasting the model, statistics generated for various parameters are shown in table 7.4 and table 7.5:

Table 7.4 Statistics of various search parameters for recast generalized linear model

Search Parameter	Reference Value	Estimate	Std. Error	Z Value	Predictability Value
Page Loading	Good	-0.08146	0.2779	-2.931	0.003383
Security	Yes	-1.2214	0.2783	-4.389	1.14e-05
Personalized Query	Yes	1.0205	0.2712	3.762	0.000168

Table 7.5 Model deviance statistics for recast model

Null Deviance	373.93
Residual Deviance	331.26
Degrees of Freedom	269 (null deviance) and 266 (residual deviance)
Fisher scoring Iterations	4

Deviance calculation between original and recast model

The deviance between original and recast model can be calculated using anova () function as shown below:

```
anova (feedback_model,feedback_model2, "PChiSq")
```

The first two arguments of anova () will be two generalized models to be compared using PChisq test. The difference between the degrees of freedom for both the models is one while calculating residual deviance. The deviance difference calculated is -0.096992. The small deviance difference between two models represents a low impact of the response parameter on the generalized linear model.

Testing for multicollinearity and overdispersion

In order to check whether the generated model suffers from multicollinearity or overdispersion, there is need to install the DAAG library in R studio. However, to check multicollinearity in the recast model, the vif() function is used with feedback_model as an argument. The statistics for various search parameters obtained using vif is given in table 7.6.

Table 7.6 Vif value for various search parameters

Search Parameter	Vif value
Loading	1.0986
Security	1.0924
Personalized	1.0063

The vif stands for Variance Inflation Factor. As shown above, the vif value for various search parameters is less than 5. Hence, the model is not suffering from multicollinearity.

The command to calculate and test for over-dispersion for recast personalized search model is

```
overdisp_indicator <- feedback_model2$residuals / feedback_model2$df.residual
```

As calculated in R, over-dispersion indicator value is less than 0.5, hence implemented regression model in the present research work is not suffering from over-dispersion. So the generated model can accurately predict the user's personalized preferences while searching the web.

Plotting Diagnostic Curves for recast feedback_model2

Diagnostic plots after recasting feedback_model to feedback_model2 to show and compare residuals in four different ways are plotted as shown in Fig. 7.9 to Fig. 7.12. The four diagnostic curves are as follows:

- (i) Residuals vs. Fitted values plot
- (ii) Normal QQ Plot- Standard Residuals vs. Theoretical Quantities
- (iii) Scale –Location plot-Standard Residuals vs. Fitted Values
- (iv) Standard Residuals vs. Leverage Plot

The first plot, i.e., Residuals vs. Fitted values shows a non-linear relationship between response and predictor variables. The points lying at the fit line, i.e., dotted line at $y=0$ represent zero residuals while the points lying above the fit line represent positive residuals and below the fit line represent negative residuals. The smooth red non-linear curve represents an excellent fitted model for both `feedback_model` and `feedback_model2`.

The second plot, i.e., QQ plot shows whether residuals follow a linear normal distribution or not. The points are shown as firmly placed near the dotted line in both cases. Hence both `feedback_model` and `feedback_model2` passes normal distribution test.

The third plot, i.e., Scale Location plot is also sometimes referred to as Spread Location plot as it represents a pattern of spreading points across the range of predicted values. The ideal scale- location curve is horizontal and represents Homoscedasticity, i.e., a uniform variation of points across the expected range. However, in the present case, the curve for intermediate points is Homoscedastic and for initial and final points is Heteroscedastic in nature. This red curve represents that the deployed machine learning model will work well for the standard number of search predictors and not with a very small or large number of predictor variables and the same is true for data observations of generated models, i.e., `feedback_model` and `feedback_model2`.

The fourth plot, i.e., Residuals vs. Leverage assists in finding those observations that can potentially determine a regression line. There are a majority of observations that may be included or excluded without affecting the result of the analysis. However, a few observations can hugely impact the regression line and can change the outcome of the analysis. Whenever observations have a high value of Cook's distance scores, they can easily influence the result of the analysis. The points shown near the $y=0$ line represent all those feedback data observations having the high value of Cook's distance scores and hence cannot be excluded from the data used for regression analysis.

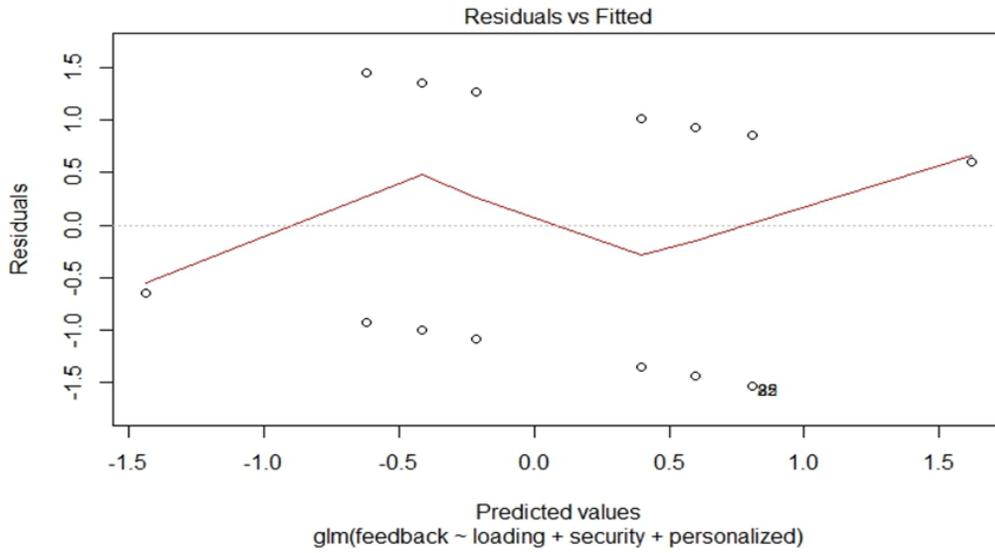


Fig 7.9 Feedback_Model2 diagnostic plot- Residuals vs. Fitted

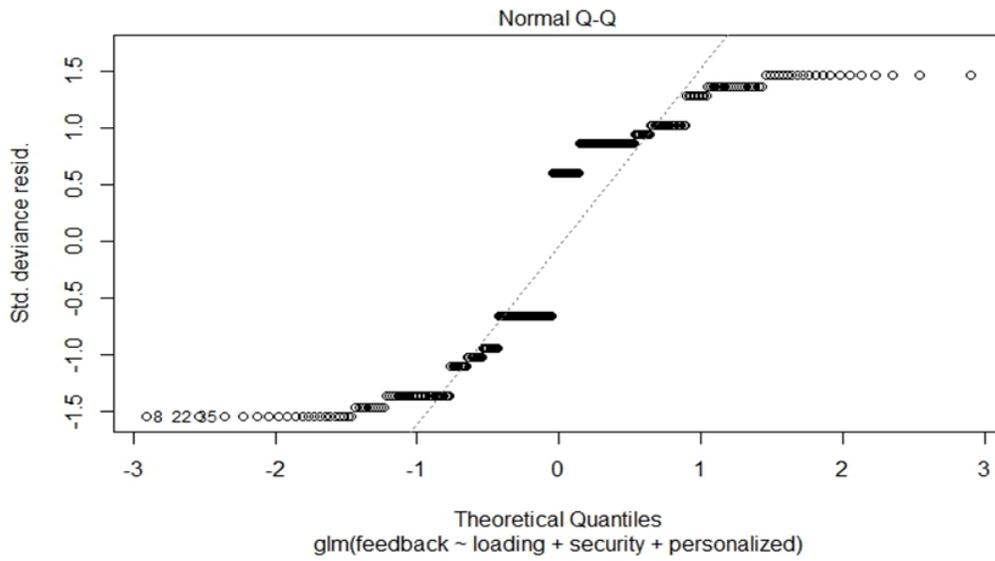


Fig 7.10 Feedback_Model2 diagnostic plot- Normal Q-Q

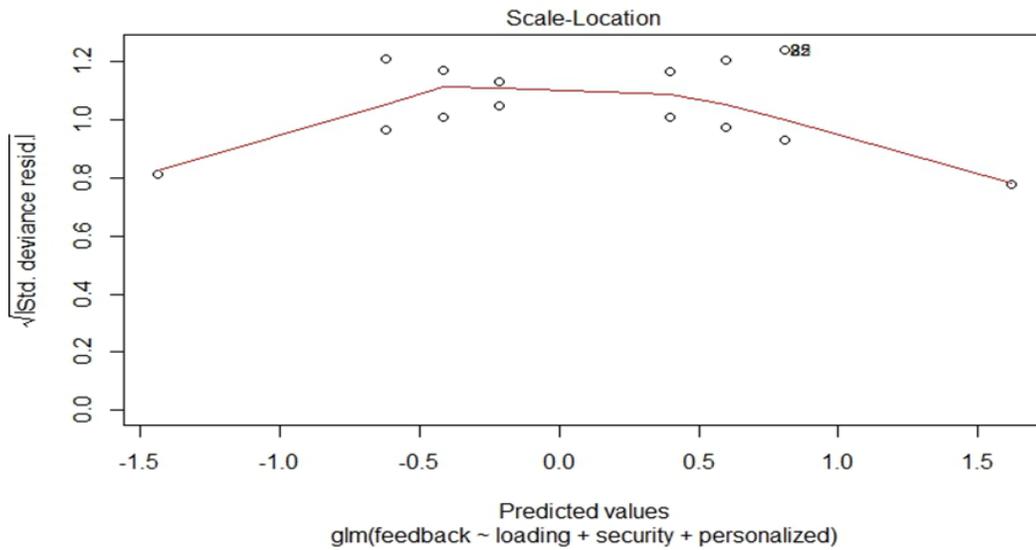


Fig 7.11 Feedback_Model2 diagnostic plot- Scale-Location

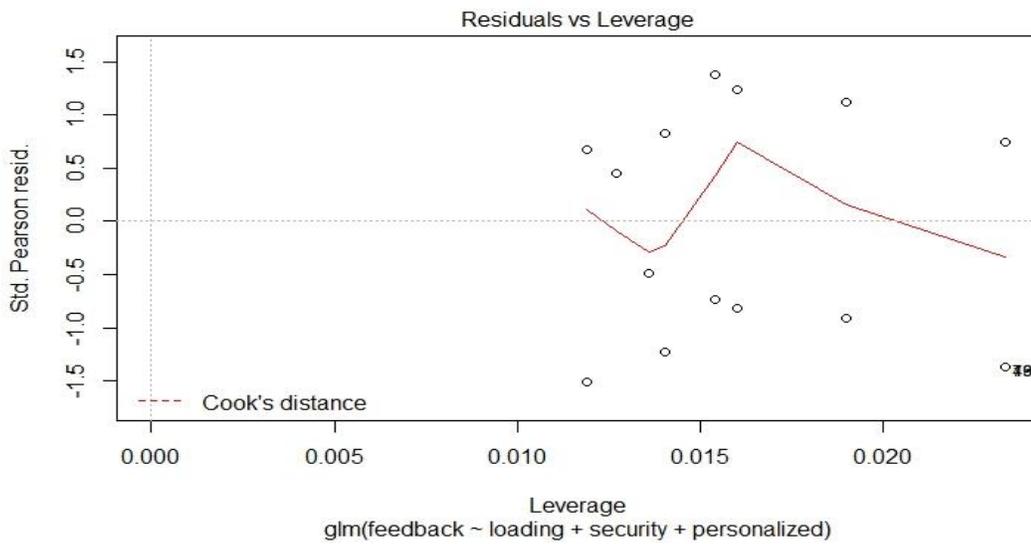


Fig 7.12 Feedback_Model2 diagnostic plot- Residuals vs. Leverage

7.3.1.2. Training and Testing of Model

In order to evaluate the effectiveness of the deployed machine learning model, data sets within the feedback.txt file are subdivided into training and testing data using various commands in R studio as discussed below:

```
feedback_data_partition <- createDataPartition ( feedback_data$feedback, p=0.80, list = false )  
feedback_training_data <- feedback_data[ feedback_data_partition, ]  
feedback_testing_data <- feedback_data[ -feedback_data_partition, ]
```

Here user-specific previous search feedback data initially provided within feedback.csv is divided into two parts, i.e., feedback_training_data consisting of 80 % (0.80) observations, while feedback_testing_data consists of rest 20% observations. The randomly generated subsets of the data may also be used as training dataset and testing dataset for accurate evaluation of the generalized linear model. After creating the data partition, the summarized details can be verified to check the random distribution of records within training and testing data supported by the fact that the serial number of various observations included within training data is not same as that of testing data. Moreover the observations used within testing data are not sequentially picked from total observations; instead, randomly chosen observations are included. Furthermore, summarized details also verify the data partition with the 80-20 ratio, i.e., out of initial 270 observations taken from 110 volunteers, 216 observations are contained within the training data, and the remaining 54 observations are included within testing data. The similar procedure was used for testing and training of the model using replicated big data. The procedure and reason to replicate data are discussed in detail within section 7.2.2.

Training the Model

In the next step, a generalized linear model will be generated using `feedback_training_data` via `glm ()` function in a similar way as described earlier while casting and recasting `feedback_model` and `feedback_model2`. Here, null deviance of the generated model with training data is 299.717 with $216-1=215$ degrees of freedom, while residual deviance is 258.152 with $216-1-4=211$ degrees of freedom as the model calculates residual deviance by subtracting the number of search parameters. Moreover, Fisher scoring iterations of the training data model are 7. As evident from fisher scoring and deviation statistics, model is likely to predict accurately the feedback of prospective web link by the specific web user to successfully implement a personalized search system. Here the training phase is over. However, further analysis is required to verify the predictions of the generated model by using the remaining 20% of the test data.

Testing the Model using Confusion Matrix

The procedure used for testing the model may be step by step summarized as follows:

- Use the `feedback_model` to predict the response variable for all observations within the test data.
- The predicted response variable was compared with the actual values of the response variables stored within the test data.
- After comparison, a confusion matrix was generated to determine False Negatives (FN), True Positives (TP), False Positives (FP) and True Negatives (TN). The objective of a confusion matrix is to demonstrate how many times the response variable, i.e., feedback is correctly predicted as Yes or No. Here False Negatives stand for observations in the test data that were predicted as negative (0) but were positives (1). The True Positives stand for observations in the test data that were predicted as positive (1) and were positive (1). The False Positives stand for observations within test data that were predicted as positive (1) but were negative (0) while True Negatives stand for observations within test data that were predicted as negative (0) and were negative (0).

The accurate model will generate more True Positives and more True Negatives and negligible False Positives and False Negatives to verify its effectiveness in predicting accuracy. The confusion matrix for testing data, i.e., `feedback_testing_data` is shown in table 7.7. The confusion matrix represents the actual values of the feedback variable concerning the predicted value of the feedback variable by the machine learning model. As shown, the number of Correct Yes, i.e., True Positives and Correct No, i.e., True Negatives are much higher than the Incorrect Yes, i.e., False Positives and Incorrect No, i.e., False Negatives. Hence the model generated is a useful and accurate model. There was a total of 49 observations (TN-20 + TP-29) being correctly predicted, and only five observations were wrongly predicted (FN-2 + FP-3) out of total 54 observations in test data, i.e., `feedback_testing_data`.

Table 7.7 Confusion matrix

	ACTUAL YES	ACTUAL NO
PREDICTED YES	29 (TP)	3 (FP)
PREDICTED NO	2 (FN)	20 (TN)

7.3.2. Result Analysis

In order to analyze the effectiveness of logistic regression based machine learning model, various evaluation metrics are calculated, and Receiver Operating Characteristic (ROC) curves are plotted as discussed in subsections 7.3.3.1 and 7.3.3.2.

7.3.2.1. Evaluation Metrics

In order to determine the prediction accuracy of the generated model, the following metrics were evaluated:

- Accuracy
- Specificity

- Sensitivity
- Precision
- Recall

The mathematical expressions or formulas used to calculate all of the above evaluation metrics are shown here in equation (4), (5), (6) and (7)

$$\text{Accuracy} = (\text{TP} + \text{TN}) / \text{Total Observations} \dots\dots (4)$$

$$\text{Specificity} = \text{True Negative Rate} = \text{TN} / (\text{FP} + \text{TN}) \dots\dots (5)$$

$$\text{Sensitivity} = \text{Recall} = \text{True Positive Rate} = \text{TP} / (\text{FN} + \text{TP}) = \text{TP} / (\text{All positives}) \dots (6)$$

$$\text{Precision} = \text{TP} / (\text{FP} + \text{TP}) \dots\dots (7)$$

These metrics can be evaluated for confusion matrix shown in table 7.7:

Here, TP = 29, TN =20, FN =2 and FP =3

Therefore, using equation (4), (5), (6) and (7)

$$\text{Accuracy} = (29 + 20) / 54 = 0.9074 = 90.74\%$$

$$\text{Specificity} = 20 / (3+20) = 0.8695 = 86.95\%$$

$$\text{Sensitivity} = \text{Recall} = \text{TPR} = 29 / (2+ 29) = 0.9354 = 93.54\%$$

$$\text{Precision} = 29 / (3 +29) = 0.9062 = 90.62\%$$

As shown above, high values of accuracy, specificity, sensitivity, precision, and recall metrics proved the prediction accuracy of the pioneered Advanced Cluster Vector Page Ranking (ACVPR) algorithm and Intelligent Meta Search System (IMSS) tool.

7.3.2.2. ROC Curves

To further analyze the effectiveness of deployed regression model, performance () function of ROCR package may be used to obtain True Positive Rate (TPR) and False Positive Rate (FPR), i.e, $TPR = TP / (FN + TP) = TP / (\text{All Positives})$, $FPR = FP / (FP + TN) = FP / (\text{All Negatives})$ and plot TPR against FPR to obtain Receiver Operating Characteristic, i.e. ROC curve. The TPR statistic represents many positives which were correctly predicted by the IMSS tool, i.e., positive feedback for a prospective web page suggested by the machine learning model to adapt to personalized requirements of the specific web user and was accepted as relevant by the user. This prediction is further supported by the web user when in his or her feedback; web page was marked as relevant. On the other side, FPR represents those positive predictions that were not finally marked as relevant by the user. The ROC curve in Fig. 7.13 represents that TPR is improving rapidly compared to FPR. This observation proves the capabilities of the implemented IMSS tool regarding correct prediction of relevant web links for the user.

The specificity represents a true negative rate which means all those negative predictions by the machine learning model regarding a specific web link returned by the background search engine that was also marked as non-relevant by the user. On the other hand, sensitivity represents the true positive rate, i.e., high values of both sensitivity and specificity metrics support the accurate prediction by proposed and implemented ACVPR algorithm and IMSS tool. The ROC curve in Fig. 7.14 indicates that sensitivity falls with an increase in specificity. This curve is again supporting the accurate web page prediction by the deployed model and tool. Moreover, both sensitivity and specificity cannot simultaneously be dominant because of the binomial nature of the response variable, i.e., feedback, as, either a suggested web link by the deployed model is marked as relevant/positive (1) or non-relevant/ negative (0) by the web user in his or her feedback.

In web search domain, precision metric represent relevance of returned web links in the output while recall represents the comprehensiveness of the search result. For instance, a web search tool returns ten relevant web links out of a total of 50 links in its output. However, it misses to return any of the remaining 60 more relevant web links then the precision of the search tool is $10/50 = 1/5$. However, the recall of the search tool is $10 / (10+60) = 1/7$. Fig. 7.15 represents a plot between precision and recall for the IMSS tool. Here, precision is initially constant; however at larger values of recall, precision starts falling. The plotted relationship represents that when the total number of relevant links returned by background search engines is less; IMSS tool can quickly identify the relevant links and can correctly predict their ranking. This low value of precision can be sorted by identifying and incorporating more personalized search parameters to maintain a good contrast between various web links and hence to quickly decide the ranking order among relevant web links returned by the background search engines to the IMSS tool.

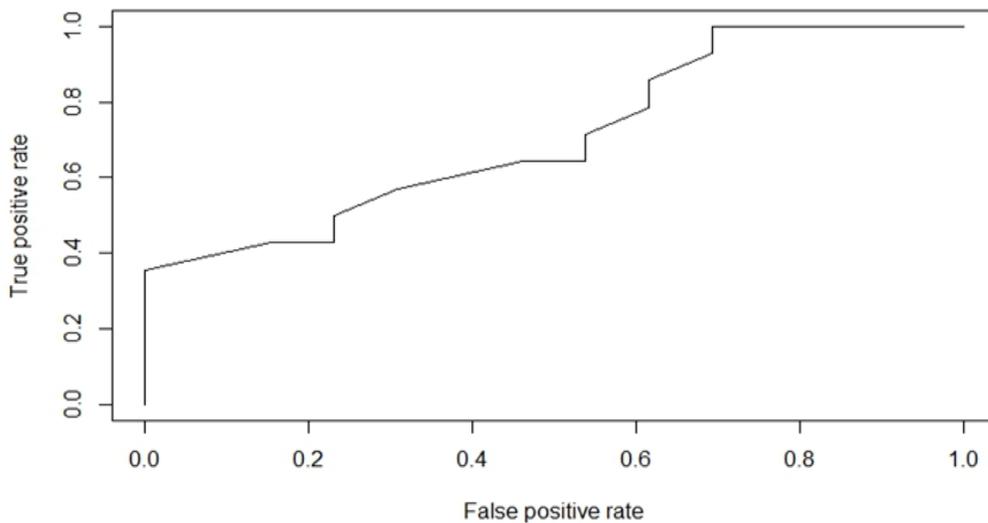


Fig 7.13 TPR vs. FPR for the deployed regression model

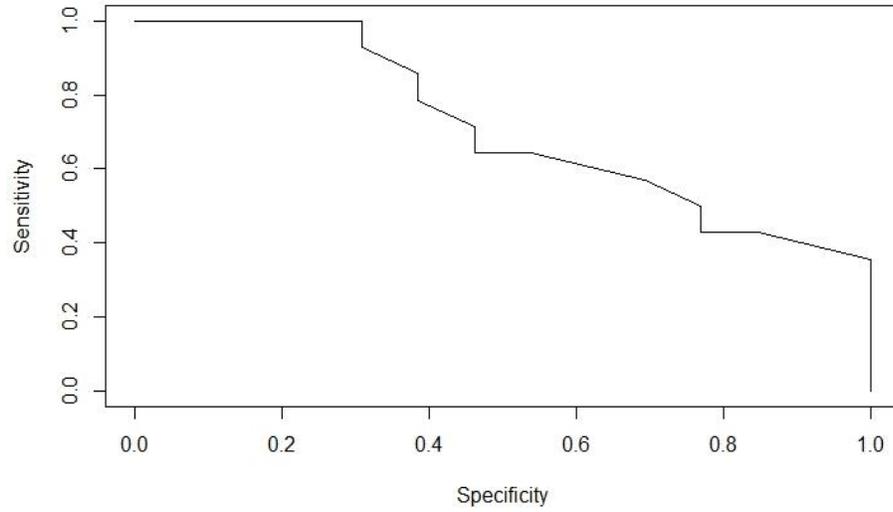


Fig 7.14 Sensitivity vs. specificity for the deployed regression model

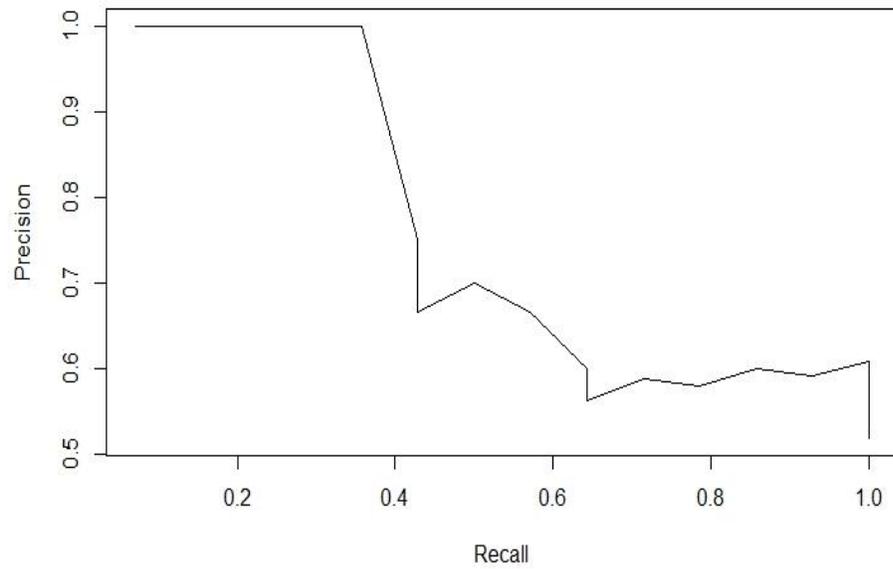


Fig 7.15 Precision vs. recall for the deployed regression model

7.3.3. Collaborative Filtering Model

To further analyze and improve the web search personalization and learning capabilities of the pioneered ACVPR algorithm and IMSS system. An alternative machine learning model is developed to predict the best match existing User ID of the system. The collaborative filtering model is responsible to suitably expand or disambiguate the search queries by referring search history of the similar user and suggesting queries with sharing keywords made by the best match user by assuming that the users who exhibit proximity by behavior analysis will like to prefer similar web link ranking order. The collaborative filtering based machine learning model is also used in the current research work to predict the user response about the answer of multiple questions for behavior analysis during the registration phase of the deployed IMSS tool. The implementation of the collaborative filtering model is based on user similarity calculation using web triplets and is discussed as follows:

7.3.3.1. User Similarity and Web Triplet

In order to calculate the user similarity matrices and hence to determine the best match user ID, the answers given by the user within the *Add Skills* tab during the signup process of deployed metasearch tool are analyzed. Moreover, the web search history of existing users is required to be examined to know the time spend statistic and nature of queries to determine the latest user interests. The web triplet plays a crucial role in deciding user interests and hence assists in the calculation of user similarity matrix. The web triplet consists of:

- Title
- URL
- Summary

A screenshot in Fig. 7.16 shows two web triplets corresponding to a search query. The title portion is highlighted in blue color, after then, URL in green color and finally summary within black color.

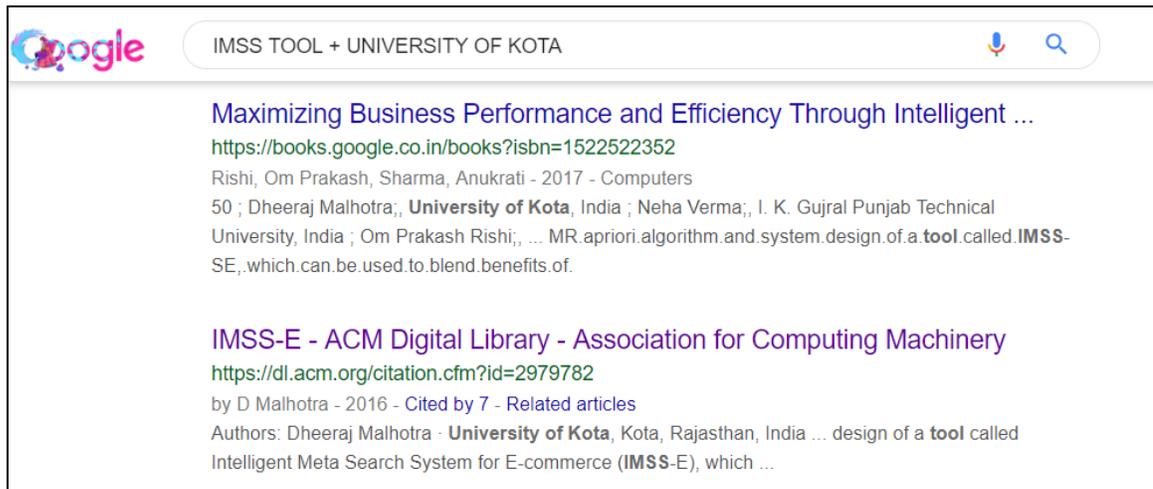


Fig. 7.16 Web triplets for a search query

7.3.3.2 Common Subsequence Identification

A common subsequence is a group of keywords that frequently co-occur within web triplets. A common subsequence represents personalized user interests. The common subsequences are retrieved from the web triplets. The web triplets are finally stored in the form of subsequences. However, only those subsequences having frequency more than the user-specified threshold frequency are saved in the database of the IMSS tool. The best match is determined by calculating two types of match similarities:

- Coexistence Match (CMS)
- Nesting Match (CPS)

If a group of two or more words co-occurs in different web triplets within web search results, then there is strong possibility to have the domain based similarity relationship between such subsequences and is commonly termed as coexistence match (Bibi et al.,

2014). In order to calculate the coexistence similarity between the two subsequences, S_i and S_j , there is a need to check the frequency of co-occurrence within the title or summary or both against the user specified frequency threshold.

In order to calculate the Coexistence Match Similarity (CMS) within the title and summary, the following relationship is considered and is also shown in equation (8):

$$\text{CMS} = \text{SS} / \log x \quad \dots\dots (8)$$

Where SS is Subsequence Similarity in x number of web triplets. The SS is calculated as shown in equation (9):

$$\text{SS} = \log (f (S_i \cup S_j) / f (S_i) \cdot f (S_j)) \quad \dots\dots (9)$$

Where $f (S_i \cup S_j)$ is combined triplet frequency of subsequences S_i and S_j in web triplets

$f (S_i), f (S_j)$ are frequencies of subsequences S_i and S_j respectively in web triplets

The CMS is required to satisfy the user-specified threshold to consider the web triplet in best match calculation.

The nesting match is based on the fact that subsequences S_i and S_j occur in such a way that whenever S_i occur then S_j must happen. However, the occurrence of S_j without S_i is also observed on an individual basis. The nesting match similarity calculation is based on conditional probability based match calculation and hence is also referred here as Conditional Probability Similarity (CPS) and is shown here in equation (10).

$$\text{CPS} = P (S_i / S_j) \quad \dots\dots (10)$$

Where $P(S_i / S_j)$ refers to the conditional probability of S_i given S_j .

The *CPS* like *CMS* is also required to satisfy the user-specified threshold to consider the web triplet in best match calculation.

7.4. EVALUATION METRICS- COLLABORATIVE FILTERING MODEL

In order to evaluate the effectiveness of machine learning model generated through collaborative filtering, the following evaluation metrics are considered and calculated as shown in Fig. 7.17

- Pearson Correlation Coefficient
- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- Euclidean Distance

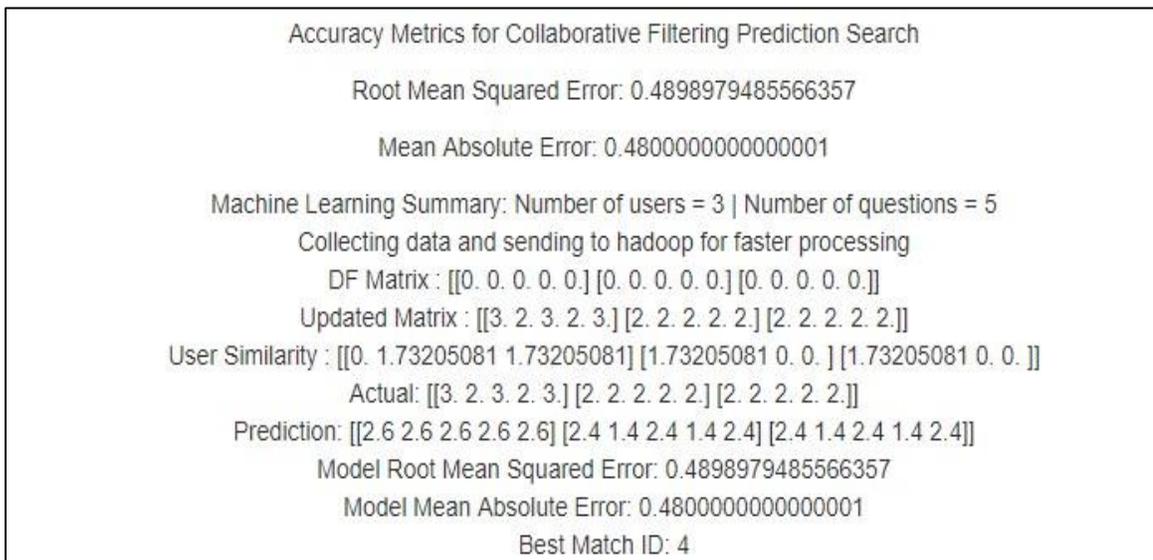


Fig. 7.17 Evaluation metrics for collaborative filtering model

7.4.1 Pearson Correlation Coefficient

In order to identify the best existing match of a user and hence to suitably expand the search queries for the current user by the proposed and implemented IMSS tool, the concept of Pearson correlation coefficient is used in the current research work. The Pearson correlation coefficient tells about the relationship between two variables and is used to describe the linear relationship between two variables. It varies between -1 to +1. The correlation coefficient of +1 represents a robust positive relationship or a directly proportional relationship, i.e., if one variable increased then another variable also increases. The correlation coefficient of -1 depicts a robust negative relationship or inversely proportional relationship, i.e., if one variable increase then another one decreases or vice versa. The correlation coefficient equals to 0 represent no correlation between the two variables.

The Pearson correlation coefficient is calculated using the equation (11).

$$r_{XY} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \dots\dots\dots (11)$$

The formula shown in equation (11) is calculated as the ratio of covariance with the product of the standard deviation of the two variables, i.e., X and Y.

In the present research work, the Pearson correlation coefficient is used to find the correlation matrix. The correlation function of Python uses Pearson correlation coefficient to determine the relationship between actual and predicted values of answers corresponding to various questions asked for behavior analysis during registration/signup step while using IMSS tool for the first time to *Add Skills*. The behavior analysis through the answers given by the user assists in predicting the existing best match user of the IMSS tool for the current user.

The strong correlation between two variables is represented by dark red color and weak correlation between the two variables is represented by the dark blue color on the scale from +1.0 to -1.0 as shown in Fig. 7.18 and Fig. 7.20 for user1 and user2 of the IMSS tool. The correlation here represent the relation between actual answers given by the user and predicted responses by the system. Fig. 7.19 shows the line graph to demonstrate the prediction capabilities of the IMSS tool. The red line shows the predicted value of the answer for user 1 by the tool and green line represent the actual answer given by user1. The proximity between the red and green line represents powerful prediction capabilities of the system.

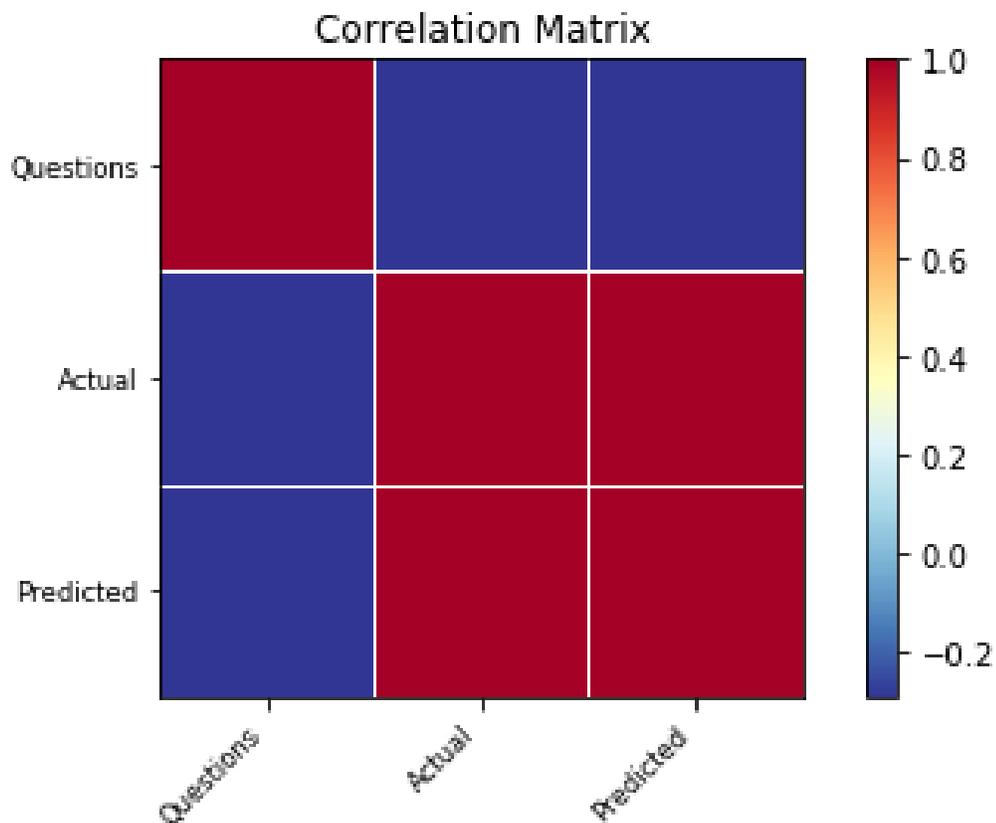


Fig. 7.18 Correlation analysis- User1

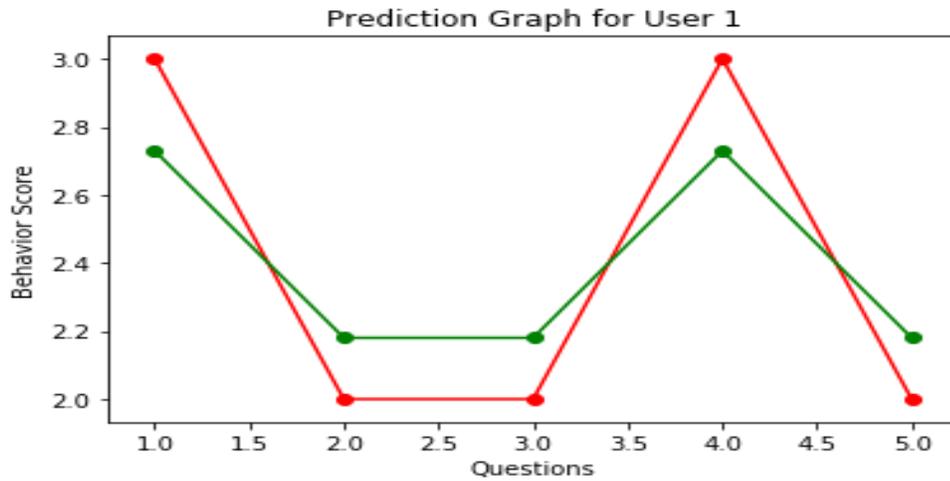


Fig. 7.19 Prediction graph- User1

As shown above, there is a strong correlation between the actual and predicted value of answers for the user1 and hence is shown with red color. However, there is no correlation between questions and values. So, the same is indicated by a dark blue color to represent the least correlation for user1 and user2.

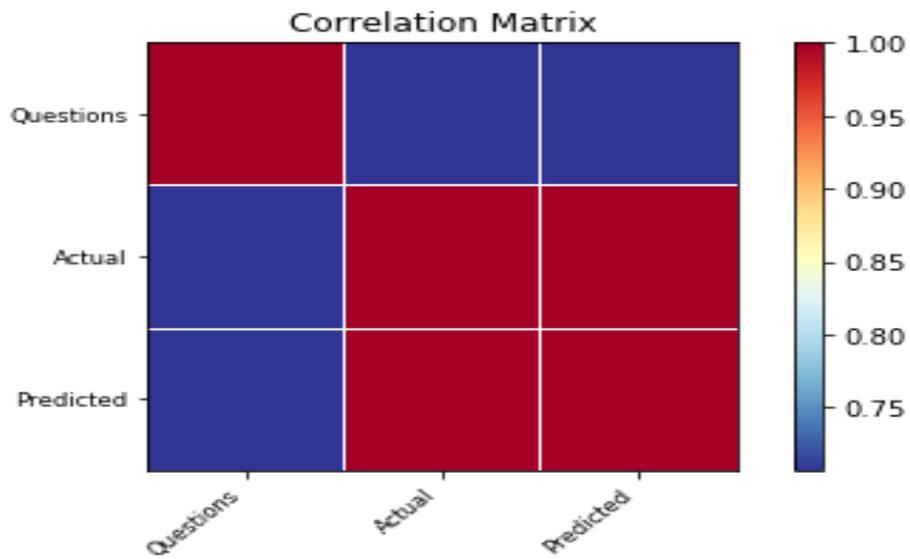


Fig. 7.20 Correlation analysis- User2

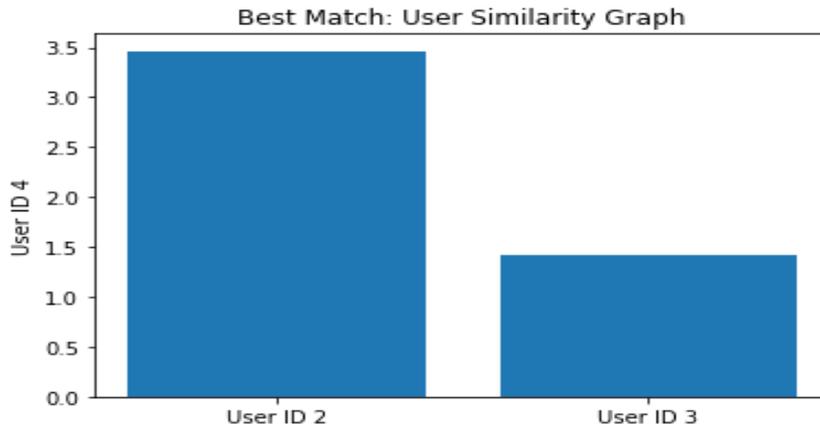


Fig. 7.21 Best match- User similarity graph

The above-shown graph compares the similarity of user2 and user3 concerning user4. The comparison is being made using a correlation statistic between various users. The graph clearly shows the best match for user4 is user2.

7.4.2 Mean Absolute Error (MAE)

Mean Absolute Error refers to the mean of the absolute value of errors in prediction. In the current research work, MAE is calculated while predicting answers for questions used for behavior analysis, i.e., a Predicted value of response – Actual value of the response. The MAE is computed using the equation (12):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}| \dots\dots\dots (12)$$

7.4.3 Root Mean Squared Error (RMSE)

RMSE is the square root of the mean of the squared errors calculated as the difference between predicted and the actual values of answers. It is calculated using equation (13):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2} \dots\dots\dots (13)$$

7.4.4 Comparison between MAE and RMSE

RMSE is more appropriate to highlight large errors as it involves square calculation. However, if the small or large error, that is, the magnitude of the difference between actual and predicted value is immaterial, then MAE is more appropriate. Moreover, it is easy to interpret the MAE.

7.4.5 MAE and RMSE of the Pioneered ACVPR Algorithm

The MAE and RMSE both found to be low while finding search recommendations using implementation of the ACVPR algorithm, that is, IMSS tool. Moreover, it is realized that both MAE and RMSE further decreases with the increase in the number of search queries and an increase in the number of users. Fig. 7.22 shows MAE, RMSE retrieval on the interface of the implemented tool.

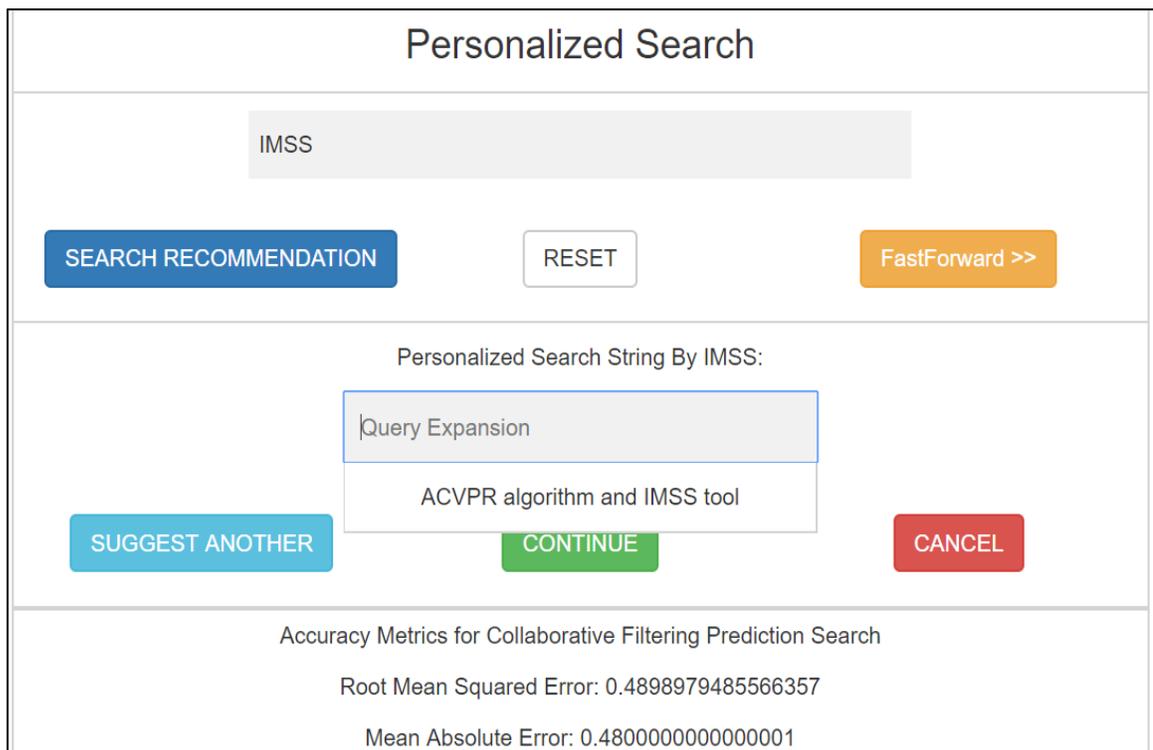


Fig. 7.22 MAE, RMSE calculation for the search query "IMSS"

7.4.6 Euclidean Distance

In the present research work, K means clustering is employed to implement unsupervised machine learning. K means uses Euclidean distance to find the distance between the centroid and various other data points to determine the appropriate cluster of users with similar tastes. The objective is to achieve high intracluster similarity and low value of inter-cluster similarity. The Euclidean Distance is used to calculate user similarity matrix to calculate the magnitude of similarity between various users of the system and hence to ascertain the web links for achieving personalization via deployed IMSS tool. The user similarity matrices are estimated and are shown in the machine learning screenshots in the next section. The Euclidean distance is calculated as shown in equation (14):

$$d = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \dots\dots\dots (14)$$

7.5 MACHINE LEARNING SUMMARY

Here, machine learning capabilities of the pioneered ACVPR algorithm and IMSS tool is evaluated and is listed within screenshots from Fig. 7.23 to Fig. 7.28. The machine learning statistic is calculated with the different number of users in the system to determine the effect of the number of users on the learning and recommendation competences of the recommended approach. The prediction accuracy is dependent on the magnitude of various errors in the system. The lower values of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) with an increase in the number of users of the system as exhibited by the IMSS tool is shown in table 7.8. Hence machine learning and recommendation capabilities of the IMSS tool improve with an increase in the number of users. The reason behind such an observation is that the personalized search system is designed and implemented here in such a way that search history of matching


```

Accuracy Metrics for Collaborative Filtering Prediction Search

Root Mean Squared Error: 0.37046038744236703

Mean Absolute Error: 0.3301208618636423

Machine Learning Summary: Number of users = 5 | Number of questions = 5
Collecting data and sending to hadoop for faster processing
DF Matrix : [[0. 0. 0. 0. 0.] [0. 0. 0. 0. 0.] [0. 0. 0. 0. 0.] [0. 0. 0. 0. 0.]]
Updated Matrix : [[3. 2. 3. 2. 3.] [3. 2. 3. 2. 2.] [3. 2. 3. 2. 2.] [3. 2. 2. 2. 2.] [2. 2. 2. 2. 2.]]
User Similarity : [[0. 1. 1. 1.41421356 1.73205081] [1. 0. 0. 1. 1.41421356] [1. 0. 0. 1. 1.41421356] [1.41421356 1. 1. 0. 1.] [1.73205081 1.41421356 1.41421356 1. 0. ]]
Actual: [[3. 2. 3. 2. 3.] [3. 2. 3. 2. 2.] [3. 2. 3. 2. 2.] [3. 2. 2. 2. 2.] [2. 2. 2. 2. 2.]]
Prediction: [[3.05302198 2.38958665 2.77821807 2.38958665 2.38958665] [2.75147186 2.16568542 2.45857864 2.16568542 2.45857864] [2.75147186 2.16568542 2.45857864
2.16568542 2.45857864] [2.6182654092 2.6182654092 2.14691816] [2.5736695 1.5736695 2.39382884 1.5736695 1.88516265]]
Model Root Mean Squared Error: 0.37046038744236703
Model Mean Absolute Error: 0.3301208618636423
Best Match ID: 2

```

Fig.7.26 Machine learning summary for five users

```

Accuracy Metrics for Collaborative Filtering Prediction Search

Root Mean Squared Error: 0.3565912050589316

Mean Absolute Error: 0.26970562748477145

Machine Learning Summary: Number of users = 4 | Number of questions = 5
Collecting data and sending to hadoop for faster processing
DF Matrix : [[0. 0. 0. 0. 0.] [0. 0. 0. 0. 0.] [0. 0. 0. 0. 0.] [0. 0. 0. 0. 0.]]
Updated Matrix : [[3. 2. 3. 2. 3.] [3. 2. 3. 2. 2.] [3. 2. 3. 2. 2.] [3. 2. 2. 2. 2.]]
User Similarity : [[0. 1. 1. 1.41421356] [1. 0. 0. 1.] [1. 0. 0. 1.] [1.41421356 1. 1. 0. ]]
Actual: [[3. 2. 3. 2. 3.] [3. 2. 3. 2. 2.] [3. 2. 3. 2. 2.] [3. 2. 2. 2. 2.]]
Prediction: [[3.28284271 2.28284271 2.86862915 2.28284271 2.28284271] [3. 2. 2.5 2. 2.5] [3. 2. 2.5 2. 2.5] [2.71715729 1.71715729 2.71715729 1.71715729 2.13137085]]
Model Root Mean Squared Error: 0.3565912050589316
Model Mean Absolute Error: 0.26970562748477145
Best Match ID: 2

```

Fig.7.27 Machine learning summary for four users

```

Accuracy Metrics for Collaborative Filtering Prediction Search

Root Mean Squared Error: 0.39999999999999997

Mean Absolute Error: 0.32000000000000001

Machine Learning Summary: Number of users = 3 | Number of questions = 5
Collecting data and sending to hadoop for faster processing
DF Matrix : [[0. 0. 0. 0. 0.] [0. 0. 0. 0. 0.] [0. 0. 0. 0. 0.]]
Updated Matrix : [[3. 2. 3. 2. 3.] [3. 2. 3. 2. 2.] [3. 2. 3. 2. 2.]]
User Similarity : [[0. 1. 1.] [1. 0. 0.] [1. 0. 0.]]
Actual: [[3. 2. 3. 2. 3.] [3. 2. 3. 2. 2.] [3. 2. 3. 2. 2.]]
Prediction: [[3.2 2.2 3.2 2.2 2.2] [2.8 1.8 2.8 1.8 2.8] [2.8 1.8 2.8 1.8 2.8]]
Model Root Mean Squared Error: 0.39999999999999997
Model Mean Absolute Error: 0.32000000000000001
Best Match ID: 2

```

Fig. 7.28 Machine learning summary for three users

The two essential evaluation metrics, i.e., MAE and RMSE for the machine learning model based on collaborative filtering may be summarized as shown in table 7.8.

Table 7.8 Evaluation metrics summary for deployed collaborative filtering model

S. No.	No. of Users	RMSE	MAE
1	3	0.39999999999999997	0.32000000000000001
2	4	0.3565912050589316	0.26970562748477145
3	5	0.37046038744236703	0.3301208618636423
4	6	0.35239137542973176	0.31776287474083337
5	7	0.341679109565523	0.3038283247828333
6	8	0.33647971094793666	0.296512433673409

7.6 EXPERIMENT DESIGN

7.6.1 Implementation

All experiments performed in the current research work are implemented by writing scripts using Python, PHP, HTML 5 and CSS 3 to deploy Advanced Cluster Vector Page Ranking (ACVPR) algorithm in the form of a metasearch tool, i.e., Intelligent Meta Search System (IMSS). This IMSS tool is a machine learning enabled metasearch engine which uses Python interpreter on the server side for implementation of data analysis and ACVPR algorithm. Results of the analysis are processed using PHP, HTML 5, and CSS 3 and finally displayed on a user browser. The tool uses a MySQL engine for the user and query management. This tool is deployed on the Google cloud engine with Hadoop features enabled for computation of user similarity, rating based recommendations. Here, in the present research work, multi-node and single node cluster are alternately implemented to deploy the IMSS tool. However, a single node cluster is found to be comparatively more efficient in terms of resource consumption.

Here, Hadoop second generation analytics using Google cloud platform for deployment of IMSS tool is preferred as it offers dynamic and elastic scaling because virtual machine instances may be easily added or deleted depending upon the load. Moreover, Google provides efficient resource management regarding virtual machine instances and network bandwidth. The chosen platform offers high reliability and lesser operational issues by using *Rack Placement* strategy. The detailed discussion about the implementation and instance set up on the Google cloud platform is discussed in detail within Chapter 4.

7.6.2. User Survey

Human volunteers verified the effectiveness and efficiency of the implemented IMSS tool and ACVPR algorithm over traditional search engines. There are a total of 110 human volunteers employed for precision comparison of IMSS tool with three popular search engines, Google, Bing, Qwant and a popular metasearch engine, Dogpile. Out of

110 volunteers, sixty are females and rest fifty is male within the age group of 21 years to 49 years with a minimum of five years of experience to carry out web search and browsing. They were asked to access the IMSS application on Google cloud platform using the external IP address of the tool. After completing the signup process, volunteers were asked to repeat at least ten times following steps on IMSS tool to allow the tool to learn their individual personalized preferences and to build their feedback.csv file with columns of relevancy feedback, page loading speed, response time, security and personalized query expansion. The queries searched by volunteers were chosen out of 9 categories as shown in table 7.1 pulled from Google Trends 2017 in India to assure that only those queries should be considered for the experimental purpose which is popular and is searched by real users. This selection, in turn, is required to assure the accurate personalized search precision comparison between professional search engines like Google, Qwant, and Bing with IMSS tool as search engines exercise personalization on popular search queries. The trending search queries from 9 categories well assisted in determining personalized search precision of various professional search engines under consideration and the precision comparison with IMSS tool. The trending queries were searched on IMSS, Google, Bing, Qwant and Dogpile to rank these search tools by determining various search precision parameters as discussed below.

- Firstly, volunteers were asked to search a partial or ambiguous search query on all the search tools under consideration, for instance, search for an incomplete query like “Bitcoin” rather than “How to buy Bitcoin” or “How to mine Bitcoin” or “What is Bitcoin.”
- Secondly, volunteers were asked to assign weight to various search precision parameters in between 1(worst) to 5(best) to top five web links produced in the output of the considered search engines
- The precision data obtained is then normalized by recursively applying equation (15) on each precision parameter for each of the candidate web page returned by

the background search engine to IMSS tool using following equation, Normalized Value (NV) is calculated as shown in equation (15):

$$NV = \frac{(\text{Maximum value of parameter}) - (\text{Measured value of parameter})}{(\text{Maximum Value} - \text{Minimum Value})} \dots\dots\dots (15)$$

The applicable mathematical equivalent expression is shown in equation (16):

$$NP_{ab} = \frac{(\text{MAX} (PP_{ab}) - PP_{abr})}{(\text{MAX} (PP_{ab}) - \text{MIN} (PP_{ab}))} \dots\dots\dots (16)$$

Where, PP_{ab} = Value of b_{th} precision parameter of a_{th} webpage; NP_{ab} = Normalized value of b_{th} precision parameter of a_{th} webpage; MIN, MAX = Minimum and Maximum value of each of the precision parameter

- The overall precision of a candidate web page is then obtained using the weighted summation of the normalized value of each of the precision parameter as shown in equation (17):

$$N_a = \sum W_b \cdot NP_{ab} \dots\dots\dots (17)$$

Where, N_a = weighted precision of a_{th} webpage; W_b = Weight assigned to b_{th} parameter by volunteer, where $0 \leq W_b \leq 1$

- At last, the overall precision of search engine/tool is determined by calculating the average of all the weighted precisions as gathered by volunteers for a given parameter among response time; page updated content, personalized relevancy at a time. The precision is determined by using equation (18)

$$\text{Precision (ID)} = \text{AVERAGE} (N_a) \dots\dots\dots (18)$$

7.7 RESULT ANALYSIS

The graph shown in Fig. 7.32 is comparing the precision of various search parameters and represent that the human volunteer's judgment proves that the precision of ACVPR algorithm and its implementation, i.e., IMSS tool designed and developed in the current research work is better than Google, Bing, Qwant, and Dogpile. The precision of each of the parameter, i.e., Response Time, Page Updation and Personalized Relevancy of IMSS dominate over professional search engines and metasearch engines. This observation, in turn, proves the effectiveness and efficiency of deployed logistic regression and collaborative filtering based machine learning models to merge the output links from all three professional search engines, i.e., Google, Bing, and Qwant.

The primary reason behind improved precision of ACVPR algorithm and IMSS tool over professional search engines is because Intelligent Meta Search System (IMSS) is a metasearch tool and is utilizing the strength of all of its three background engines, i.e., Google, Qwant and Bing. The IMSS tool is offering the search capabilities of all three of these under a single search platform by merging the top links of each of the individual search engines. Furthermore, IMSS tool is well utilizing the machine learning capabilities in predicting user preferences to satisfy the personalized search needs of the user as evident from evaluation metrics and discussed in section 7.3.2 and 7.4. The scalable analytics support of next-generation HDFS framework is also playing an important role in improved performance and hence better satisfaction of the end user.

The graph shown in Fig. 7.29, Fig. 7.30 and Fig. 7.31 demonstrate the average precision metric comparison between IMSS tool with a popular metasearch engine, i.e., Dogpile as calculated by volunteers for various precision parameters, i.e., Response Time, Page Freshness and Personalized Relevancy respectively. The graphs shown indicate that the initial average precision of IMSS is lesser than Dogpile for page freshness, response time and personalized relevancy. However, soon after a few trials run, the precision of the tool improves in comparison to Dogpile. This improved precision demonstrates the effective

collaborative filtering based machine learning capabilities of IMSS tool. This tool can build customer profile database by monitoring his or her personalized browsing preferences with some trial runs and hence tool will be able to calculate various important relevancy vectors as discussed in section 5.2.3, i.e., SRV, FRV, TRV more accurately. However, precision improvement in Fig. 7.29 for response time is not as significant as in Fig. 7.30 and Fig. 7.31, i.e., for page freshness and personalized relevancy. This difference in precision statistics is because of background implementation of Map-Reduce based second generation HDFS used in the tool. This small precision difference is due to time delay occurring on account of iterative analytics. This suspension can be improved further by use of Spark based HDFS system as in-memory computation models implemented through Spark allow intermediate results to be kept in memory and hence reduces the overhead of iterative analytics as discussed by Malhotra and Rishi (2017). However, tools and infrastructural requirements of Spark based implementation are still under development stage, and hence Map Reduce analytics is preferred in the current research work. The extensive experimental evaluation and graphical demonstration in Figures 7.29, 7.30, 7.31, and 7.32 indicates the improvement in various precision parameters at much faster pace when a personalized search is accomplished with IMSS tool over other professional and popular search engines, i.e., Google, Qwant, Bing and a popular metasearch engine, i.e., Dogpile.

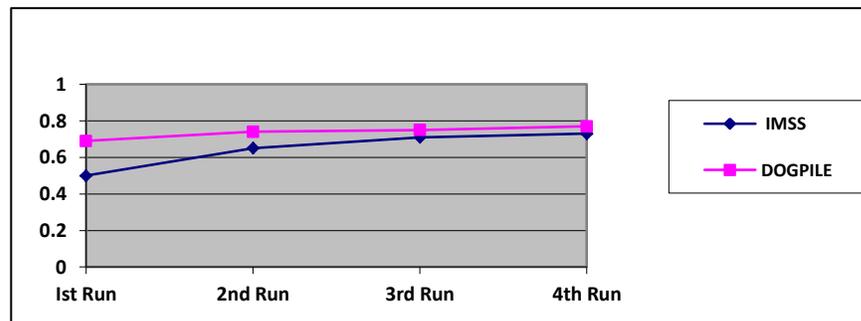


Fig. 7.29 Precision comparison between IMSS and Dogpile- Response Time

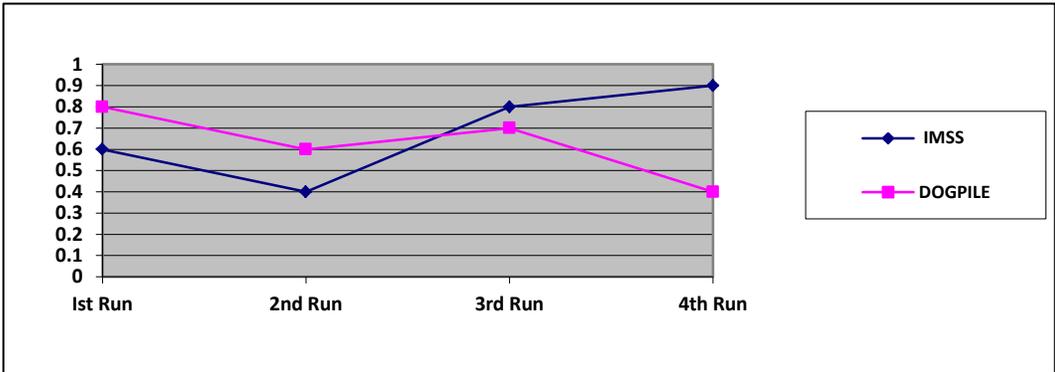


Fig. 7.30 Precision comparison between IMSS and Dogpile – Page Freshness

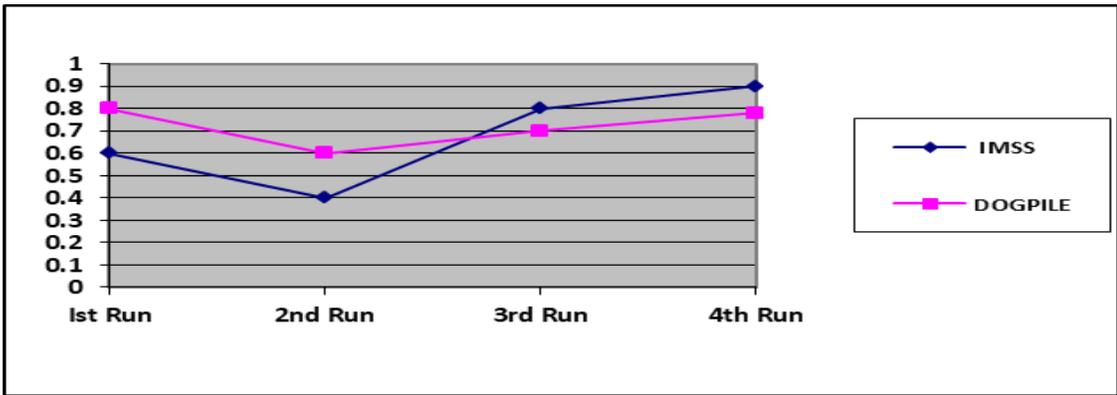


Fig. 7.31 Precision comparison between IMSS and Dogpile- Personalized Relevancy

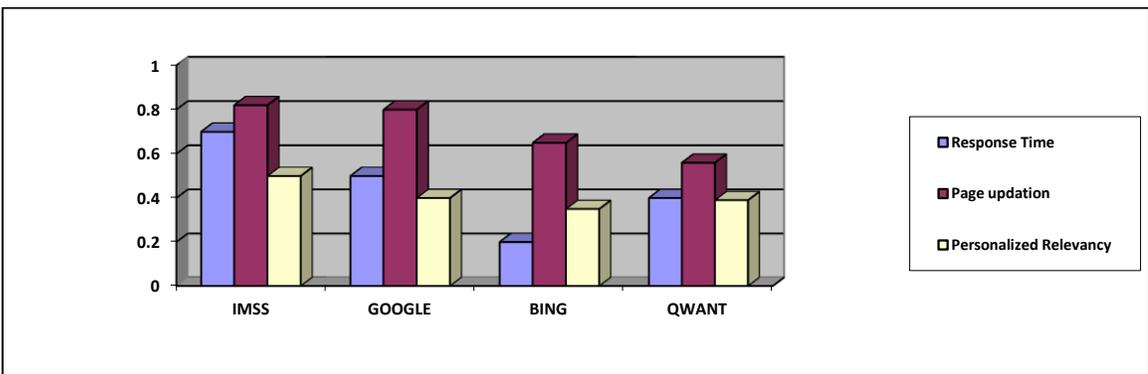


Fig. 7.32 Precision comparison- IMSS with *Google, Bing, and Qwant*

7.7.1 Experimental Verification- Personalized Page Rank Improvement

After, extensive training and testing of IMSS tool through queries in Zeitgeist datasets, volunteers were asked to execute a few random queries out of Google Zeitgeist datasets to experimentally verify query expansion and web page rank improvement for non-popular search queries. The steps performed in the following sequence on the interface of the IMSS tool.

- In the first step, user1 with system generated ID=2 search for a query *apple iPhone* as shown in Fig. 7.33. The results produced are displayed to the user to mention rating to his or her favorite links as shown in Fig. 7.34. User1 had given a high rating to a web link *https://www.apple.com/* ranked seven by Google.

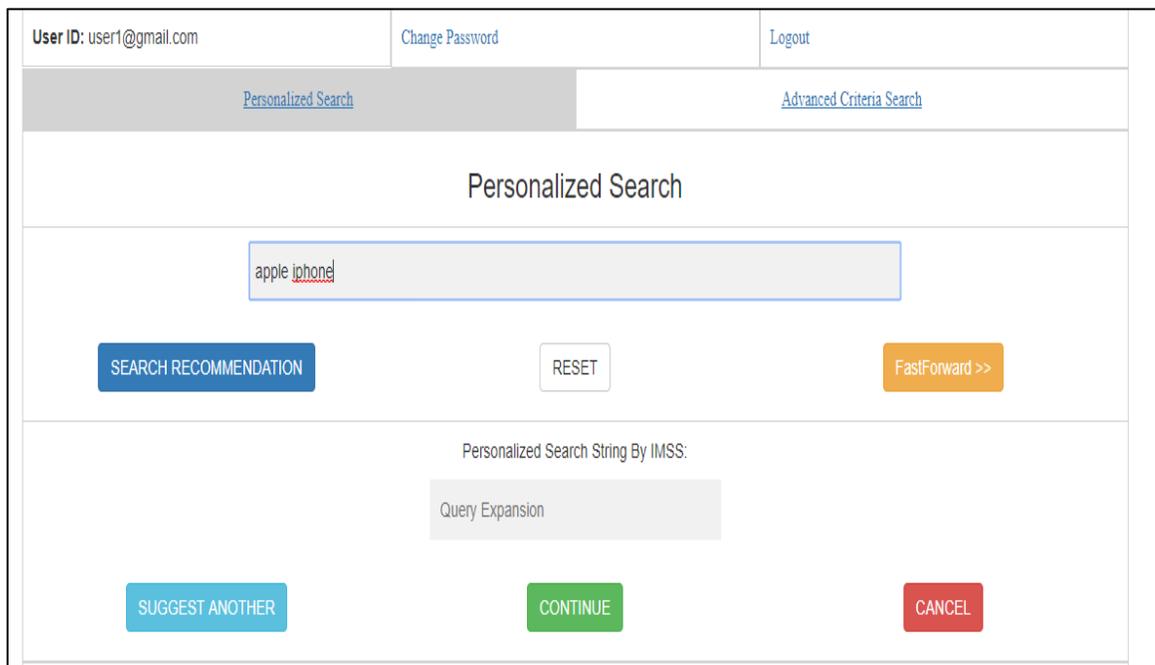


Fig. 7.33. User1 search query- *apple iPhone*

Priority / Rank	Web Link (Search Engine)	Response Time	Security	Loading speed	Rating
1	https://www.forbes.com/sites/gordonkelly/2018/10/21/apples-new-iphone-has-a-serious-problem/ (G)	0.95 ms	Secure	0.97 ms	1 ▼
2	https://support.apple.com/en-us/HT201354 (G)	0.95 ms	Secure	0.97 ms	1 ▼
3	https://www.apple.com/shop/buy-iphone/iphone-xs (G)	0.95 ms	Secure	0.96 ms	1 ▼
4	https://www.apple.com/iphone-xs/ (G)	0.95 ms	Secure	0.96 ms	1 ▼
5	https://www.theverge.com/2018/10/23/18011306/apple-iphone-xr-review-camera-screen-battery-price (G,B)	0.95 ms	Secure	0.95 ms	1 ▼
6	https://www.nytimes.com/2018/10/23/technology/personaltech/apple-iphone-xr-review.html (G,B)	0.95 ms	Secure	0.96 ms	1 ▼
7	https://www.apple.com/ (G)	0.95 ms	Secure	0.96 ms	5 ▼ 

Fig. 7.34 User1- high rating to web link at rank # 7

- In the second step, user2 with system generated ID=3, logged in the system and search for a query *apple fruit*. User2 had also given a high rating to one of the links ranked six as shown in Fig. 7.35.

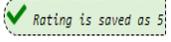
2	https://www.medicalnewstoday.com/articles/267290.php (G)	1.08 ms	Secure	1.08 ms	1 ▼
3	https://www.youtube.com/watch?v=3Y116QN-RLs (G)	1.08 ms	Secure	1.09 ms	1 ▼
4	https://www.nutrition-and-you.com/apple-fruit.html (G)	1.08 ms	Secure	1.08 ms	1 ▼
5	https://www.youtube.com/watch?v=SFZk945I_HE (G)	1.08 ms	Secure	1.1 ms	1 ▼
6	https://www.britannica.com/plant/apple-fruit-and-tree (G)	1.08 ms	Secure	1.09 ms	5 ▼ 
7	https://food.ndtv.com/food-drinks/apple-fruit-benefits-8-incredible-health-benefits-of-apple-that-you-may-not-have-known-1761603 (G)	1.08 ms	Secure	1.08 ms	1 ▼
1	https://www.youtube.com/watch?v=k4lpGQW2Os0 (B)	1.44 ms	Secure	1.50 ms	1 ▼

Fig. 7.35 Results presented to User2

- The last two steps may be considered as the training phase of the IMSS tool, and now there is a need to test the capabilities of the system. In the previous step user3 with system generated ID =4, logged in the system and search for an incomplete query, that is, *apple* followed by clicking search recommendation button. The search recommendations are shown to user3 concerning user1 as user1 with ID =2 is detected as best match ID by the machine learning model of IMSS tool and the query searched by the user1, that is, *Apple iPhone* is shown as a recommendation to complete the query. As user3 continue to explore the recommended query, the rank of the highly rated link, that is [https:// www.apple.com](https://www.apple.com) by user1 is improved to rank 4 for user3 as oppose to rank 7, previously shown to the user1. The search query suggestion to user 3 is shown in Fig. 7.36.

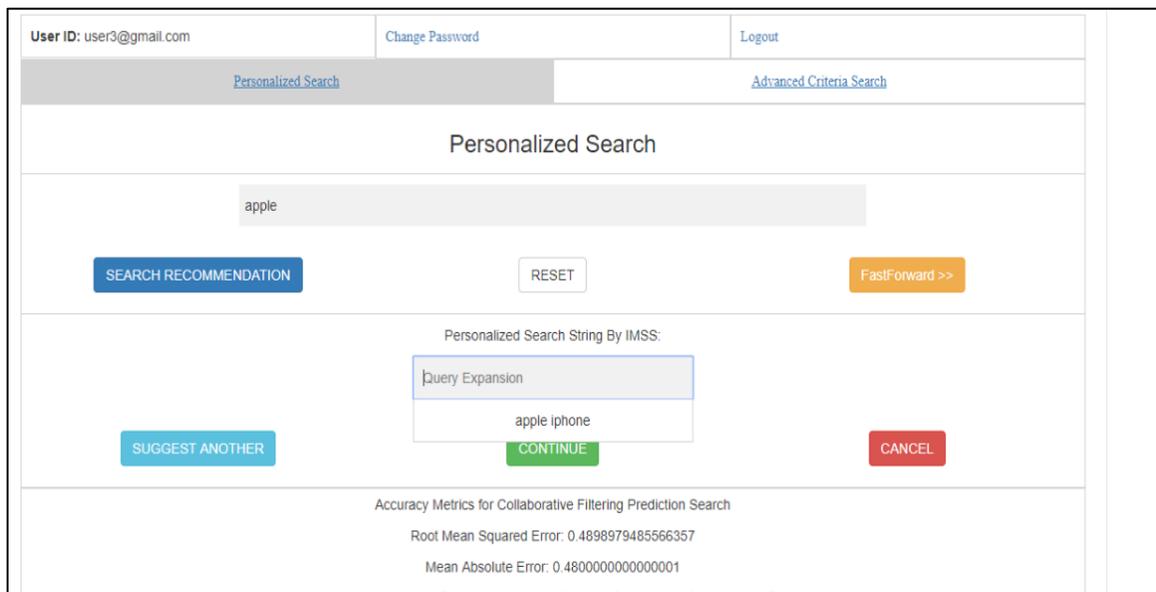


Fig. 7.36 Search suggestion for an incomplete query

The screenshots in Fig. 7.34 and Fig. 7.37 show that the previous rank of the same web link, that is, [https://www. apple.com/](https://www.apple.com/) is improved from rank 7 to rank 4 as the high rating is given to the web link by another user. Hence, the IMSS tool can assist in getting personalized search recommendations and can suitably re-rank the web links to improve user personalized web search experience.

User Similarity : [[0. 1.73205081 1.73205081] [1.73205081 0. 0.] [1.73205081 0. 0.]] Actual: [[3. 2. 3. 2. 3.] [2. 2. 2. 2.] [2. 2. 2. 2.]] Prediction: [[2.6 2.6 2.6 2.6] [2.4 1.4 2.4 1.4 2.4] [2.4 1.4 2.4 1.4 2.4]] Model Root Mean Squared Error: 0.4898979485666357 Model Mean Absolute Error: 0.4800000000000001 Best Match ID: 2 Ranking :					
Priority / Rank	Web Link (Search Engine)	Response Time	Security	Loading speed	Rating
1	https://support.apple.com/en-us/HT201354 (G)	1.37 ms	Secure	1.39 ms	1 ▼
2	https://www.forbes.com/sites/gordonkelly/2018/10/21/apples-new-iphone-has-a-serious-problem/ (G)	1.37 ms	Secure	1.39 ms	1 ▼
3	https://www.theverge.com/2018/10/23/18011306/apple-iphone-xr-review-camera-screen-battery-price (G,B)	1.37 ms	Secure	1.37 ms	1 ▼
4	https://www.apple.com/ (G)	1.37 ms	Secure	1.37 ms	1 ▼
5	https://www.apple.com/shop/buy-iphone/iphone-xs (G)	1.37 ms	Secure	1.38 ms	1 ▼
6	https://www.apple.com/iphone-xs/ (G)	1.37 ms	Secure	1.38 ms	1 ▼
7	https://www.nytimes.com/2018/10/23/technology/personaltech/apple-iphone-xr-review.html (G,B)	1.37 ms	Secure	1.38 ms	1 ▼

Fig. 7.37 Improvement of link rank to rank # 4

7.8 COMPARATIVE ANALYSIS

7.8.1 Comparison with Baselines

Table 7.9 Baseline comparison

S. No	Baseline	Description
1	Arya et al. (2018)	The US patent baseline proposes job search personalization using social network based recommender system with evaluation metrics similarity score. However, present approach offers general web search personalization and in contrast is based on metasearch and machine learning model with comparatively more independent evaluation metrics- MAE, RMSE, Precision, Recall, Specificity, and Sensitivity.

2	Bouadjenk et al. (2016)	The baseline recommends social information based personalization using vector space models and measures personalization extent using MAP and MRR. However, the present approach, in contrast, offers meta-search and machine learning enabled web search personalization with independent evaluation metrics to justify the improvement in the precision of web page ranking.
3	Shafiq et al. (2015)	The authors propose a generic approach to re-ranking of web search results based on social network information of the user and used evaluation metrics like MAP and DCG. However, recent studies have shown that metrics like DCG has a severe drawback regarding an assumption that document at rank i is independent of the documents with ranks less than i. However, the fact is that the document at rank i is dependent on how satisfied a user is with previously traversed results. Here in the present approach, metrics like MAE, RMSE are used and are independent of such assumptions.
4	Bibi et al. (2014)	The baseline proposes a framework for re-ranking of search results obtained from a search engine. However, present approach recommends and implements a metasearch engine which can re-rank the results obtained from three search engines, i.e., Google, Qwant and Bing and uses machine learning and scalable HDFS based big data analytics framework on handling a massive number of links to effectively and efficiently satisfy the personalized needs of the user.

5	Moawad et al. (2012)	The authors propose web search personalization based on multi-agent system and semantic web with evaluation metric as search precision but lag the big data processing capabilities.
6	Collins-Thompson et al. (2011)	The baseline proposes web search personalization based on reading proficiency using semantic-based capabilities with MRR and page reading level as an evaluation metric. However, the baseline does not offer metasearch and big data processing capabilities.
7	Kim et al. (2010)	The baseline proposed web search personalization using a concept network based recommender system and evaluated the same using MAP metric. However, the metasearch and big data capabilities are missing in contrast to current research work.

7.8.2. Comparison between Various Versions of IMSS Tool

The current research work design and develop Advanced Cluster Vector Page Ranking (ACVPR) algorithm and implements the same in the form of Intelligent Meta Search System (IMSS) tool. However, since the inception of current research work, we have step by step improved and extended the capabilities of the initially proposed algorithm to achieve various objectives of the current research work. We have also verified the effectiveness and efficiency of various earlier versions of the ACVPR algorithm as implemented time to time in the form of various versions of the IMSS tool and listed in table 7.10. The different proposed versions of the IMSS tool are also published in our research papers, i.e. (Malhotra & Rishi, 2018a, b), (Malhotra & Rishi, 2017a, b), (Malhotra & Rishi, 2016). The detailed discussion about the interface and capabilities of

different versions of the IMSS tool is also included in section 2.2. The tabular comparison between various versions of proposed and implemented Intelligent Meta Search System (IMSS) to highlight how various versions of IMSS differ from each other is shown in table 7.10.

Table 7.10 Comparison between various versions of IMSS tool

S. No	IMSS Version	Description
1	IMSS-AE [77]	The earlier proposed version by us, i.e., IMSS-AE deploys RV page ranking algorithm for E-Commerce website personalization and is based on HDFS version with significant overhead on account of iterative analytics. Moreover, the IMSS-AE also lacks an appropriate machine learning model to improve web search precision. However, presently deployed version, IMSS is not restricted to any specific domain but offer general web search personalization with adequate machine learning capabilities.
2	IMSS [75]	The earlier proposed version of the deployed metasearch tool uses a system design for personalized query expansion based on semantic web and HDFS platform. Although, old approach discusses map and reduce functions for personalized page ranking but is lacking appropriate page ranking algorithm. However, the present approach proposes a well explained ACVPR algorithm and is well proven to provide effective and efficient personalization through evaluation of machine learning and user survey based personalization metrics.

3	IMSS-E [73]	The IMSS-E version uses Apriori mining, and Map Reduce based framework supported by back propagation neural network and semantic web for E-Commerce website rank personalization. The IMSS-E version uses average search precision as an evaluation metric. However, the present version in current research work deploys logistic regression and collaborative filtering based HDFS framework and perform better regarding average precision metric.
4	Meta Search & Page Ranking Tool [74]	The previously proposed tool by us uses HDFS and Map Reduce based architecture to deploy the CPR algorithm. The previously proposed approach lag appropriate machine learning model to enhance web search personalization experience of the end user. However, the present approach deploys IMSS tool based on logistic regression and collaborative filtering based machine learning models to provide comparatively more satisfactory results to the end users as determined by user survey in the current research work.
5	Web Page Ranking Tool [72]	The earlier proposed approach uses a back-propagation based neural network to implement web search personalization by discussing web page ranking tool. However, the earlier approach does not offer meta-search and big data based scalable processing capabilities as deployed in current research work to better handle the personalized search needs of the users of the modern generation.

7.8.3. IMSS vs. Popular Recommendation Approaches

The performance of collaborative filtering based recommendation module of the pioneered ACVPR algorithm and its implementation, i.e., IMSS tool is evaluated by measuring and comparing evaluation metrics, i.e., Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) against various popular techniques for web search personalization discussed in the literature.

The evaluation regarding the effect on increasing the number of users of the IMSS tool is discussed in detail within section 7.5 and is also summarized in table 7.8. The recommendation module of the deployed system is responsible for predicting the existing best match User ID of the system and hence is quite critical for search query expansion or disambiguation. Here in this section, the recommendation capabilities of the IMSS tool with other popular recommendation approaches are compared by MAE and RMSE metrics. The recommendation approaches considered for the comparison are discussed in the literature and are also summarized by Jiang et al., (2012). As shown in table 7.11, the lowermost values of errors, i.e., MAE and RMSE for the IMSS system demonstrate the fact that the recommendation accuracy of the designed and developed ACVPR algorithm in the current research work is better than various popular approaches proposed in the literature used for online recommendations and search personalization.

Table 7.11 IMSS vs. Popular Recommendation Approaches

Approach	Root Mean Squared Error	Mean Absolute Error
IMSS	0.3364	0.2965
Content-based [7]	0.4769	0.3842
Interpersonal influence based [45]	0.4686	0.3859

Probabilistic matrix based [69]	0.4127	0.3276
Social regularization based [70]	0.3537	0.2985
Feedback based [84]	0.4684	0.3764
Item-based [95]	0.4513	0.3601

7.8.4. IMSS vs. Professional Metasearch Engines

The IMSS tool is also compared with various professional metasearch engines, Dogpile, Mamma, Kartoo and MetaCrawler. As shown in table 7.12, IMSS tool leads over other popular professional metasearch engines. The IMSS tool is the only metasearch tool that enables the end user to select the backend search engine among three background search engines, i.e., Google, Bing, and Qwant within advanced criteria search. Moreover, the IMSS tool allows the end user to experience personalized web search by recommending various search queries of the previous similar users as predicted by the machine learning model. The IMSS tool also allows the end user to sort web links in the order of loading speed. However, the IMSS tool does not allow searching multimedia content that is, images, videos, etc. as provided by Dogpile and Metacrawler engines. The IMSS tool is the only tool that offers detailed machine learning details on its interface to allow researchers to study and further improve the machine learning capabilities of the search engine. Moreover, IMSS tool is the only metasearch tool that provides various search attributes in its output, that is, page loading speed, response time and security to allow the end user to have an idea of the search experience before actually visiting the web link.

Table 7.12 Comparison between IMSS tool and professional metasearch engines

Meta Search Engine	Multimedia Search Options	Backend Search Engine Selection	Personalized Search	Machine Learning Summary	Search Attributes in Output
	Yes	No	No	No	No
	No	No	No	No	No
	No	No	No	No	No
	Yes	No	No	No	No
	No	Yes	Yes	Yes	Yes

7.9. CHAPTER SUMMARY

This chapter discusses two different machine learning approaches used to implement the ACVPR algorithm. The detail regarding implementation and calculation of various evaluation metrics for logistic regression and collaborative filtering based model for the deployment of machine learning capabilities of the IMSS tool is also discussed. The machine learning summary and recommendation capabilities of the recommended tool are shown via relevant metrics like specificity, sensitivity, accuracy for classification based regression model and MAE, RMSE and correlation statistics evaluation metrics. The experimental design, user survey, and verification of query expansion and personalized web page rank improvement are also demonstrated through screenshots of

the live tool. The category of search queries used for training and testing of the system are chosen from various popular search datasets of queries available in Google Zeitgeist 2017. The detailed comparison between different earlier proposed versions of the IMSS tool helps in understanding the improvement and step by step achievement of the various objectives of the current research work. The comparison of the recommended personalization approach with baselines establishes significant improvement regarding capabilities required by modern web search personalization approach. Moreover, comparison of IMSS tool with popular recommendation approaches and professional metasearch engines demonstrate more powerful and user/ developer friendly features of IMSS tool like personalized search, machine learning summary, and statistics of various search parameters like response time, security and page loading speed in the output of the IMSS tool. These statistics help in establishing improvement and enhanced user satisfaction through designed and developed ACVPR algorithm and IMSS tool in current research work when compared with other web search personalization approaches discussed in the literature.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1. CONCLUSION

The significant contribution of present research work is the innovative Advanced Cluster Vector Page Ranking (ACVPR) algorithm and its implementation in the form of an Intelligent Meta Search System (IMSS) for web search personalization. The pioneered personalization approach is backed by intelligent and advanced technologies like machine learning and Hadoop2 based big data analytics to predict meta keywords to suitably expand or suggest the appropriate search query to the user as searched by previous users with matching the profile. The similarity of users and preferred web page ranking order is determined by using collaborative filtering and logistic regression based machine learning models. The deployed metasearch system can competently handle a massive number of web links returned by background search engines by using the second generation of HDFS and Map Reduce based big data analytics framework. The metasearch tool, i.e., IMSS is implemented using machine learning enabled Advanced Cluster Vector Page Ranking Algorithm (ACVPR). The tool can easily predict the personalized relevancy of a prospective web link returned by backend search engines, i.e., Google, Bing, and Qwant for a specific user. The tool can also remove those web links from its final output which are not relevant as evident from user personalized profile, browsing history and the machine learning model decides the same. The machine learning model of the current research work is implemented through two approaches, i.e. (i) Logistic Regression (ii) Collaborative Filtering. The logistic regression model

undergoes extensive training and testing and hence can well predict the response variable, i.e., feedback of a user about a prospective web link. The capabilities of the deployed model are verified by generating a confusion matrix which in turn assisted in the calculation of various evaluation metrics like specificity, sensitivity, TPR, FPR, precision, and recall. The ROC plots between TPR v/s FPR, specificity v/s sensitivity and precision v/s recall establish the effectiveness and efficiency of the regression-based machine learning model used to design and develop innovative ACVPR algorithm and IMSS tool. The diagnostic curves for both feedback_model and feedback_model2 represent the successful generation of the machine learning model for accurate prediction of relevant web links. The collaborative filtering model is used to predict the best match User ID to suitably disambiguate the search queries. The low value of considered evaluation metrics, i.e., Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and machine learning summary depicts the improved machine learning capabilities of the system. Moreover, human volunteers judgment regarding the enhanced precision of various predictor variables of IMSS tool when compared with popular search engines like Google, Bing and Qwant also establish the fact that deployed machine learning model, IMSS tool, and ACVPR algorithm can well satisfy the personalized search needs of the user. The comparison of the pioneered approach with various baselines also establishes the fact that the ACVPR algorithm can improve the personalized search experience of the end user. Moreover features based comparison of the deployed tool with professional metasearch engines, like Dogpile, Kartoo, Mamma and meta-crawler also strengthen the claim of improved personalized search capabilities of the IMSS system and hence more satisfactory end-user experience. The innovative approach to web search personalization as discussed in the current research work is also published in papers by us (Malhotra & Rishi, 2018 a, b). The practical significance for various audiences of the present research work is discussed in sections 8.1.1 to 8.1.3.

8.1.1. Significance for End User

The end user of the pioneered personalized metasearch system is a web user wishing to locate a relevant URL to satisfy his personalized web search requirements. The deployed IMSS tool can assist the end user in searching and listing web links in the personalized order. These web links produced in the output of the IMSS tool are free from biased ranking. The conventional search engines usually show advertised or paid links and hence irrelevant web links on the top of their search output. Thus, the IMSS tool saves a lot of time and energy of the user spent in searching a relevant web link as compared to traditional search engines. The IMSS tool is a metasearch tool which in turn will combine relevant top links from different search engines and hence recall of the result will be better than recall achieved by any of the individual search engines.

8.1.2. Significance for Online Businesses

Online businesses will be motivated to build user-friendly websites rather than just search engine friendly websites. The site that has the potential to satisfy the personalized needs of the user will automatically be listed among the top links in search engine output without any fear of biased ranking or wrong ranking support to paid/incompetent web link by the search engine. This assurance, in turn, will motivate online businesses for positive competition.

8.1.3. Significance for Researchers & Developers

The current research work will motivate researchers and developers to design and develop various metasearch applications by using the potential of machine learning based big data analytics and hence to improve the experience of the end user on the web by incorporating more and more powerful personalized search algorithms. The machine learning summary displayed on the interface of the tool can be used for research and development purpose and hence to further improve the personalized search algorithms.

8.2. FUTURE SCOPE

In future, Advanced Cluster Vector Page Ranking Algorithm (ACVPR) and IMSS tool can be further refined to perform an image-based personalized web search, i.e., search for useful & personalized web links using images. For instance, face recognition based web search can be used to find helpful web links or to locate parents and address of a lost child on WWW or social media using his or her image.

The present research work can also be enriched by incorporating domain-based search tabs on the interface of the IMSS tool. The domain-specific search may include personalized search tabs for e-commerce websites, airline websites to compare and contrast a particular product/ticket/offering from many online businesses. This feature can assist customers in easily searching a specific webpage to satisfy his or her personalized requirements without manually requiring visiting many websites to compare the offerings.

The technological advancements may further improve the performance of the tool; for instance, second-generation HDFS implementation for big data analytics may be replaced by Spark platform specialized for iterative analysis. However, the infrastructural requirements for platforms like Spark are first required to be satisfied to carry out such a change in the future.

SUMMARY

In the present era of big data, web page searching and ranking efficiently on the World Wide Web to satisfy the personalized search needs of the modern user is undoubtedly a major challenge for search engines (Malhotra et al., 2017a). The current research work proposes a novel approach to address the need for web search personalization. Intelligent technologies back the personalization approach in the current research work, i.e., regression and collaborative filtering based machine learning and Hadoop2 - Map Reduce based framework to support big data analytics as required by next-generation search systems. The deployed approach can effectively and efficiently carry out web page re-ranking as evident from calculation and comparison of various relevant evaluation metrics with baselines and feature-based comparison with professional metasearch engines. The efficiency of the inherent machine learning model is verified by plotting different diagnostic and ROC curves. The effectiveness of the pioneered algorithm is further confirmed by user survey and extensive experimental evaluation. The results are plotted in the form of graphs to carry out the precision comparison of the deployed IMSS tool with professional and popular state of the art search engines and metasearch tools. The current research work carries out web search personalization through innovative Advanced Cluster Vector Page Ranking (ACVPR) algorithm and its implementation in the form of a metasearch tool, i.e., Intelligent Meta Search System (IMSS) to suitably expand or disambiguate incomplete or erroneously framed search queries of a user to easily satisfy his or her personalized search needs. The five objectives of the current research work are addressed in detail within eight different chapters of the thesis, and the same is also published in our research papers (Malhotra & Rishi, 2018 a, b), (Malhotra & Rishi, 2017), (Malhotra & Rishi, 2016). The detailed summary and contribution of each chapter are given at the end of each chapter.

BIBLIOGRAPHY & WEBLIOGRAPHY

- [1] Adamopoulos, P. (2014). On discovering non-obvious recommendations: Using unexpectedness and neighborhood selection methods in collaborative filtering systems. Proceedings of the 7th ACM international conference on Web search and data mining, ACM, 655-660.
- [2] Ahmad, M. W., Doja, M. N., & Ahmad, T. (2017). Enumerative feature subset based ranking system for learning to rank in presence of implicit user feedback. Journal of King Saud University-Computer and Information Sciences. Elsevier.
- [3] Al-Smadi, M., Qawasmeh, O., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. (2018) Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews. Journal of computational science, 27, 386-393.
- [4] Alam, M. and Sadaf, K. (2015). Labeling of Web Search Result Clusters using Heuristic Search and Frequent Itemset. Procedia Computer Science, Elsevier, 216-222.
- [5] Aoki, Y., Koshijima, R. and Toyama, M. (2015). Automatic Determination of Hyperlink Destination in Web Index. In Proceedings of the 19th International Database Engineering & Applications Symposium, ACM, 206-207.

- [6] Arya, D., Le, B.H., Venkataraman, G. and Sinha, (2018). Personalized job search and recommendations using job seeker behavioral features. S.D., Microsoft Technology Licensing LLC, 2018. U.S. Patent Application 15/295,505.
- [7] Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3), 66-72.
- [8] Beitzel, S. M., Jensen, E. C., Frieder, O., Grossman, D., Lewis, D. D., Chowdhury, A., & Kolcz, A. (2005, August). Automatic web query classification using labeled and unlabeled training data. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 581-582). ACM.
- [9] Bennett, P. N., White, R. W., Chu, W., Dumais, S. T., Bailey, P., Borisyuk, F., & Cui, X. (2012, August). Modeling the impact of short-and long-term behavior on search personalization. In Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval (pp. 185-194). ACM.
- [10] Bibi, T., Dixit, P., Ghule, R., & Jadhav, R. (2014). Web search personalization using machine learning techniques. In Advance Computing Conference (IACC), 2014 IEEE International (pp. 1296-1299). IEEE.
- [11] Bo, C. and Yang-Mei, L. (2014). Design and Development of Semantic-Based Search Engine Model. Intelligent Computation Technology and Automation (ICICTA), 2014 7th International Conference, IEEE, 145-148.

- [12] Bouadjenek, M. R., Hacid, H., Bouzeghoub, M., & Vakali, A. (2016). Persador: personalized social document representation for improving web search. *Information Sciences*, Elsevier, 369, 614-633.
- [13] Cacheda, F., Carneiro, V., Fernández, D. and Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web (TWEB)*, 5(1), doi 10.1145/1921591.1921593.
- [14] Carterette, B., & Jones, R. (2008). Evaluating search engines by modeling the relationship between relevance and clicks. In *Advances in Neural Information Processing Systems* (pp. 217-224).
- [15] Chawla, S. (2018). Web Page Recommender System using hybrid of Genetic Algorithm and Trust for Personalized Web Search. *Journal of Information Technology Research (JITR)*, 11(2), 110-127.
- [16] Chen, C. P., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, Elsevier, 275, 314-347.
- [17] Chi, C. C., Kuo, C. H., Lu, M. Y., & Tsao, N. L. (2008, July). Concept-based pages recommendation by using cluster algorithm. In *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on* (pp. 298-300). IEEE.
- [18] Chirita, P. A., Olmedilla, D., & Nejdl, W. (2004, September). Finding related pages using the link structure of the WWW. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 632-635). IEEE Computer Society.

- [19] Chirita, P. A., Nejdl, W., Paiu, R., & Kohlschütter, C. (2005, August). Using ODP metadata to personalize search. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 178-185). ACM.
- [20] Cho, Y. H., Kim, J. K., & Kim, S. H. (2002). A personalized recommender system based on web usage mining and decision tree induction. *Expert systems with Applications*, 23(3), 329-342.
- [21] Chua, C. E. H., Chiang, R. H., & Storey, V. C. (2009). Building customized search engines: An interoperability architecture. *International Journal of Intelligent Information Technologies (IJIT)*, 5(3), 1-27.
- [22] Collins-Thompson, K., Bennett, P. N., White, R. W., De La Chica, S., & Sontag, D. (2011, October). Personalizing web search results by reading level. In Proceedings of the 20th ACM international conference on Information and knowledge management (pp. 403-412). ACM.
- [23] Daoud, M., Tamine, L., & Boughanem, M. (2009, April). A contextual evaluation protocol for a session-based personalized search. In Workshop on Contextual Information Access, Seeking and Retrieval Evaluation (In conjunction with European Conference on Information retrieval-ECIR), Toulouse, France, Springer (pp. 105).
- [24] Debnath, S., Mitra, P., Pal, N., & Giles, C. L. (2005). Automatic identification of informative sections of web pages. *IEEE transactions on knowledge and data engineering*, 17(9), 1233-1246.
- [25] Dou, Z., Song, R., & Wen, J. R. (2007, May). A large-scale evaluation and analysis of personalized search strategies. In Proceedings of the 16th international conference on World Wide Web (pp. 581-590). ACM.

- [26] Eirinaki, M., & Vazirgiannis, M. (2005, November). Usage-based page rank for web personalization. In *Data Mining, Fifth IEEE International Conference on* (pp. 8-pp). IEEE.
- [27] Fang, J., Guo, L., Wang, X., & Yang, N. (2007, August). Ontology-based automatic classification and ranking for web documents. In *fskd* (pp. 627-631). IEEE.
- [28] Ferretti, S., Mirri, S., Prandi, C., & Salomoni, P. (2016). Automatic web content personalization through reinforcement learning. *Journal of Systems and Software, Elsevier*, 121, 157-169.
- [29] Gauch, S., Speretta, M., Chandramouli, A., & Micarelli, A. (2007). User profiles for personalized information access. In *The adaptive web* (pp. 54-89). Springer, Berlin, Heidelberg.
- [30] Gebara, F., Hofstee, H, & Nowka, K. (2015). *Second Generation Big data Systems: Cover Feature Outlook*, IEEE Computer Society, IEEE, 36-41.
- [31] Ghegade, P. C., & Wadane, V. (2014). A Survey of Personalized Web Search in Current Techniques. *International Journal of Computer Science and Information Technologies*, 5(6).
- [32] Glover, E. J., Lawrence, S., Gordon, M. D., Birmingham, W. P., & Giles, C. L. (1999). Recommending web documents based on user preferences. *Ann Arbor*, 1001, 48109-2110.
- [33] Glover, E. J., Lawrence, S., Birmingham, W. P., & Giles, C. L. (1999, November). Architecture of a metasearch engine that supports user information needs. In *Proceedings of the eighth international conference on Information and knowledge management* (pp. 210-216). ACM.

- [34] Gollub, T., Genc, E., Lipka, N., & Stein, B. (2018). Pseudo Descriptions for Meta-Data Retrieval. In Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval (pp. 139-146). ACM.
- [35] Gomez-Nieto, E., San Roman, F., Pagliosa, P., Casaca, W., Helou, E.S., de Oliveira, M.C.F. and Nonato, L.G. (2014). Similarity preserving snippet-based visualization of web search results. IEEE transactions on visualization and computer graphics, 20(3), 457-470.
- [36] Guy, I., Jaimes, A., Agulló, P., Moore, P., Nandy, P., Nastar, C. and Schinzel, H. (2010). Will recommenders kill search?: Recommender systems-an industry perspective. Proceedings of the fourth ACM conference on Recommender systems, ACM, 7-12.
- [37] Hannak, A., Sapiezynski, P., Molavi Kakhki, A., Krishnamurthy, B., Lazer, D., Mislove, A., & Wilson, C. (2013, May). Measuring personalization of web search. In Proceedings of the 22nd international conference on World Wide Web (pp. 527-538). ACM.
- [38] Haveliwala, T. H., Gionis, A., Klein, D., & Indyk, P. (2002, May). Evaluating strategies for similarity search on the web. In Proceedings of the 11th international conference on World Wide Web (pp. 432-442). ACM.
- [39] Hoeber, O., & Yang, X. D. (2006, July). The Visual Exploration of Web Search Results Using Hot Map. In Information Visualization, 2006. IV 2006. Tenth International Conference on (pp. 157-165). IEEE.

- [40] Hou, J., & Zhang, Y. (2003). Effectively finding relevant web pages from linkage information. *IEEE Transactions on Knowledge and Data Engineering*, 15(4), 940-951.
- [41] Howe, A. E., & Dreilinger, D. (1997). SAVVYSEARCH: A metasearch engine that learns which search engines to query. *Ai Magazine*, 18(2), 19.
- [42] Hsu, Y. W., Moon, N., & Singh, R. (2006, December). Designing interaction paradigms for web-information search and retrieval. In *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on* (pp. 815-822). IEEE.
- [43] Hu, J. (2008). Personalized web search by using learned user profiles in re-ranking. *KDD Conference*, 84-97.
- [44] Huang, D., & Li, H. (2008, December). The research of web page recommendation model based on FCA and enterprise ontology. In *Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on* (Vol. 1, pp. 232-236). IEEE.
- [45] Huang, J., Cheng, X., Guo, J., Shen, H., & Yang, K. (2010, August). Social Recommendation with Interpersonal Influence. In *ECAI* (Vol. 10, pp. 601-606).
- [46] Jiang, M., Cui, P., Liu, R., Yang, Q., Wang, F., Zhu, W., & Yang, S. (2012, October). Social contextual recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 45-54). ACM.
- [47] Jung, S., Harris, K., Webster, J. and Herlocker, J.L. (2004). SERF: integrating human recommendations with search. *Proceedings of the*

thirteenth ACM international conference on Information and knowledge management, ACM, 571-580.

- [48] Kajaba, M., Navrat, P., & Chuda, D. (2009). A simple personalization layer improving relevancy of web search. *Computing and Information Systems Journal*, 13(3), 29-35.
- [49] Kakulapati, V., Vasumathi, D., & Jena, S. (2013). Survey on Web Search Results Personalization Techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, issue 11, ISSN: 2277 128X.
- [50] Kamvar, S. D., Haveliwala, T. H., Manning, C. D., & Golub, G. H. (2003, May). Extrapolation methods for accelerating PageRank computations. In *Proceedings of the 12th international conference on World Wide Web* (pp. 261-270). ACM.
- [51] Kim, H. J., Lee, S., Lee, B., & Kang, S. (2010, August). Building concept network-based user profile for personalized web search. In *Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on* (pp. 567-572). IEEE.
- [52] Khurana, A. (2014). Bringing big data systems to the cloud. *IEEE Cloud Computing*, 1(3), IEEE, 72-75.
- [53] Kim, H. R., & Chan, P. K. (2005, August). Personalized ranking of search results with learned user interest hierarchies from bookmarks. In *Proc. of WebKDD* (pp. 21-24).
- [54] Krriztiann, V. S. (2018). On-Page SEO: How to Make Google Fall in Love with Your Website *Google Marketing Strategies and Tricks*.

- [55] Kumar, R., & Sharan, A. (2014, February). Personalized web search using browsing history and domain knowledge. In *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on* (pp. 493-497). IEEE.
- [56] Kuppusamy, K.S., and Aghila, G. (2014). CaSePer: An efficient model for personalized web page change detection based on segmentation. *Journal of King Saud University-Computer and Information Sciences*, 26(1), Elsevier, 19-27.
- [57] Lamberti, F., Sanna, A., & Demartini, C. (2009). A relation-based page rank algorithm for semantic web search engines. *IEEE Transactions on Knowledge and Data Engineering*, 21(1), 123-136.
- [58] Lee, H. C., & Borodin, A. (2009). Criteria for cluster-based personalized search. *Internet Mathematics*, 6(3), 399-435.
- [59] Lee, U., Liu, Z., & Cho, J. (2005, May). Automatic identification of user goals in web search. In *Proceedings of the 14th international conference on World Wide Web* (pp. 391-400). ACM.
- [60] Li, L., Yang, Z., Wang, B., & Kitsuregawa, M. (2007). Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In *Advances in Data and Web Management* (pp. 228-240). Springer, Berlin, Heidelberg.
- [61] Li, Y., Wang, Y., & Huang, X. (2007). A relation-based search engine in semantic web. *IEEE transactions on knowledge and data engineering*, 19(2).

- [62] Li, Y., & Zhong, N. (2006). Mining ontology for automatically acquiring web user information needs. *IEEE transactions on Knowledge and Data Engineering*, 18(4), 554-568.
- [63] Limbu, D.K., Connor, A., Pears, R. and MacDonell, S. (2006). Contextual relevance feedback in web information retrieval. *Proceedings of the 1st International Conference on Information Interaction in Context*, ACM, 138-143.
- [64] Liu, F., Yu, C., & Meng, W. (2002, November). Personalized web search by mapping user queries to categories. In *Proceedings of the eleventh international conference on Information and knowledge management* (pp. 558-565). ACM.
- [65] Liu, F., Yu, C., & Meng, W. (2004). Personalized web search for improving retrieval effectiveness. *IEEE Transactions on knowledge and data engineering*, 16(1), 28-40.
- [66] Liu, Y., Bi, J.W. and Fan, Z.P. (2017). Ranking products through online reviews: A method based on sentiment analysis technique and intuitionistic fuzzy set theory. *Information Fusion*, 36, 149-161.
- [67] Lu, M., Zhou, Q., Li, F., Lu, Y., & Zhou, L. (2002). Recommendation of web pages based on concept association. In *Advanced Issues of E-Commerce and Web-Based Information Systems, 2002.(WECWIS 2002). Proceedings. Fourth IEEE International Workshop on* (pp. 221-227). IEEE.
- [68] Makris, C., Panagis, Y., Sakkopoulos, E., & Tsakalidis, A. (2007). Category ranking for personalized search. *Data & Knowledge Engineering*, 60(1), 109-125.

- [69] Ma, H., Yang, H., Lyu, M. R., & King, I. (2008, October). Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management* (pp. 931-940). ACM.
- [70] Ma, H., Zhou, D., Liu, C., Lyu, M. R., & King, I. (2011, February). Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining* (pp. 287-296). ACM.
- [71] Malhotra, D. and Verma, N. (2013). An ingenious pattern matching approach to ameliorate web page rank. *International Journal of Computer Applications*, 65(24), 33-39.
- [72] Malhotra, D. (2014). Intelligent web mining to ameliorate Web Page Rank using Back-Propagation neural network. *Confluence the Next Generation Information Technology Summit (Confluence), 5th International Conference*, IEEE, 77-81.
- [73] Malhotra, D. and Rishi, O.P. (2016). IMSS-E: An Intelligent Approach to Design of Adaptive Meta Search System for E-Commerce Website Ranking. *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*, ACM, doi>10.1145/2979779.2979782.
- [74] Malhotra, D., Malhotra, M. and Rishi, O.P. (2017a). An Innovative Approach of Web Page Ranking Using Hadoop- and Map Reduce-Based Cloud Framework. *Proceedings of Advances in Intelligent Systems and Computing*, Vol.654, CSI-2015, Springer, 421-427.

- [75] Malhotra, D. and Rishi, O.P. (2017). IMSS: A Novel Approach to Design of Adaptive Search System Using Second Generation Big data Analytics. Proceedings of International Conference on Communication and Networks, Springer, 189-196.
- [76] Malhotra, D., Verma, N., Rishi, O.P. and Singh, J. (2017b). Intelligent Big data Analytics: Adaptive E-Commerce Website Ranking Using Apriori Hadoop–BDAS-Based Cloud Framework. Maximizing Business Performance and Efficiency through Intelligent Systems, IGI Global, 50-72.
- [77] Malhotra, D., & Rishi, O. P. (2018a). An intelligent approach to design of E-Commerce metasearch and ranking system using next-generation big data analytics. Journal of King Saud University-Computer and Information Sciences, <https://doi.org/10.1016/j.jksuci.2018.02.015>, Elsevier.
- [78] Malhotra, D., & Rishi, O. P. (2018b). IMSS-P: An Intelligent Approach to Design & Development of Personalized Meta Search & Page Ranking System, Journal of King Saud University-Computer and Information Sciences, <https://doi.org/10.1016/j.jksuci.2018.11.013>, Elsevier.
- [79] Malthankar, S. V., & Kolte, S. (2016). Client Side Privacy Protection Using Personalized Web Search. Procedia Computer Science, Elsevier, 79, 1029-1035.
- [80] Maratea, A., & Petrosino, A. (2009, November). An heuristic approach to page recommendation in web usage mining. In Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on (pp. 1043-1048). IEEE.

- [81] Marchiori, M. (1997). The quest for correct information on the web: Hyper search engines. *Computer Networks and ISDN Systems*, 29(8), 1225-1236.
- [82] Matthijs, N., & Radlinski, F. (2011, February). Personalizing web search using long term browsing history. In *Proceedings of the fourth ACM international conference on Web search and data mining* (pp. 25-34). ACM.
- [83] Moawad, I. F., Talha, H., Hosny, E., & Hashim, M. (2012). Agent-based web search personalization approach using dynamic user profile. *Egyptian Informatics Journal*, 13(3), 191-198.
- [84] Moghaddam, S., Jamali, M., Ester, M., & Habibi, J. (2009, October). FeedbackTrust: using feedback effects in trust-based recommendation systems. In *Proceedings of the third ACM conference on Recommender systems* (pp. 269-272). ACM.
- [85] Nakayama, T., Kato, H., & Yamane, Y. (2000). Discovering the gap between Web site designers' expectations and users' behavior. *Computer Networks*, 33(1-6), 811-822.
- [86] Pare, S., & Vasgi, B. (2014). Personalization of the Web Search. *IJMER*, Vol. 4, Issue 10, 59-70.
- [87] Patil, M. A., Ghonge, M. M., & Sarode, M. V (2014). User customizable Privacy-preserving Search Framework-UPS for Personalized Web Search. *International Journal of Research in Advent Technology*, vol. 2, no.4, E-ISSN: 2420-14141.

- [88] Peng, W. C., & Lin, Y. C. (2006, June). Ranking web search results from personalized perspective. In E-Commerce Technology, 2006. The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services, The 3rd IEEE International Conference on (pp. 12-12). IEEE.
- [89] Prabowo, R., Jackson, M., Burden, P., & Knoell, H. D. (2002, December). Ontology-based automatic classification for the web pages: Design, implementation and evaluation. In null (p. 182). IEEE.
- [90] Pretschner, A., & Gauch, S. (1999). Ontology based personalized search. In Tools with artificial intelligence, 1999. Proceedings. 11th IEEE international conference on (pp. 391-398). IEEE.
- [91] Qi, X., & Davison, B. D. (2009). Web page classification: Features and algorithms. *ACM computing surveys (CSUR)*, 41(2).
- [92] Radlinski, F., & Dumais, S. (2006, August). Improving personalized web search using result diversification. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 691-692). ACM.
- [93] Rasekh, I., (2015). A new competitive intelligence-based strategy for web page search. *Procedia Computer Science*, Elsevier, 450-456.
- [94] Salonen, V., & Karjaluoto, H. (2016). Web personalization: the state of the art and future avenues for research and practice. *Telematics and Informatics*, Elsevier, 33(4), 1088-1104.

- [95] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295). ACM.
- [96] Sakkopoulos, E., Antoniou, D., Adamopoulou, P., Tsirakis, N., & Tsakalidis, A. (2010). A web personalizing technique using adaptive data structures: The case of bursts in web visits. *Journal of Systems and Software*, 83(11), 2200-2210.
- [97] Sendhilkumar, S., & Geetha, T. V. (2007, July). An Evaluation of Personalized Web Search for an Individual User. In *Artificial Intelligence and Pattern Recognition* (pp. 484-490).
- [98] Sieg, A., Mobasher, B., & Burke, R. D. (2007). Learning ontology-based user profiles: A semantic approach to personalized web search. *IEEE Intelligent Informatics Bulletin*, 8(1), 7-18.
- [99] Shafiq, O., Alhaji, R., & Rokne, J. G. (2015). On personalizing Web search using social network analysis. *Information Sciences*, Elsevier 314, 55-76.
- [100] Sharma, D. K., & Sharma, A. K. (2010). Deep web information retrieval process: A technical survey. *International Journal of Information Technology and Web Engineering (IJITWE)*, 5(1), 1-22.
- [101] Shou, L., Bai, H., Chen, K. and Chen, G. (2014). Supporting privacy protection in personalized web search. *IEEE transactions on knowledge and data engineering*, 26(2), IEEE, 453-467.

- [102] Sieg, A., Mobasher, B., & Burke, R. D. (2007). Learning ontology-based user profiles: A semantic approach to personalized web search. *IEEE Intelligent Informatics Bulletin*, 8(1), 7-18.
- [103] Singh, A. and Vélez, H.G. (2014). Hierarchical multi-log cloud-based search engine. *Complex, Intelligent and Software Intensive Systems (CISIS)*, Eighth International Conference, IEEE, 211-219.
- [104] Singh, D. and Reddy, C.K. (2015). A survey on platforms for big data analytics. *Journal of Big data*, 2(1), p.8, doi> 10.1186/s40537-014-0008-6.
- [105] Speretta, M., & Gauch, S. (2000). Personalizing search based on user search histories. *Proc. of CIKM 2004*.
- [106] Su, J. H., Wang, B. W., & Tseng, V. S. (2008, December). Effective ranking and recommendation on web page retrieval by integrating association mining and PageRank. In *Web Intelligence and Intelligent Agent Technology. WI-IAT'08. IEEE/WIC/ACM International Conference on* (Vol. 3, pp. 455-458). IEEE.
- [107] Sudhakar, P., Poonkuzhali, G., & Kumar, R. K. (2012). Content Based Ranking for Search Engines. In *Proc. International Multi Conference of Engineers and Computer Scientists (IMECS 12)*.
- [108] Sugiyama, K., Hatano, K. and Yoshikawa, M. (2004). Adaptive web search based on user profile constructed without any effort from users. *Proceedings of the 13th International Conference on World Wide Web, ACM*, 675-684.
- [109] Swamynathan, M. (2017). *Mastering Machine Learning with Python in Six Steps*. Apress, Springer.

- [110] Takano, K., & Li, K. F. (2009, May). An adaptive personalized recommender based on web-browsing behavior learning. In *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on* (pp. 654-660). IEEE.
- [111] Tanapaisankit, P., Watrous-deVersterre, L. and Song, M. (2012). Personalized query expansion in the QIC system. *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries, ACM*, 259-262.
- [112] Tao, W. X., & Zuo, W. L. (2003, November). Query-sensitive self-adaptable web page ranking algorithm. In *Machine Learning and Cybernetics, 2003 International Conference on* (Vol. 1, pp. 413-418). IEEE.
- [113] Tsai, C.W., Lai, C.F., Chao, H.C. and Vasilakos, A.V. (2015). Big data analytics: a survey. *Journal of Big data*, 2(1), p.21, doi> 10.1186/s40537-015-0030-3.
- [114] Verma, N., Malhotra, D., Malhotra, M. and Singh, J. (2015). E-commerce website ranking using semantic web mining and neural computing. *Procedia Computer Science, Science Direct, Elsevier*, 42-51.
- [115] Verma, N. and Singh, J. (2017). An intelligent approach to Big Data analytics for sustainable retail environment using Apriori-MapReduce framework. *Industrial Management & Data Systems*, 117(7), Emerald, 1503-1520.
- [116] Verma, N. And Singh, J. (2017). A comprehensive review from sequential association computing to Hadoop MapReduce parallel computing in a retail scenario. *Journal of management analytics*, Taylor and Francis, doi> 10.1080/23270012.2017.1373261.

- [117] Vinay, V., Wood, K., Milic-Frayling, N. and Cox, I.J. (2005). Comparing relevance feedback algorithms for web search. In Special interest tracks and posters of the 14th international conference on World Wide Web, ACM, 1042-1053.
- [118] Wang, S., Xu, K., Zhang, Y. and Li, F. (2011). Search engine optimization based on algorithm of BP neural networks. In Computational Intelligence and Security (CIS), Seventh International Conference, IEEE, 390-394.
- [119] Wang, H., He, M., Zhou, L., Li, Z., Zhan, H., & Wang, R. (2018). Remove-Duplicate Algorithm Based on Meta Search Result. In International Conference on Cloud Computing and Security (pp. 33-44). Springer, Cham.
- [120] Wang, H. and Wong, K. (2014). Personalized search: An interactive and iterative approach. In Services (SERVICES), IEEE World Congress, IEEE, 3-10.
- [121] Wasid, M. and Kant, V. (2015). A particle swarm approach to collaborative filtering based recommender systems through fuzzy features. *Procedia Computer Science*, 54, Elsevier, 440-448.
- [122] White, R. W., Jose, J. M., & Ruthven, I. (2003). A task-oriented study on the influencing effects of query-biased summarization in web searching. *Information Processing & Management*, 39(5), 707-733.
- [123] Xiang, B., Jiang, D., Pei, J., Sun, X., Chen, E., & Li, H. (2010). Context-aware ranking in web search. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, ACM, 451-458.

- [124] Yang, Y. F., Hwang, S. L., & Schenkman, B. (2012). An improved Web search engine for visually impaired users. *Universal Access in the Information Society*, 11(2), 113-124.
- [125] Youssif, A.A., Ghalwash, A.Z. and Amer, E.A. (2011). HSWS: Enhancing efficiency of web search engine via semantic web. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, ACM, 212-219.
- [126] Yu, X., & Sun, S. (2010, May). Research on personalized recommendation system based on web mining. In *E-Business and E-Government (ICEE), 2010 International Conference on* (pp. 346-349). IEEE.
- [127] Zhang, G., Li, C. and Xing, C. (2012). A Semantic++ Social Search Engine Framework in the Cloud. In *Semantics, Knowledge and Grids (SKG), Eighth International Conference*, IEEE, 270-278.
- [128] Zhou, D., Zhao, W., Wu, X., Lawless, S., & Liu, J. (2018). An iterative method for personalized results adaptation in cross-language search. *Information Sciences*, Elsevier, 430, 200-215.
- [129] Zhu, H., Ou, C. X., Van den Heuvel, W. J. A. M., & Liu, H. (2017). Privacy calculus and its utility for personalization services in e-commerce: An analysis of consumer decision-making. *Information & Management*, Elsevier, 54(4), 427-437.
- [130] Zhou, Y., & Croft, W. B. (2007, July). Query performance prediction in web search environments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 543-550). ACM.

- [131] Zoldi, S. M., Sowani, A. K., Kumar, U., & Chawla, S. S. (2018). U.S. Patent Application No. 15/442,434.
- [132] <https://www.reuters.com/article/us-india-google-antitrust/indias-antitrust-watchdog-fines-Google-for-abusing-dominant-position-idUSKBN1FS2AD>, accessed on 12/02/2018.
- [133] <https://www.optimizely.com/optimization-glossary/website-personalization/>, accessed on 10/03/2018.
- [134] <https://searchenginewatch.com/tag/reuters/>, accessed on 25/03/2018.
- [135] <https://www.reuters.com/article/us-internet-search/former-googleers-unveil-cuil-a-new-search-engine-idUSN2741939220080728>, accessed on 01/04/2018.
- [136] http://cs.thomsonreuters.com/ua/webbuild/cs_us_en/misc/improve_serach_ranking.htm, accessed on 10/4/2018.
- [137] https://books.google.co.in/books?id=tEZj2sG6IKEC&pg=PA22&lpg=PA22&dq=meta+search+engine+%2B+reuters&source=bl&ots=X352zTHrHY&sig=T8Wo5S7ptsznsY4Hy8jWJCu11y0&hl=en&sa=X&ved=2ahUKEwj4_IaI06jdAhXGbSsKHUHsDLMQ6AEwCXoECAEQAAQ#v=onepage&q=meta%20search%20engine%20%2B%20reuters&f=false, accessed on 15/04/2018.
- [138] <https://www.searchenginepeople.com/blog/10-meta-search-engines-reviewed-and-compared.html>, accessed on 15/04/2018.
- [139] <https://www.lifewire.com/best-search-engines-2483352>, accessed on 01/05/2018.

- [140] <https://www.thomsonreuters.com/content/dam/openweb/documents/pdf/financial/why-big-data-is-a-big-deal.pdf>, accessed on 03/05/2018.
- [141] <https://www.reuters.com/brandfeatures/venture-capital/article?Id=28830>, accessed on 10/05/2018.
- [142] <https://www.gartner.com/it-glossary/analytics/>, accessed on 02/06/2018.
- [143] <https://www.forbes.com/sites/ciocentral/2018/02/28/gartner-magic-quadrant-whos-winning-in-the-data-machine-learning-space/#41894b27dab3>, accessed on 15/06/2018.
- [144] <https://www.gartner.com/it-glossary/machine-learning>, accessed on 25/06/2018.
- [145] <https://www.gartner.com/webinar/3597718>, accessed on 01/07/2018.
- [146] <https://www.datanami.com/2018/02/28/winners-losers-gartners-data-science-ml-platform-report/>, accessed on 10/07/2018.
- [147] <https://www.marketwatch.com/press-release/thomson-reuters-unveils-new-legal-research-platform-with-advanced-ai-westlaw-edge-2018-07-12>, accessed on 15/07/2018.
- [148] <https://www.acm.org/education/ai-ml-webinars>, accessed on 20/07/2018.
- [149] <https://on.acm.org/t/adversarial-machine-learning/764>, accessed on 29/07/2018.
- [150] <https://www.thomsonreuters.com/en/reports/2018-ai-predictions.html>, accessed on 01/08/2018.

- [151] <https://www.cloudera.com/more/customers/thomson-reuters.html>, accessed on 11/08/2018.
- [152] <https://www.knime.com/about/news/gartner-recognizes-knime-as-leader-in-data-science-and-machine-learning-platforms>, accessed on 07/09/2018.
- [153] <https://stackoverflow.com/questions/28359125/yarn-site-xml-missing>, accessed on 20/12/2018.
- [154] <https://scialert.net/fulltext/?doi=jai.2013.82.88>, accessed on 20/12/2018.
- [155] <http://www10.org/cdrom/papers/519/node5.html>, accessed on 26/12/2018.
- [156] <https://www.lifewire.com/google-zeitgeist-3481903>, accessed on 26/12/2018.
- [157] https://www.tutorialspoint.com/hadoop/hadoop_multi_node_cluster.htm, accessed on 02/01/2018.
- [158] <https://www.linode.com/docs/databases/hadoop/how-to-install-and-set-up-hadoop-cluster/>, accessed on 02/06/2018
- [159] <https://backtobasics.com/big-data/hadoop/simple-explanation-of-hadoop-core-componentshdfs-and-mapreduce/>, accessed on 05/08/2018
- [160] <http://ioenotes.edu.np/media/notes/big-data/kec-notes/Hadoop.pptx>, accessed on 06/08/2018
- [161] <http://blog.socratesk.com/assets/pdf/Hadoop.pdf>, accessed on 06/08/2018
- [162] <http://pramodgampa.blogspot.com/2013/06/the-building-blocks-of-hadoop.html>, accessed on 07/08/2018
- [163] <https://www.bbc.com/news/technology-40406542>, accessed on 07/08/18

- [164] <http://osaipl.com/cloud-training.html>, accessed on 07/08/18
- [165] http://www.ijmer.com/papers/Vol4_Issue10/Version-1/H04010_01-5970.pdf, accessed on 15/07/18
- [166] https://link.springer.com/chapter/10.1007/978-981-10-2750-5_20 ,
accessed on 18/06/18
- [167] <http://a4academics.com/component/attachments/download/480>, accessed
on 10/12/18

PUBLICATIONS

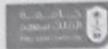
- [1] Malhotra, D., & Rishi, O. P. (2018, November). IMSS-P: An Intelligent Approach to Design & Development of Personalized Meta Search & Page Ranking System. **Journal of King Saud University-Computer and Information Sciences- Elsevier**, Scopus and ESCI Indexed, Web of Science- Clarivate Analytics, Thomson Reuters
<https://doi.org/10.1016/j.jksuci.2018.11.013>
- [2] Malhotra, D., & Rishi, O. P. (2018, March). An Intelligent Approach to Design of E-Commerce Meta Search and Ranking System using Next-Generation Big Data Analytics. **Journal of King Saud University-Computer and Information Sciences- Elsevier**, Scopus and ESCI Indexed , Web of Science - Clarivate Analytics, Thomson Reuters
<https://doi.org/10.1016/j.jksuci.2018.02.015>
- [3] Malhotra, D. and Rishi, O.P. (2017, April). IMSS: A Novel Approach to Design of Adaptive Search System Using Second Generation Big data Analytics. Proceedings of International Conference on Communication and Networks, **Springer**, Scopus Indexed
https://doi.org/10.1007/978-981-10-2750-5_20
- [4] Malhotra, D. and Rishi, O.P. (2016, August). IMSS-E: An Intelligent Approach to Design of Adaptive Meta Search System for E-Commerce Website Ranking”. Proceedings of the International Conference on Advances

in Information Communication Technology & Computing, **ACM**, Scopus
Indexed

<https://dl.acm.org/citation.cfm?doid=2979779.2979782>

Communicated Papers (Under Review)

- [5] Malhotra, D., & Rishi, O. P. An Ingenious Approach to Design of Personalized Meta-Search and Page Ranking System using Machine Learning and Big Data Analytics.
- [6] Malhotra, D., & Rishi, O. P. A comprehensive review from Hyperlink to intelligent technologies based personalized search systems.



An intelligent approach to design of E-Commerce metasearch and ranking system using next-generation big data analytics

Dheeraj Malhotra^{*}, O.P. Rishi

Department of Computer Science and Informatics, University of Kota, Kota, Rajasthan 324 005, India

ARTICLE INFO

Article history:

Received 18 October 2017

Revised 22 February 2018

Accepted 28 February 2018

Available online xxx

Keywords:

E-Commerce website ranking

IMSS- AE tool

RV page ranking algorithm

Second generation big data analytics

Hadoop-MapReduce

Personalized page ranking

ABSTRACT

The purpose of this research work is to explore various limitations of conventional search and page ranking systems in an E-Commerce environment. The key objective is to assist customers in making an online purchase decision by providing personalized page ranking order of E-Commerce web links in response to E-Commerce query by analyzing the customer preferences and browsing behavior. This research work first employs an orderly and category wise literature review. The findings reveal that conventional search systems have not evolved to support big data analysis as required by modern E-Commerce environment. This work aims to develop and implement second-generation HDFS- MapReduce based innovative page ranking algorithm, i.e. Relevancy Vector (RV) algorithm. This research equips the customer with a robust metasearch tool, i.e. IMSS-AE to easily understand personalized search requirements and purchase preferences of customer. The proposed approach can well satisfy all critical parameters such as scalability, partial failure support, extensibility as expected from next-generation big data processing systems. An extensive and comprehensive experimental evaluation shows the efficiency and effectiveness of proposed RV page ranking algorithm and IMSS-AE tool over and above other popular search engines.

© 2018 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

In this modern era of big data, the shopping activity is modified a lot because of enormous growth in online shopping websites, also known as E-Tailers. The new age customers prefer to shop through these online portals because of various attractions in countries like India such as easy and cheap availability of the Internet. The primary reason is intense competition between telecoms, for instance, *Reliance Jio* prime membership offers free unlimited internet data usage for three months for all of its users in nominal charges. Some of the other reasons include lucrative cash back and easy returns without deduction of shipping charges from portals like *PayTm*, Cash on delivery type regular features from E-Commerce sites like *Flipkart*, *Amazon*, and other E-Tailers. Moreover, searching a suitable E-Commerce website to best suit the customer purchase requirements is not so easy as customers are

primarily dependent on conventional search engines like *Google*, *Bing* to find a suitable E-Commerce web site. However, when different users search the same E-Commerce query, even a most advanced and popular search engine retrieves the same result as discussed by Gomez-Nieto et al. (2014). Thus, irrespective of the background and personalized tastes of customer submitting the query as most of the modern search engines tend to return the results by interpreting the E-Commerce query in various possible ways. Moreover, if the query is ambiguous or incomplete, then the situation will get even worse as discussed by Malhotra and Verma (2013). For instance, for the incomplete E-Commerce search query “*Galaxy*”, some customers may be interested in links to buy a new *Samsung Galaxy* series mobile phone, while another customer may be interested in searching links for online booking of tickets for a movie *Guardians of the Galaxy Vol. 2*. Hence, there is an urgent need for personalized E-Commerce search system. The personalized system may modify the E-Commerce search query by keeping track of customer’s preferences by maintaining his/her profile, search preferences through browsing history, etc. over a period and return results in correct order of ranking with customer’s relevant output links on top to best suit the customer requirements (See Fig. 1).

E-Commerce data is explosively increasing on the scale of Tera-Bytes (TB) to PetaBytes (PB) on a daily basis due to the continuous

^{*} Corresponding author.

E-mail address: dheerajmalhotra4@gmail.com (D. Malhotra).

Peer review under responsibility of King Saud University.



Production and hosting by Elsevier

<https://doi.org/10.1016/j.jksuci.2018.02.015>

1319–1578/© 2018 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Please cite this article in press as: Malhotra, D., Rishi, O.P. An intelligent approach to design of E-Commerce metasearch and ranking system using next-generation big data analytics. *Journal of King Saud University – Computer and Information Sciences* (2018). <https://doi.org/10.1016/j.jksuci.2018.02.015>

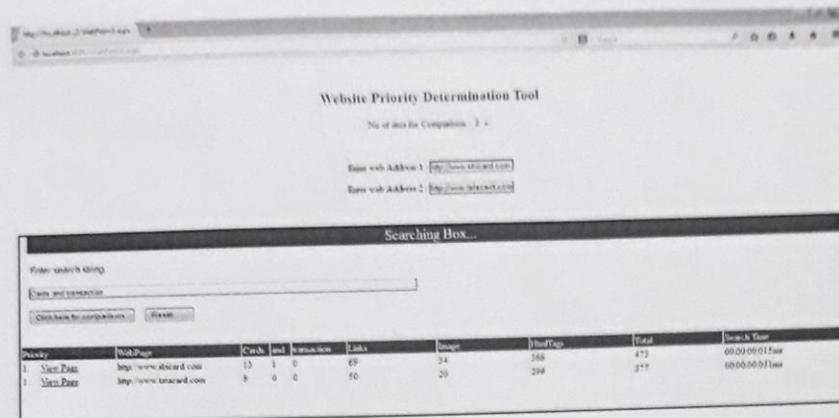


Fig. 1. Website Priority Determination Tool by Verma et al., (2015).

increase in WWW traffic. For instance, to purchase an item on the web, a customer may explore many websites to have satisfactory E-Commerce transaction which not only provides the high quality branded product but also at best possible discounted price or maximum wallet cash back. Hence, as a result, many of the online shopping portals getting big data on daily basis like Amazon or PayTm Mall android based E-Commerce portal, which handles around a million customer transaction logs on a regular basis, resulting into many TB of data generated on a daily basis. This excessive online generated data is commonly known as 'Big Data' with emphasis on high values of various popular V's, i.e., Value, Velocity, Variety, Veracity, and Volume. Big data may be defined as a collection of a huge number of data sets, the speed of incoming data before processing, outgoing data after processing and range of data sources are beyond the capabilities of conventional relational databases systems for processing and management. Verma and Singh (2017a,b) proposed that 'Big Data' consist of many useful patterns in the form of association rules which are never extracted and hence advanced big data analytics is required to explore these hidden patterns. These patterns are helpful for E-Commerce websites. E-Tailer may utilize such patterns for market basket analysis and hence to increase sales by extracting customer's favorite purchase patterns, efficient and easy inventory management to avoid situations like overstock or out of stock by identifying significant purchase trends for a specific product from various sources such as social media trend analysis. An online merchandise seller may use a big data analytics tool to analyze multiple posts on social media like Instagram, Facebook. The images of popular celebrities that are most shared / most liked recently to determine the latest fashion of dress material and hence can order more stock for similar dresses to quickly satisfy the increased demand of the market.

Market basket analysis using big data analytics for ranking of E-Commerce websites can be easily accomplished by using -RV-Map-Reduce framework, which is robust and scalable and is an open source platform for efficient processing of E-Commerce based big data. Hadoop cluster is characterized by some parallel machines that can easily store and process big data sets, a significant number of clients may easily submit their processes to distributed Hadoop cluster from different locations. Map-Reduce is a simplified programming model that can be used to process big data in Hadoop cluster with the help of primary functions known as Map and Reduce to process big data in (Key, Value) pair format. Hadoop and Map - Reduce based cloud computing framework

may be used for efficient deployment of big data based advanced E-Commerce website ranking system.

The overall objective of this research work is to assist the customer in easy searching and correctly ranking E-Commerce websites to buy genuinely priced authentic products as well as to help E-Tailers in optimizing the structure of their websites to take advantage over competitors.

2. Literature review

Advance adaptive E-Commerce search is a personalized search for retrieval and ranking of relevant E-commerce websites by using intelligent technologies like semantic web, neural networks. The personalized search mechanism requires big data analytics to retrieve useful association rules from data in text, images or videos format as available on social media and purchase history of various customers to retrieve customer specific E-Commerce website ranking patterns efficiently. There are different types of traditional personalized search systems as discussed in the literature.

2.1. Review of hyperlink based adaptive search methods

In general, E-Commerce applications employ hyperlink personalization to assist the client by recommending E-Commerce websites that are more relevant as determined by feedback obtained through his/her buying history and explicit or implicit ratings. It is assumed that consumers who gave similar ratings to related products have similar preferences and accordingly algorithm recommend various website links to the users that are most popular in the similar category as determined by previous customers. E-Commerce portals/websites like Paytm Mall, Mynta uses hyperlink personalization to aid their clients in searching, ranking and purchasing appropriate products. Aoki et al., (2015) discussed the architecture of Web index (WIX) system for hyperlink generation that can be used to insert links to web pages by replacement of keywords as per customer's choice. This, in turn, will reduce his/her load to go through all the web links produced in search engine result. However, if multiple web links can be associated with a keyword, then relevance computation takes time which is the major limitation of the proposed system. Alam and Sadaf (2015) discussed that fetching the significant information from WWW is moderately difficult. The modern search engines may return quite

a huge number of web pages in response to user's query, and the result becomes unmanageable and irrelevant if the query is erroneous as general purpose search engines retrieve documents consequent to all probable meanings of a query. They discussed heuristic search mechanism to extract a group of the pages to assist a user in locating her / his required information effortlessly. They worked out significant cluster label from the title of various documents sharing similar hyperlinks by using Apriori algorithm. However, the usefulness of proposed method using only title information is not guaranteed on heterogeneous data sets. Verma et al., (2015) developed SNEC page ranking algorithm based on various intelligent technologies like artificial neural networks and semantic web. In this paper, we had discussed website priority tool for easy evaluation of E-Commerce search queries and to obtain the relevant ranking of E-Commerce websites. The proposed tool may be used to get E-Commerce website correct ranking concerning its competitor websites efficiently. However, as discussed in future work of this paper, we are going to incorporate various capabilities in our presently proposed IMSS- AE tool such as page loading speed, image-based search, security comparison to rank the E-Commerce websites as required by modern day customer. Hence, proposed algorithm and tool in present research work is an improvement of previously published SNEC algorithm and website priority determination tool.

2.2. Review of search methods based on content personalization

Content personalization on WWW refers to the process of showing different content to different customers on same portal/ website. Sugiyama et al., (2004) explained that sites like Yahoo present the relevant information to users in which they are probably more interested. Users/customers may specify the tabs of his/her choice on such websites that may include Bollywood/Hollywood movies, news, fashion updates, forecasting- sun sign/weather. Users may build their favorite page outline as per their requirement on content personalized portals. However, such systems usually suffer from various limitations like constant effort from the user is requisite as such systems are directly reliant on user inputs. Moreover, these portals cannot automatically adapt to changing needs of the user unless he/she explicitly modify his previously registered preferences. Kuppusamy and Aghila (2014) proposed all-purpose CaSePer, an adaptive website change detection architecture to help the users who frequently browse a specific website and are concerned in knowing the most recent changes rather than considering the complete content of the websites on repeated visits. This model requires being adapted as a custom-made personal search system. Moreover, the experimental efficiency of such a search system is required to be evaluated.

2.3. Review of search methods based on recommender system

In this current age of big data, there is an emerged need of recommender system to deal with information explosion on the web. Wasid and Kant (2015) discussed that recommender systems might help users by suggesting entertainment material like games, shopping deals to make efficient use of their usual search time on the web. They suggested a technique known as particle swarm optimization to determine priorities of various users and accordingly to present recommendations personalized for a specific user. They also suggested different filtering techniques usable by Recommender System, i.e., demographic filtering, collaborative filtering, content-based filtering and hybrid filtering techniques for web-based personalization. Adamopoulos (2014) proposed Probabilistic Neighborhood approach to conquer the regular problems of development of K nearest neighbor's method. They discussed the concept of unexpectedness in popular recommender systems for

easily satisfying the requirements of the user. CACHEDA et al., (2011) suggested an efficient method for collaborative filtering based on differences between customers and products rather than based on their similarities. They suggested latest metrics, GPIM and GIM to calculate the accuracy of prediction for web personalization and unwanted biased prediction of recommendation system. They carried out a detailed comparison between various collaborative filtering algorithms to differentiate their strengths and weakness in diverse conditions. Guy et al., (2010), suggested that recommender system may be merged into search engines for implementation of personalized search. They also discussed that user experience is more important than the performance of recommender systems. Jung et al., (2004), discussed a prototype SERF developed for a university library. This system learns from the user regarding document relevancy corresponding to a search query. It motivates the customer to enter meaningful and non-ambiguous queries and then asks for explicit ratings of search results to measure the level up to which could system satisfy requirements of the user. However, the success of the proposed system depends on the fact that how easily it can compel the user to provide the ratings. Hence, extensive research is required for recommender systems utilization as a personalized search system.

2.4. Review of search methods based on contextual relevance feedback

The contextual systems use user's implicit and explicit data to develop a contextual knowledge base through gathering different customer contextual profiles. Limbu et al., (2006) suggested modification/expansion of queries to appropriately reveal the user's interest and hence to obtain contextually personalized search results. The proposed approach efficiently improves various search criteria like recall and precision by expanding the incomplete/ambiguous query appropriately using thesaurus approach and by adding meta keywords to search query respectively. Tanpaisankit et al., (2012) suggested a personalized search model, the Query in Context (QIC) which improves search query by including user preferences and hence ranking search results with context enrichment to cut down the number of contextually inaccurate search results. The proposed model can be implemented by allowing search query terms with multiple meanings to get weighted towards correct contexts. Vinay et al., (2005) compared three different types of contextual relevance based feedback algorithms by employing target testing procedure and experimentally established that the Bayesian algorithm is more efficient than RSJ and Rocchio algorithms. They also explored that modern search engines do not provide the option for Relevance Feedback and hence users are often dissatisfied with the returned results and are required to modify their query to obtain relevant results manually.

2.5. Review of intelligent technologies based search methods

Singh and Vélez (2014) discussed the model of a search engine Simha to competently search over different cloud platforms for unstructured and structured data using backend elastic search engine. They also reviewed the significance of cautiously designed processes such as Extraction, Transform and Load while indexing big data. Malhotra (2014) explored that huge size of web and SEO interference leads to difficulty in retrieving valuable information from the internet through search engines. However, an artificial neural network can be efficiently trained to provide better search results by implementing supervised learning. Zhang et al., (2012) discussed cloud-based semantic++ search framework to provide results from social networks. They explored the failure of general purpose search engines to establish the relationships between objects, people and web pages by various social networking portals such as Facebook, Instagram, Twitter. Wang et al., (2011)

proposed a methodology for search engine optimization based on customer feedback which may be implicit or explicit and artificial neural networks and hence their use in the implementation of unbiased website ranking model.

3. Motivation

The vast repository of data on the web may be termed as big data. In the present situation, it sometimes becomes quite difficult for a customer to search relevant E-Commerce website on the Internet easily. One of the commonly followed temporary measures is to use a popular search engine like Google. However, as discussed in the literature, none of the search engines can completely solve retrieval problem as no search engine can index entire information available on the web. Bo and Yang-Mei (2014) discussed that most of the conventional search engines suffer from various limitations such as incomplete indexing, low precision, SEO manipulated page rank, low recall. Moreover, a conventional search engine presents the same output consequent to the same query, despite current requirements or personalized preferences of customer submitting the query as discussed by Rasekh (2015). This approach is not suitable for customers with a different set of requirements. Let us take an example, a female or male customer searching for "Online purchase of Belt" on a conventional search engine. The customer will get the same rank of various listed web links in output without any consideration to the fact that one of the customers usually make queries for products meant for ladies and another one for males. Hence, ideally, the search query should be intermediately expanded to "Online purchase of men belts" or "Online purchase of women belts" to make the output more personalized and relevant to E-Commerce customer. A few of the modern search engines provide an option for personalized search. However, they usually fail to adapt to continuously changing needs of the customer as discussed by Wang and Wong (2014). Moreover, users are frequently required to modify their E-Commerce search query number of times to retrieve relevant web links in correct order of ranking as discussed by Verma et al., (2015).

Metasearch engines can address partial indexing problem of conventional search engines to a modest extent. They are built on the top of some search engines, and they search for a query on all of supporting search engines followed by integration and ranking of output links retrieved from each of the search engines to display the result and hence improving recall and precision. However, metasearch engine approach has its own set of limitations. The usual number of web links returned in output for E-Commerce query by each of the supporting search engines is out-sized. Youssif et al., (2011) discussed as if the search query is ambiguous, output links in result becomes even more massive as traditional search engines try to retrieve web links corresponding to all probable meanings of a query hence, integrating and correctly ranking vast number of E-Commerce websites require enormous efforts. Moreover, E-Commerce website ranking using conventional data mining techniques is not efficient as discussed by Verma and Singh (2017a,b) and require to deal with many problems like:

- The credibility of high-ranking E-Commerce websites in search engines output appeared to have declined as a customer is not usually able to find the suitable and genuine product at reasonable price. For example, some of the E-Commerce websites are selling goods without acquiring preauthorization from the manufacturer of the product at unreasonable prices leading to various difficulties for the customer while applying for guarantee/warranty services from the manufacturer. Moreover, E-Tailer

also finds it complicated to structure their E-Commerce websites appropriately to survive in this modern age of intense competition.

- Conventional website ranking systems do not focus on essential features as required by big data management systems. These features include partial failure support, infrastructure and application scalability, component recovery, data recoverability and ability to respond in real time as needed by modern metasearch systems or search engines to search in today's age of big data as discussed by Tsai et al.,(2015).
- Traditional search engines usually perform semantic less page ranking process regarding frequency count of keywords, the proximity between candidate website and E-Commerce query. The queries which can be interpreted in various contexts are likely to produce unexpected results, and user ends up either with lot many website links and sometimes not even a single link in the output.

The proposed research work focuses on addressing above mentioned problems as faced by various stakeholders viz. E-Tailers, End Users, and Search engine developers. The research problem can be summarized to develop a personalized metasearch engine for the benefit of all stakeholders. Moreover, the proposed approach will overcome the restrictions of traditional data mining approaches to extract useful E-Commerce web links from big databases of various search engines by providing essential features of the second generation big data systems like partial failure support, scalability, real-time response.

4. Comparison of platforms for big data analytics

To appropriately choose deployment framework for a web search and ranking application, we need to compare various aspects such as capabilities for partial failure support, fault tolerance, scaling, real-time processing and efficiency in iterative execution. Here, we compared various existing deployment paradigms in Sections 4.1, 4.2 and 4.3 to explain some of the characteristics of different cloud-based platforms useful for deployment of E-Commerce website search and ranking system.

4.1. Types of deployment platforms

Various existing cloud-based deployment platforms are explained as follows (Khurana 2014; Malhotra et al., 2017a,b)

- In one of a kind, cluster utilizes blob storage space as a primary storage space such as Azure blob store, S3. Here temporary clusters are implemented, and they exist only till the period of workflow execution. Blob store act as a source and destination of the workflow. Here, virtual machines may be considered as task execution containers.
- In another type, first generation HDFS (Hadoop Distributed File System) is used as a primary storage space. In contrast, here, persistent clusters are used for long-term storage. Moreover, virtual machines are persistent, and they can perform execution as well as data storage. This type may even use blob storage for cyclic backups and to give data to HDFS. This kind of cloud deployment platform is useful for workloads of type SLA batch workloads, Ad Hoc Interactive and Ad Hoc Batch. For instance, interactive SLA workloads are usually deployed on HDFS due to virtual machines requirement as servers and blob storage requirement as a backup.

4.2. Second generation HDFS

With the recent technological shifts, second generation big data processing systems need to support multiple analytic methods on varied data types, and the ability to respond in real time. Malhotra and Rishi (2017) discussed the essential characteristics of first-generation HDFS like partial failure support, scalability through data streaming and global memory scheduling is also required to be continued by second-generation HDFS as shown in Fig. 2.

There are two significant trends of Second Generation HDFS based Big data search and ranking systems (Gebara et al., 2015; Malhotra and Rishi, 2016)

- There is rapid growth in network bandwidth as compared to hard drive bandwidth.
- Development of In-Memory computation models such as Spark allow intermediate results to be kept in memory and hence reduces overhead of iterative analytics

Second Generation HDFS is adapted as a long-term store from where web applications read their initial data and write back their final results. The data layer is subdivided into various segments for steady storage and provides storage for intermediate objects separately. However, one of the limitations of HDFS lies in running iterative algorithms efficiently. Map function requires to read data at the start of iteration and to write back the results to the disk at the end of the iteration. This frequent access to disk in writing and reading data is responsible for performance and efficiency degradation as discussed by Singh and Reddy (2015).

4.3. Ranking comparison of existing and proposed deployment platforms

Table 1 shows a ranking comparison of various possible big data deployment frameworks on different characteristics such as scaling, fault tolerance. Here Rank -1 shows the best option and Rank - 5 for worst option among all of the listed platforms. It may be noted that this ranking table provides a general idea regarding strengths and weakness of various platforms and it mainly depends on the specific application/purpose. In general, big data applications, there is a tradeoff between Scaling and Real-Time Processing capabilities.

For example, in web search applications, indexing process requires a highly scalable platform to handle billion of web pages returned by some supporting search engines. This indexing accom-

plished via HDFS and Spark are the optimal choices for web search applications as discussed by Shou et al. (2014), and hence these are preferred and proposed deployment frameworks for E-Commerce website search and ranking applications. The detailed ranking comparison between various deployment platforms is shown in Table 1. In the implementation of our proposed IMSS-AE tool, we have chosen HDFS platform due to its high scaling and fault tolerance rank which are two most important requisites in an E-Commerce environment. We have given preference to HDFS over SPARK platform due to easy availability and adaptability of hardware and software related infrastructural requirements for HDFS- Map reduce environment and hence to improve the probability of increased usage and popularity among retailers.

5. System Design

This proposed research work addresses above mentioned E-Commerce website search and ranking problem as discussed in Section 3 using Intelligent Technologies based Personalized Big data Analytics. The simplified modular block diagram of the system is shown in Fig. 3.

5.1. Phase 1: Query preprocessing using semantic analysis

The proposed E-Commerce website ranking system can easily keep track of customer preferences, i.e., short term and long term preferences by building customer's profile. This system can closely monitor customer's browsing history, and the system will automatically update customer's profile with a change in his/her browsing patterns of websites without requiring any additional effort from the customer. Here long-term preferences can be retrieved using customer's past browsing history and registered preferences while short-term preferences will be retrieved using browsing history of last two days only. This phase can extract search queries and visited web links from browsing history by fetching meta keywords and hence by developing customer's profile which can be further used to establish customer's contextual database. These Meta keywords can be utilized for selecting concepts with the ontology-based database. These Meta keywords through selected ideas will be used to disambiguate the search query and hence to expand a simple keyword query into more meaningful customer personalized query to improve the search results through backend search engines as discussed by Malhotra and Rishi (2017). The Semantic Relevancy Vector (SRV) is determined by using Longest Common Subsequence (LCS) to determine

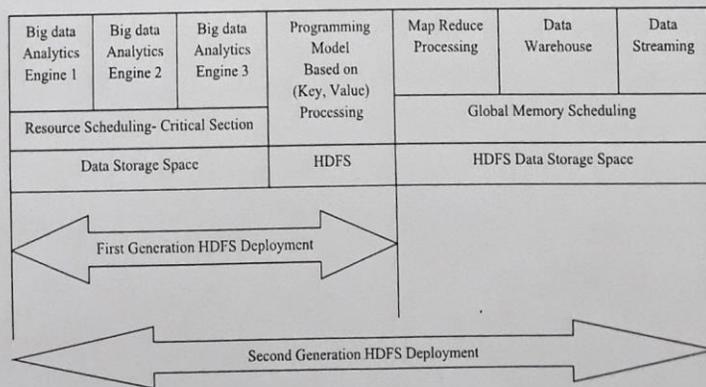


Fig. 2. Second Generation HDFS V/S First Generation HDFS (Malhotra and Rishi, 2017).

Table 1
Ranking Comparison of Existing and Proposed Deployment Platforms.

Platform	Scaling Rank (Type)	Fault Tolerance Rank	Real-Time Processing Rank	Iterative Tasks Rank
HDFS	1 (Horizontal)	1	4	4
SPARK	1 (Horizontal)	1	4	3
PEER TO PEER	1 (Horizontal)	5	5	4
HPC CLUSTERS	3 (Vertical)	2	3	2
MULTICORE	4 (Vertical)	2	3	2
GPU	4 (Vertical)	2	1	2
FPGA	5 (Vertical)	2	1	2

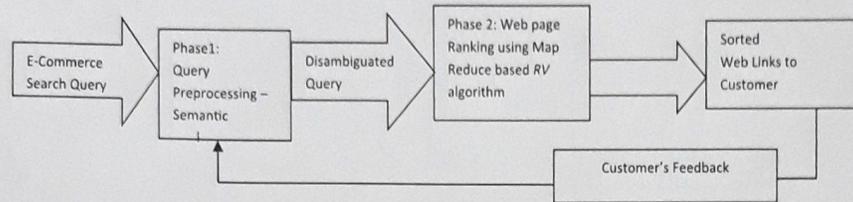


Fig. 3. Simplified system Design.

the proximity of web page and contextual similarity concerning customer requirement. The detailed conceptual flow of this step is shown in Fig. 4.

5.2. Phase 2: Website ranking using map reduce based RV page ranking algorithm

This research paper uses Hadoop-RV- Map Reduce based big data mining and analytics framework to simplify the E-Commerce website personalized search and ranking process through the implementation of *Intelligent Meta Search System for Advanced E-Commerce*. IMSS-AE tool is built on the top of some other popular search directories like Yahoo, Meta search engines like Dogpile and search engines like Google. This proposed research work is implemented at middle layer of public cloud

for service level agreement. This phase accepts preprocessed disambiguated query as generated in the last step. In this step, we will first search for user-specified query on each of the back end search engines and will assign a unique id to each of the retrieved clusters of web pages from 1 to n. These clusters are then compared with user specifications such as privacy/security, response time and ease of accessibility to find relevant cluster list L which should be further processed for ranking purpose. Short listing of clusters can be performed by performing a parametric match. The very first criterion is to determine accessibility which may be public, private or community type cloud. The second criterion is related to security which can be determined by https: transmission capability or SSL availability followed by the third criterion of response time which should be less than that of customer specified value. The first stage of ranking will be

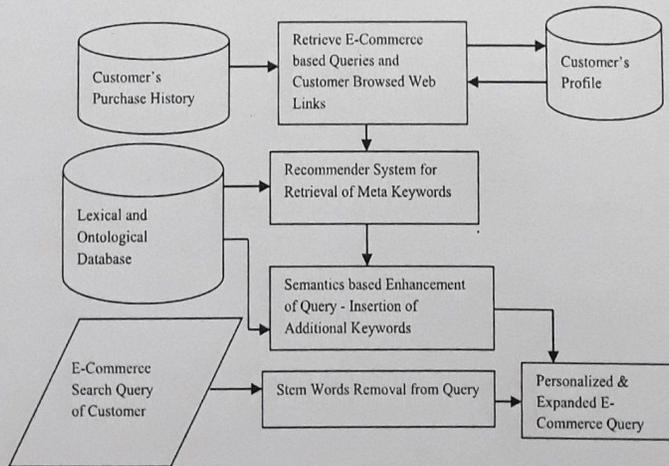


Fig. 4. System Design- Query Preprocessing.

implemented by determination of content relevancy in two-phase programming model called Map and Reduce to support HDFS based cloud framework. Map and Reduce code used in the proposed algorithm is as follows:

```
Map (SEngine_ID: Integer, Web_Log: String)/Web Log Cluster
processing
{
  List <String> TL = Tokenize (Web_Log) // TL- Token List
  While (Web_Token in TL)
  {
    Insert ((String) KL, (Integer) 1) // KL- Keyword List
  }
}
Reduce (KL: String, count: List <Integer>)
{
  Integer Freq = 0
  While(KL)
  {
    Freq = Freq + 1
  }
  Insert ((String) Web_Token, (Integer) Freq)
}
```

Here, Map method will accept a key as search engine ID for each retrieved web links cluster from various background search engines and the second argument is weblog to tokenize each of the entry of link entry in the weblog for counting frequency of each of the keyword in E-Commerce search query. Insert () method is used to generate elements in the list by inserting numeric one corresponding to each occurrence of a keyword as we token. However, Reduce method is implemented to cumulate over all the occurrence of each keyword. This is accomplished by the insertion of numeric 1 (one) to determine the frequency of the keyword in each of the web document and hence to conclude the content relevancy vector of retrieved web documents from various search engines. The second stage of ranking concludes the time relevancy vector (TRV) for each web page using its last time of update on the web as well as by considering previous customer time spend statistic with similar E-Commerce search query. The third stage of ranking includes feedback relevancy vector (FRV) which may include explicit and implicit feedback of past customer. Some of the previous research results show that explicit feedback of a product / E-Commerce website in the form of online reviews can significantly impact the purchase decision of a customer. Liu et al., (2017) discussed that it is quite difficult for a customer to review a large number of online reviews easily. Hence, there is an urgent need to develop a method to rank E-Commerce websites based on sentiment analysis. Online reviews usually expressed in sentences and hence dictionary based semantic analysis is used in this research work to determine neutral, negative or positive reviews. The Semantic Relevancy Vector (SRV) is already determined in step1. At last, all of these vectors with their weighted contribution as mentioned by customer helps in determination of rank for each of the cluster of web pages as discussed in proposed RV page ranking algorithm. The various evaluation metrics involved in the determination of rank of an E-Commerce website are Semantic Relevancy Vector (SRV), Feedback Relevancy Vector (FRV), Content Relevancy Vector (CRV), Privacy Vector (PV) and Accessibility Vector (AV). The step by step calculation of various evaluation metrics and their role in the determination of overall rank and personalized search precision of an E-Commerce website along with system design of phase 2 is shown in Fig. 5.

5.3. Relevancy Vector (RV) page ranking algorithm

Relevancy Vector, the page ranking algorithm is an extended algorithm of earlier published CPR algorithm by Malhotra et al. (2017a,b). RV algorithm is an improvement over CPR algorithm due to two main reasons (i) RV algorithm is designed to take benefit of cloud technology (ii) RV algorithm unlike CPR algorithm is specially tailored for E-Commerce website ranking. RV algorithm is discussed in detail as follows:

- Start
- Accept E-Commerce query from a customer.
- Personalize search query using customer profile database and semantic enhancement.
- Split the query into various keywords W_1, W_2, \dots, W_n and remove stem words from the query.
- Determine minimum and maximum length of each of the keyword as follows

```
Set min = strlen(W1), max = strlen(W1)
Set c = 2
While (c < n) do
  If MIN > Wc then
    MIN = strlen(Wc)
EndIf
If MAX < Wc then
  MAX = strlen(Wc)
EndIf
EndWhile
```

- Execute E-Commerce query on various backend search engines and assign ID to retrieved websites can easily
- Determine customer navigation session. This process can be accomplished by comparing customer's query with each of the past E-Commerce, and other search queries present in customer profile database using LCS. The LCS, i.e., Longest Common Subsequence is used to determine proximity between website and customer preferences and store the same in SRV[ID] to represent the semantic rank of particular E-Commerce website identified by ID.
- Calculate timestamp T_s of creation and average time spent by past customer T_p to calculate Time relevance vector $TRV[ID] = (T_s + T_p)/2$

// Calculation of CRV [ID]

- For $x = 1$ to n do // n refer to total number of websites
 - o Calculate frequency of each keyword using web dictionary
 - o Eliminate all those websites with frequency of found keywords less than not found keywords
 - o Call **Map**(WebPage_ID, WebPage_Content)
 - o Call **Reduce**(Web_Link, Count)
 - o Calculate average frequency of the frequency of individual keywords
 - o Store average frequency in CRV[ID]

EndFor

- For $x = 1$ to r do // r refer to remaining websites after elimination in last step
 - Calculate Privacy vector, $PV[ID] = 0$; If(linkprivacy = privacy(wbsite(ID))) then set $PV[ID] = 1$

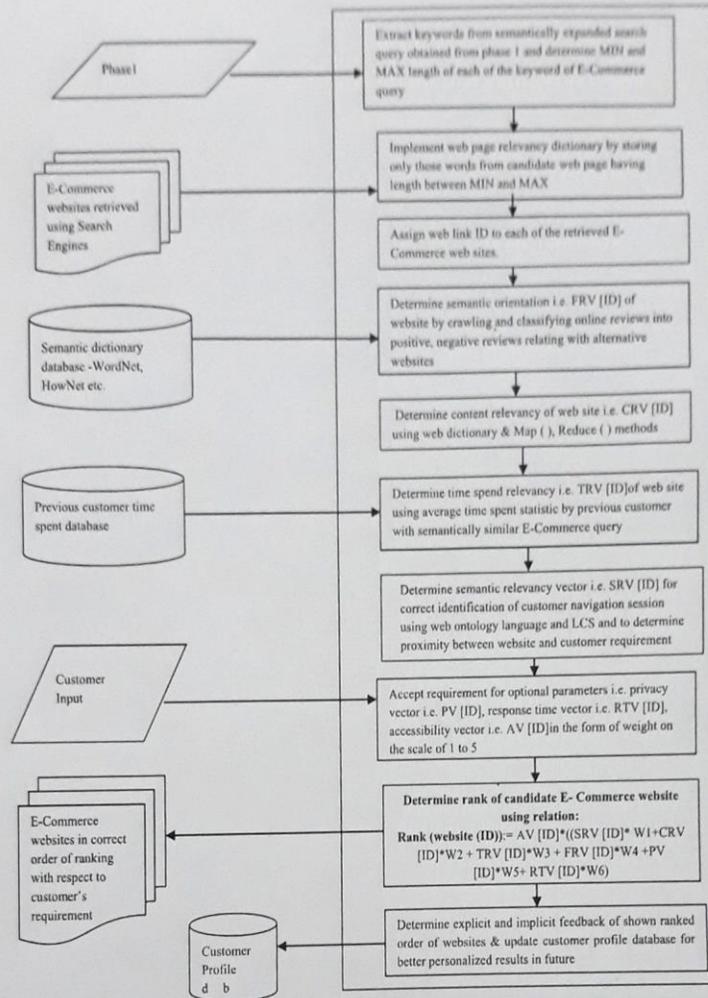


Fig. 5. Web Site Ranking using Map Reduce based RV page ranking Algorithm.

- Calculate Accessibility Vector, $AV[ID] = 0$; If (Cloud = Public) then set $AV[ID] = 1$
- Calculate Reply Time Vector, Set $RTV[ID] = 0$
- If (linkresponse > ReplyTime(website[ID])) then

```

RTV [ID] = strresponse - ReplyTime(website[ID])
EndFor

```

- Eliminate all E-Commerce websites with either $RTV[ID] = 0$, $PV [ID] = 0$ or $AV[ID] = 0$
- Determine Feedback Relevancy Vector i.e. $FRV[ID]$ using semantic dictionaries to analyze online reviews and categorize them into negative, positive and neutral reviews and calculate FRV as follows:

```

Set Count = 0
If (Review is Positive) then // Well Satisfied Experience of Past Customer
Count = Count + 2
Else If (Review is Negative) then // Unsatisfied Experience of Past Customer
Count = Count - 2
Else If (Review is Neutral) // Hesitant or Confused Past Customer
Count = Count - 1
EndIF
Set FRV [ID] = Count

```

- Calculate Rank (website (ID))= AV [ID]+(SRV [ID]*W1) + CRV [ID]*W2 + TRV [ID]*W3 +
- FRV [ID]*W4 + PV [ID]*W5 + RTV [ID]*W6)
- Accept feedback from the customer about the shown ranking order and update customer profile database.

The RV page ranking algorithm determines the relevancy of an E-Commerce website for a specific customer using the calculation of various relevancy vectors such as Content Relevancy Vector, Semantic Relevancy Vector, Reply Time Vector, Feedback relevancy Vector, Privacy Vector. The algorithm starts with personalized expansion of search query as discussed in Section 5.1. After removal of stem words viz. a, the, an from the query. The RV algorithm will calculate the minimum and maximum length of each of the keyword of the search string. The SRV is determined using Longest Common Subsequence. The CRV is determined using Map and Reduce functions. Moreover, the algorithm will remove all those E-Commerce websites from final output with Reply Time Vector = 0, Accessibility Vector = 0 or Privacy Vector = 0. The previous step is further followed by calculation of Feedback Relevancy Vector depending on the experience of past customer. In the last, rank of a website is calculated by weighted summation of various relevancy vectors.

5.4. Intelligent meta search system for advanced E-Commerce - IMSS-AE tool

IMSS-AE tool using second generation HDFS, Map-Reduce framework for big data analytics is implemented in the ASP.NET framework to assist the customer while performing E-Commerce transaction. This tool is also used to determine the performance of RV page ranking algorithm. The interface of IMSS-AE tool is shown below in the Fig. 6. After registration/Sign In/Sign Up, the interface of the tool on authentication will allow the customer to select few or all of the mentioned metasearch engine/search engine/search directory, i.e., Dogpile, Yahoo, and Google respectively for background retrieval of E-Commerce websites. Here, IMSS-AE tool will act like metasearch engine; the customer can specify search string in the search box on the interface of IMSS-AE tool. The tool will first expand the search query to more meaningful personalized search query. This tool will further assign the rank to some of the top web links retrieved from back-end search engines based on the calculation of various ranking vectors such as AV, FRV, SRV, CRV, RTV, TRV with appropriate weight age as determined from customer specified parameters. The detailed discussion about the calculation of ranking vectors and weighted contribution is discussed in Section 5.2. The tool will output

Intelligent Meta Search System- Advanced E-Commerce			
SIGN UP/ New Customer	User ID: DM@UOK	Password: *****	
Personalized Search		Advanced Criteria Search	
YAHOO	GOOGLE	DOGPILE	
Page Loading Speed	Transaction Security	Response Time	
Enter Search String: online belt purchase			
SEARCH		RESET	
FAST FORWARD>>			
Personalized Expanded Search String by IMSS-AE: online belt purchase for women			
SUGGEST ANOTHER		CANCEL	
CONTINUE			
Ranking Box.....			
Rank	Web Link	Response Time	Feedback
1	www.amazon.in/clothing/women	00:00:00:15ms	YES <input type="radio"/> NO <input type="radio"/>
2	www.myntra.com/women-belts	00:00:00:36ms	YES <input type="radio"/> NO <input type="radio"/>
3	m.jabong.com/women/accessories	00:00:00:49ms	YES <input type="radio"/> NO <input type="radio"/>

Fig. 6. Interface of IMSS-AE Tool.

Please cite this article in press as: Malhotra, D., Rishi, O.P. An intelligent approach to design of E-Commerce metasearch and ranking system using next-generation big data analytics. Journal of King Saud University - Computer and Information Sciences (2018), <https://doi.org/10.1016/j.jksuci.2018.02.035>

E-Commerce web links in the sorted order of their ranking along with various statistics as selected by the customer in advanced search criterion such as page loading speed, response time, transaction security as well as background search engine. However, Personalized Search tab will not allow choosing any of the search criteria and will give result directly by referring customer registered past preferences. This tool will suggest personalized expanded search string by using browsing history. Moreover, in output, the tool will rank the various links using customer's preferred search criteria along with details of selected statistics, the tool will also allow the customer to provide feedback about the ranked order of web links and hence to improve its personalized ranking capabilities to better match with changing customer preferences.

6. Experimental and graphical analysis

The personalized relevancy of an E-Commerce website to a specific customer for a given product query depends upon its position in the output of search results. To compare the IMSS- SE Tool with other popular search tools, Precision of search at X metric is considered, which is here shown by P (X). Various search tools used for comparison in this study are metasearch engine, search engine and search directory, i.e., Dogpile, Google, Yahoo and IMSS-SE Tool by Malhotra et al.,(2017a,b). For a given E-Commerce query, P (X) reports how many fractions of output links in the result, labeled as significant are presented in the top X results. Here, it is assumed that a web link ranked upper is more relevant for the customer. The tool rank is then compared with human volunteer's judgment to verify the relevancy reported by the tool as well as professional search engine/tool, and at last the difference in precision of IMSS-AE tool and professional search tool is plotted.

To evaluate the efficiency and effectiveness of proposed RV algorithm and IMSS- AE tool. Here, we employed 20 human volunteers in various age groups from 15 years to 45 years, and with a minimum of 3 years' experience of carrying out numerous E-Commerce transactions, nine of them are males, and eleven of them are females. They were asked to use personal laptops with installed IMSS-AE tool followed by sign up/registration process on the tool, we asked volunteers to repeat the following steps for at least five trial runs on each of Dogpile, Yahoo, Google and proposed IMSS- AE Tool:

1. Initially, we asked volunteers to search for intentional incomplete E-Commerce query, for instance, a query like Samsung online purchase rather than Samsung mobile online purchase
2. In next step, volunteers were asked to rank output links from 1 (worst) to 5(best) individually to all of the considered engines and proposed IMSS- AE tool. The basis of ranking is various precision parameters such as personalized relevancy, page update time and response time to the top 10 web links in the output.
3. After gathering ranked data from each of the volunteers, normalization of various precision parameters is carried out by using the following expression:

$$NP_{ab} = (MAX (PP_{ab}) - PP_{abr}) / (MAX (PP_{ab}) - MIN (PP_{ab}))$$

Where, PP_{ab} = Value of b_{th} precision parameter of metasearch Webpage; NP_{ab} = Normalized value of b_{th} precision parameter of a_{th} webpage; MIN, MAX = Minimum and Maximum value of each of the precision parameter.

1. After that, we calculated the overall weighted precision of each E-Commerce web link retrieved by the volunteer as $N_a = \sum W_b \cdot NP_{ab}$. Where, N_a = weighted precision of a_{th} webpage; W_b = Weight assigned to b_{th} parameter by customer, where $0 < W_b < 1$

2. At last, the overall precision of search engine/tool is determined by calculating the average of all the weighted precisions as gathered by volunteers for a given parameter among Response time; Page updated content, Personalized Relevancy at a time, $Precision(ID) = AVERAGE (N_a)$.

The graph is shown in Figs. 7, 8 and 9 demonstrate the average precision metric comparison between proposed IMSS-AE Tool with a popular search directory, i.e., Yahoo as calculated by volunteers by following above steps for various precision parameters, i.e., Response Time, Page Freshness and Personalized Relevancy respectively. The graphs shown indicate that initial average precision of IMSS-AE is lesser than Yahoo for page freshness, response time and personalized relevancy. However, soon after few trial runs, the precision of tool improves in comparison to Yahoo. This improved precision demonstrates the semantic-based learning capabilities of IMSS-AE. This tool can build customer profile database by monitoring his/her personalized browsing preferences with some trial runs and hence tool will be able to calculate various important relevancy vectors as discussed in Section 5.2 such as SRV, FRV, TRV more accurately. However, precision improvement in Fig. 7 for response time is not as significant as in Figs. 8 and 9, i.e. for page freshness and personalized relevancy. This difference in precision statistics is because of background implementation of Map-Reduce based second generation HDFS used in the tool. This lagging is due to time delay occurring on account of iterative analytics. This suspension can be improved further by use of Spark based HDFS system as in-memory computation models implemented through Spark allow intermediate results to be kept in memory and hence reduces the overhead of iterative analytics as

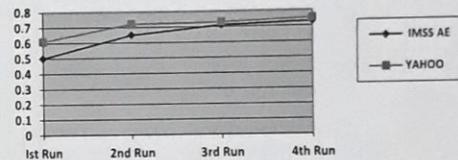


Fig. 7. Precision Comparison between IMSS-AE and YAHOO- Response Time.

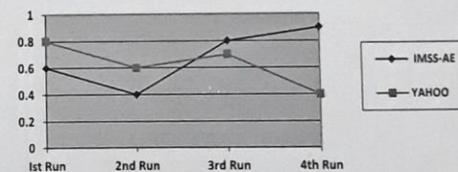


Fig. 8. Precision Comparison between IMSS-AE and YAHOO - Page Freshness.

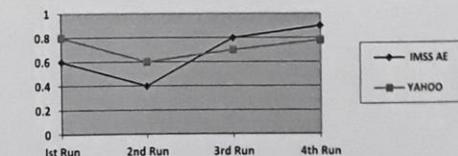


Fig. 9. Precision Comparison between IMSS-AE and YAHOO- Personalized Relevancy.

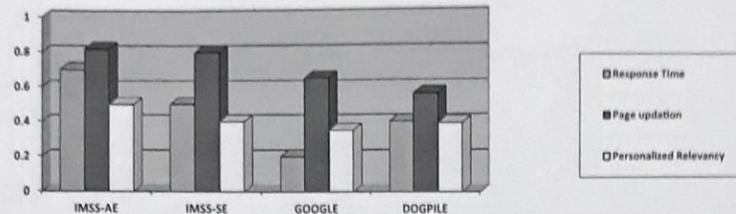


Fig. 10. Search Precision Comparison between IMSS-AE with Dogpile, IMSS- SE, and Google.

discussed by Malhotra and Rishi (2017). Similarly, we carried out an extensive experimental comparison between proposed IMSS-AE tool with Google, Dogpile and IMSS-SE regarding various precision parameters, i.e., Response Time, Page Updation and Personalized Relevancy. The extensive experimental evaluation discussed and its graphical demonstration in Figs. 7, 8, 9, and 10 indicates the improvement in various precision parameters at much faster pace when a personalized search is accomplished with proposed IMSS-AE over other professional and popular search engines, i.e., Google, Yahoo and Meta search engines, i.e., Dogpile, IMSS-SE.

7. Conclusion and future work

This research paper presents a Hadoop- Map Reduce based personalized E-Commerce search framework for the second generation big data analytics. The research gap is shown in this study by submitting various conventional search systems in the form of detailed category wise literature review. This research work proposes a novel RV page ranking algorithm and implements the same as an E-Commerce website ranking tool, i.e., Intelligent Meta Search System for Advanced E-Commerce. The IMSS- AE tool can assist modern day customer in choosing appropriate E-Commerce website for online purchase of a product. The efficiency of proposed ranking approach is justified by experimental analysis. The graphical evaluation for comparison of personalized precision of IMSS-AE tool over Yahoo, Dogpile, Google, and IMSS- SE tool demonstrates the effectiveness of proposed approach over conventional & professional page ranking methods. The practical implications for three different audiences of this research work are as follows:

Practical Implication for End User-The end user of this research work is an online customer willing to make an online transaction. The result of this research work in the form of IMSS-AE tool can assist the customers in the suitable ranking of E-Commerce websites for the purchase of a specific product. The end user will be benefitted by personalized website ranking output and hence can easily select a website that is most appropriate for satisfying the online purchase needs of a user.

Practical Implications for E-Tailers/Retailers: The E-Tailers, i.e., E-Commerce websites or Retailers will be benefitted from this research work as they can use IMSS- AE tool to improve the structure of their websites to satisfy their customers easily and hence to take the lead over the competition.

Practical Implications for Search Engine Developers: This research work can assist the developers in search engine domain to bring out their best in the form of meta search tool. They can take advantage of vast databases of various search engines and can employ Big Data Analytics to fetch out personalized page ranking patterns using an innovative algorithm like proposed RV page ranking algorithm.

In future, RV page ranking algorithm and IMSS-AE tool can further be enhanced to implement market basket analysis through

extraction of association rules from big data stored in online transactional databases. These association rules will assist various stakeholders, i.e., E-Tailers/Retailers for launching various promotional schemes such as Buy One Get One; combo discounted offers, appropriate product suggestions to online customers, target marketing. The retail transactional databases are often quite huge. The traditional data mining approaches to mine useful patterns from such voluminous databases to launch promotional schemes are quite time-consuming and inefficient when compared with Hadoop/Map reduce like big data analytics framework. These promotional schemes from E-Tailers/Retailers will be quite beneficial for another stakeholder, i.e., end user and hence will result in not just increase in sales but also in better satisfaction of end user. Moreover, for the benefit of another stakeholder, i.e., search engine developers, next-generation big data analytics may be incorporated in future editions. These future versions may include (i) SPARK model may be used for reduction in processing overhead. The overhead is on account of iterative analytics and can be reduced by keeping intermediate results in memory To overcome various limitations of conventional HDFS- Map Reduce such as lack of real-time response and dynamic initialization of multiple analytic engines (ii) proportional growth in network bandwidth requirements along with secondary storage needs. The precision of proposed IMSS-AE tool can further be improved by incorporating artificial neural networks for implementation of supervised learning of customer preferences for the better-personalized experience.

References

- Adamopoulos, P., 2014. On discovering non-obvious recommendations: Using unexpectedness and neighborhood selection methods in collaborative filtering systems. In: Proceedings of the 7th ACM international conference on Web search and data mining, ACM, pp. 655–660.
- Alam, M., Sadaf, K., 2015. Labeling of Web Search Result Clusters using Heuristic Search and Frequent Itemset. *Procedia Computer Science*, Elsevier, pp. 216–222.
- Aoki, Y., Koshijima, R., Toyama, M., 2015. Automatic Determination of Hyperlink Destination in Web Index. In: Proceedings of the 19th International Database Engineering & Applications Symposium, ACM, pp. 206–207.
- Bo, C., Yang-Mei, L., 2014. Design and Development of Semantic-Based Search Engine Model. *Intelligent Computation Technology and Automation (ICICTA)*, 2014 7th International Conference, IEEE, pp. 145–148.
- Cacheda, F., Carneiro, V., Fernández, D., Formoso, V., 2011. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Trans. Web (TWEB)* 5 (1). <https://doi.org/10.1145/1921591.1921593>.
- Gebara, F., Holstef, H., Nowka, K., 2015. Second Generation Big data Systems: Cover Feature Outlook. *IEEE Computer Society*, IEEE, pp. 36–41.
- Gomez-Nieto, E., San Roman, F., Pagliosa, P., Casaca, W., Helou, E.S., de Oliveira, M.C.F., Nonato, L.G., 2014. Similarity preserving snippet-based visualization of web search results. *IEEE Trans. Visualization Comput. Graph.* 20 (3), 457–470.
- Guy, I., Jaimes, A., Agulló, P., Moore, P., Nandy, P., Nastar, C., Schinzel, H., 2010. Will recommenders kill search?: Recommender systems-an industry perspective. In: Proceedings of the fourth ACM conference on Recommender systems, ACM, pp. 7–12.
- Jung, S., Harris, K., Webster, J., Herlocker, J.L., 2004. SERF: integrating human recommendations with search. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management, ACM, pp. 571–580.

- Khurana, A., 2014. Bringing big data systems to the cloud. *IEEE Cloud Comput.* 1(3), IEEE, 72–75.
- Kuppusamy, K.S., Aghila, G., 2014. CaSePer: an efficient model for personalized web page change detection based on segmentation. *J. King Saud Univ. Comput. Information Sci.* 26(1), Elsevier, 19–27.
- Limbu, D.K., Connor, A., Pears, R., MacDonell, S., 2006. Contextual relevance feedback in web information retrieval. In: *Proceedings of the 1st International Conference on Information Interaction in Context*, ACM, pp. 138–143.
- Liu, Y., Bi, J.W., Fan, Z.P., 2017. Ranking products through online reviews: a method based on sentiment analysis technique and intuitionistic fuzzy set theory. *Information Fusion* 36, 149–161.
- Malhotra, D., 2014. Intelligent web mining to ameliorate Web Page Rank using Back-Propagation neural network. *Confluence the Next Generation Information Technology Summit (Confluence)*. 2014 5th International Conference, IEEE, pp. 77–81.
- Malhotra, D., Verma, N., 2013. An ingenious pattern matching approach to ameliorate web page rank. *Int. J. Comput. Appl.* 65 (24), 33–39.
- Malhotra, D., Malhotra, M., Rishi, O.P., 2017. An Innovative Approach of Web Page Ranking Using Hadoop- and Map Reduce-Based Cloud Framework. In: *Proceedings of Advances in Intelligent Systems and Computing*, Vol. 654, CSI, Springer, pp. 421–427.
- Malhotra, D., Rishi, O.P., 2016. IMSS-E: An Intelligent Approach to Design of Adaptive Meta Search System for E-Commerce Website Ranking. *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*, ACM. <https://doi.org/10.1145/2979779.2979782>.
- Malhotra, D., Rishi, O.P., 2017. IMSS: A Novel Approach to Design of Adaptive Search System Using Second Generation Big data Analytics. In: *Proceedings of International Conference on Communication and Networks*, Springer, pp. 189–196.
- Malhotra, D., Verma, N., Rishi, O.P., Singh, J., 2017b. Intelligent Big data Analytics: Adaptive E-Commerce Website Ranking Using Apriori Hadoop-BDAS-Based Cloud Framework. Maximizing Business Performance and Efficiency Through Intelligent Systems, IGI Global, pp. 50–72.
- Rasekh, I., 2015. A new competitive intelligence-based strategy for web page search. Elsevier. *Procedia Computer Science*, pp. 450–456.
- Sugiyama, K., Hatano, K., Yoshikawa, M., 2004. Adaptive web search based on user profile constructed without any effort from users. In: *Proceedings of the 13th International Conference on World Wide Web*, ACM, pp. 675–684.
- Tanapatankit, P., Watrous-deVersterre, L., Song, M., 2012. Personalized query expansion in the QIC system. In: *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, ACM, pp. 259–263.
- Shou, L., Bai, H., Chen, K., Chen, G., 2014. Supporting privacy protection in personalized web search. *IEEE Trans. Knowledge Data Eng.* 26(2), IEEE, 453–467.
- Singh, A., Véléz, H.G., 2014. Hierarchical multi-log cloud-based search engine. In: *Complex, Intelligent and Software Intensive Systems (CISIS)*. Eighth International Conference, IEEE, pp. 211–219.
- Singh, D., Reddy, C.K., 2015. A survey on platforms for big data analytics. *J. Big Data* 2 (1), 8. <https://doi.org/10.1186/s40537-014-0008-6>.
- Tsai, C.W., Lai, C.F., Chao, H.C., Vasilakos, A.V., 2015. Big data analytics: a survey. *J. Big Data* 2 (1), 21. <https://doi.org/10.1186/s40537-015-0030-3>.
- Verma, N., Malhotra, D., Malhotra, M., Singh, J., 2015. E-commerce website ranking using semantic web mining and neural computing. *Procedia Computer Science*, Science Direct, Elsevier, pp. 42–51.
- Verma, N., Singh, J., 2017. An intelligent approach to Big Data analytics for sustainable retail environment using Apriori-MapReduce framework. *Ind. Manage. Data Syst.* 117(7), Emerald, 1503–1520.
- Verma, N., Singh, J., 2017. A comprehensive review from sequential association computing to Hadoop MapReduce parallel computing in a retail scenario. *J. Manage. Analytics*, Taylor and Francis. doi:10.1080/23270012.2017.1373261.
- Vinay, V., Wood, K., Milic-Frayling, N., Cox, I.J., 2005. Comparing relevance feedback algorithms for web search. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, ACM, pp. 1042–1053.
- Wang, S., Xu, K., Zhang, Y., Li, F., 2011. Search engine optimization based on algorithm of BP neural networks. In *Computational Intelligence and Security (CIS)*, 2011 Seventh International Conference, IEEE, pp. 390–394.
- Wang, H., Wong, K., 2014. Personalized search: An interactive and iterative approach. In *Services (SERVICES)*, 2014 IEEE World Congress, IEEE, pp. 3–10.
- Wasid, M., and Kant, V., 2015. A particle swarm approach to collaborative filtering based recommender systems through fuzzy features. *Procedia Comput. Sci.* 54, Elsevier, 440–448.
- Youssif, A.A., Ghalwash, A.Z., Amer, E.A., 2011. HSWS: Enhancing efficiency of web search engine via semantic web. In: *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, ACM, pp. 212–219.
- Zhang, G., Li, C., Xing, C., 2012. A Semantic++ Social Search Engine Framework in the Cloud. In *Semantics, Knowledge and Grids (SKG)*, 2012 Eighth International Conference, IEEE, pp. 270–278.



IMSS-P: An intelligent approach to design & development of personalized meta search & page ranking system

Dheeraj Malhotra^{*}, O.P. Rishi

Department of Computer Science and Informatics, University of Kota, Kota, Rajasthan 324 005, India

ARTICLE INFO

Article history:
Received 24 July 2018
Revised 19 October 2018
Accepted 26 November 2018
Available online xxxxx

Keywords:
Meta search tool
Personalized page ranking
ACVPR algorithm
IMSS-P tool
Logistic regression
Big data analytics

ABSTRACT

The proposed research work aims to discuss and explore various constraints of traditional web page search and ranking systems primarily in the present generation of big data. The primary objective is to facilitate a web user by presenting a most personalized web page ranking as a response to a user's search query by considering tastes and browsing history of the user while previously searching on the web. This research intends to design and develop a machine learning based next generation of web page ranking algorithm, i.e., Advanced Cluster Vector Page Ranking algorithm (ACVPR). This ACVPR algorithm is implemented in the form of an Intelligent Meta Search System-Personalized tool to evaluate the performance of the algorithm. The ACVPR algorithm arm the user with a powerful meta-search tool to facilitate the user by providing a web page ranking order to quickly satisfy the personalized needs especially when the search query is erroneous or incomplete. An extensive mathematical and experimental evaluation of the developed logistic regression model by calculating and comparing various evaluation metrics such as specificity, sensitivity, precision, recall using R statistical tool shows the improved efficiency as compared to other popular search engines.

© 2018 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

In the present generation of big data, the searching process is reformed a lot because of the massive growth in web resources. The modern generation user prefers to search for specific information via search engines because of the easy availability of the Internet Service Providers (ISP). The extreme competition among ISPs in a country like India leads to affordable Internet rental and hence unexpected growth in the number of Internet users in a small duration of the last two years is observed. However, searching and listing a relevant website to satisfy the personalized requirements of the user quickly is not easy as web users are mostly reliant on the generic search engines like *Bing*, *Yahoo*, *Google* to list a most relevant website among top three to five links on the first page (Ahmad et al., 2017; Bouadjenek et al., 2016; Chawla,

2018). Moreover, most of the popular search engines are biased and tend to show the paid links at the top of their search results irrespective of their relevance concerning user's query. For instance, Indian antitrust watchdog imposes a fine of 21.17 million USD on Google for the search bias in February 2018. The Competition Commission of India (CCI) found Google indulged in abusing its dominant position and using search bias to harm web user and other competitors. Earlier European Union also imposed a fine of 3 billion USD on Google for biased search output to devalue rival offerings (www.reuters.com).

Gomez-Nieto et al. (2014) highlighted when different users input the same search query, popular and advanced version of the search engine fetches the same links in the result. The advanced search engines return the search result without considering the personalized preferences of the user. Moreover, as discussed by Malhotra and Verma (2013), if the search query is partially incomplete or is ambiguous, then most of the modern search engine tends to return the result by interpreting all possible meanings of the query. For example, if we consider a partial or ambiguous search query "Jurassic World" by two different web users on the Google Search engine in June 2018. The search engine shows top web links of a latest released movie on "Jurassic World: Fallen Kingdom." It may be very much possible that one of the web

^{*} Corresponding author.

E-mail address: dheerajmalhotra4@gmail.com (D. Malhotra).

Peer review under responsibility of King Saud University



Production and hosting by Elsevier

<https://doi.org/10.1016/j.jksuci.2018.11.013>

1319-1578/© 2018 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Please cite this article as: D. Malhotra and O. P. Rishi, IMSS-P: An intelligent approach to design & development of personalized meta search & page ranking system, Journal of King Saud University – Computer and Information Sciences, <https://doi.org/10.1016/j.jksuci.2018.11.013>

users is interested in web pages to read reviews or to buy movie tickets of the latest movie. However, the same is not necessarily applicable for another user, who might be interested in visiting a themed water park with a similar name. This problem may be resolved by an intelligent meta-search system designed and developed using logistic regression based machine learning model and Hadoop2 based advanced big data analytics platform (Malhotra, 2014; Malhotra and Rishi, 2016; Malhotra and Rishi, 2017; Malhotra et al., 2017). The personalized search system will fetch the query from a user and will first modify the search query based on user preferences as mentioned in his/her profile. The personalized preferences can also be retrieved from short-term browsing history, for instance, a user usually booking tickets for themed parks will be shown web pages related to themed water parks consisting of keywords used in his/her search query, i.e., *Jurassic World* on the top of search result and not the links of a movie as discussed above. The personalized search systems are meant not just to search web relevant to a query but also relevant to the individual searching the query (Ferretti et al., 2016; Gupta et al., 2017; Salonen and Karjaluoto, 2016; Zhou et al., 2018; Wang and Tao, 2018).

The massive on stream accumulated data on the web is popularly known as 'Big Data' with more insistence on the Volume of data besides other V's to characterize the data, i.e., Velocity, Variety, Value, and Veracity. Big data is defined as an extensive collection of datasets and sources that are beyond the capabilities of traditional search and page ranking systems to process effectively and efficiently. The detailed ranking comparison of various big data analytics and cloud deployment platforms justify the choice of Hadoop2 as the best analytics platform for the deployment of the proposed intelligent meta-search system. The Hadoop2 advances the capabilities of Hadoop1 by introducing two new modules, i.e., Yarn and HDFS federation. The Yarn module enables segregation of resource management responsibilities from processing engines. The HDFS federation allows the creation of multiple name nodes as compared to a single node in Hadoop1. These advancements help in building a more reliable and robust system architecture for efficient big data analytics as discussed by Verma and Singh (2017a,b).

The machine learning model based on logistic regression may be developed using the R-statistical tool to predetermine the suitability of a web page to satisfy the personalized search needs of a user. The learning model will determine the most suitable page ranking order for a specific web user concerning various parameters such as page loading speed, response time, security while browsing the page and personalized relevancy. The scientific evaluation concerning the calculation of the confusion matrix, specificity, sensitivity, etc. easily verifies the fitment of the model for the proposed system.

The significant contribution of the proposed research work is ACVPR algorithm and meta-search tool. The IMSS-P possesses the following improvements over various page search and ranking approaches as discussed in the literature:

- (i) Proposed meta-search tool can easily predict the user preferences with accuracy by employing logistic regression based machine learning model
- (ii) Proposed meta-search tool possess characteristics of next-generation big data analytics tool and is capable of performing elastic scaling, infrastructure offloading, real-time handling of search load spikes and resource management with high reliability.
- (iii) Proposed meta-search tool is not dependent on explicit user feedback especially in intelligent mode and can automatically determine personalized preferences of the user from his/her browsing history.

2. Category specific literature review

Meta-search system when backed by machine learning technologies and next-generation big data analytics can be used for effective and efficient personalized retrieval and ranking of web pages. This research work carries out a detailed category specific literature review to quickly find the research gap between various search systems proposed by researchers time to time within literature. The conventional search systems discussed in the literature are as follows:

2.1. Review of page ranking systems based on hyperlinks

The hyperlink based personalized search systems can well assist the web user while searching for information resource on the web. It is usually assumed that customers who gave similar explicit/implicit feedback have same tastes while searching on the internet and hence various web pages are recommended to the user based on response to the webpage by a previous user having a similar profile as the current user. Alam and Sadaf (2015) suggested that the modern search engines retrieve a massive number of irrelevant and unmanageable web pages in response to a query especially when the query is incomplete or erroneous as most of the search engines tend to return result corresponding to all possible meanings of a user's query. However, clustering may be used to summarize a large number of documents in search engine output. The proper labeling of each cluster is necessary to define the content of the cluster and to assist the user in selecting a relevant cluster. They applied a heuristic search method to find all the pages of the cluster. The title of a document is an appropriate source to determine the content of the document. The label of each cluster is defined by the keywords used in the title of documents sharing hyperlinks. They took top 100 hits by searching *Jaguar* query on Google. They applied the Apriori algorithm with support = 2 for finding frequent 2 itemsets and found labels for cars, sports, and animals. The primary advantage of the proposed method is that lot of computation time could be saved as only those documents sharing hyperlinks are considered for the labeling process. However, the proposed method could be improved by considering text within meta tags for labeling process. Aoki et al. (2015) explained the system architecture of a personalized search system, i.e., the Web Index system which uses Web Index files that contain a pair of keywords and corresponding URL. The proposed method can perform *Attach* operation to associate keywords to the hyperlinks to the corresponding URL. The attach operation consists of the following substeps (i) clicking bookmark link (ii) requesting server (iii) lexicographic matching (iv) hyperlink generation (v) receiving a response and displaying page. The primary limitation of WIX system is more time required for relevancy computation.

2.2. Review of page ranking systems based on content

Kuppasamy and Aghila (2014) discussed the architecture of a personalized model to detect structural and content changes within a web page. The proposed model, i.e., *CaSePer* uses a hashing technique to identify segments used for reducing the search space and hence to quickly detect the changes within web page content. The change detection process is accompanied within two steps (i) Segmenting web page into smaller components (ii) Hash value calculation on smaller components. However, the proposed model may be improved by using advanced machine learning and big data analytics. Sugiyama et al. (2004) discussed several techniques to adapt search results to the changing needs of the web user. They carried out several experiments to verify the effectiveness of various possible approaches such as (i) Collaborative filtering based

user profiling (ii) Implicit relevance feedback (iii) Browsing history based user profiling. However, the highest accuracy was achieved by using collaborative profiling as it is more adapted to the personalized needs of the user. Yet the proposed approach can be improved by using long-term browsing history of the user.

2.3. Review of page ranking systems based on recommendation

Recommender system uses information about web user profiles, browsing history, etc. to predict the relevance of a specific web link to a web user. They make recommendations to satisfy the personalized needs of the user. Hence, a recommender system may be used as a key module for implementation of a personalized search system like the proposed Intelligent Meta Search System. Cacheda et al. (2011) carried out a detailed comparison between various collaborative filtering techniques, mentioning their strengths and limitations. They suggested two new metrics, i.e., GIM and GPIM to use prediction accuracy for determining the effectiveness of a collaborative algorithm. These two metrics can simplify the evaluation by utilizing datasets available offline. They can quickly detect any sort of bias within prediction accuracy. Wasid and Kant (2015) suggested an approach to collaborative filtering based on Fuzzy and particle swarm optimization. The discussed approach can be used to quickly learn preferences of the user and hence to provide personalized recommendations to the web user. However, the proposed system lacks the idea of concepts to improve the accuracy of personalized recommendations further. Adamopoulos (2014) discussed improvement of collaborative filtering method for enhancement of prediction accuracy for both users and businesses. The idea of unexpectedness is also addressed for meeting user expectations. However, the effectiveness of the proposed recommender systems in studying the behavior of the online user is yet to be verified.

2.4. Review of page ranking systems based on contextual knowledge

The contextual knowledge is vital for a search tool to personalized search on the web by providing hints about user interest. Xiang et al. (2010) discussed the importance of using contextual knowledge while ranking web pages. They further explained various principles and learning to rank approach to support contextual ranking of web pages. They proposed an empirical approach to solving two main issues (i) How to benefit web page ranking using contexts? (ii) How to integrate web page ranking model with contextual knowledge. However, the suggested approach will be satisfactory to deal mainly with meta page ranking in today's era of big data is yet to be verified. Tanapaisankit et al. (2012) proposed an approach for search query expansion by using contextual knowledge. They used knowledge of user's profile to make the query more personalized. The proposed method was experimentally verified to improve recall and precision parameters of page ranking. However, the proposed approach can be further enhanced by incorporating knowledge of semantics and concept tuples. Limbu et al. (2006) suggested a method to modify search queries to correctly reflect personalized tastes of the web user by utilizing implicit and explicit information such as user's browsing history and lexical database knowledge respectively. They used Thesaurus for query disambiguation and hence improved precision. Moreover, they added meta keywords for improving recall parameter of web search and page ranking. However, the process of query enhancement can be further enhanced by using a Boolean approach.

2.5. Review of page ranking systems based on intelligent techniques

Malhotra and Rishi (2018) discussed various limitations of traditional page ranking systems. They highlighted that the general

search and page ranking system is not evolved enough to work out effectively within the E-Commerce environment. They proposed Relevancy Vector page ranking algorithm that uses cloud technology and is based on the second generation big data analytics. They implemented IMSS-AE tool especially adapted to rank E-Commerce websites to suit the personalized needs of the customer. The proposed system design include search query preprocessing to enhance query by additional keywords using semantic technology. The experimental & graphical analysis compare the page ranking precision between the recommended tool and popular search engines like Google, Yahoo, Dogpile on the basis of response time, page freshness and personalized relevancy. However, the proposed work lacks accuracy in the prediction of user interest due to missing machine learning module. The proposed approach and ACVPR algorithm in the present research work is the enhancement of the RV page ranking algorithm due to the incorporation of the machine learning model. Verma et al. (2015) demonstrated usage of intelligent technologies like semantic web and neural networks for correct page ranking of E-Commerce websites. They proposed five modules (i) Module for web dictionary implementation after preprocessing of pages (ii) Module to determine priority of web page based on its textual content (iii) Module to determine priority of web page based on time spent by previous user (iv) Module for semantics-based recommendations (v) Module to determine the priority of web page using back propagation Neural Network. Malhotra et al. (2015) discussed the implementation of a meta page ranking tool to prove the effectiveness and efficiency of the proposed CPR algorithm. The proposed interface of the tool is shown in Fig. 1. The tool can use any or all of the four background search engines, i.e., Yahoo, Google, Ask and Bing. The tool will rank the various links returned by these search engines by Response time and Security protocol used by the candidate web page. However, the system does not incorporate features of personalized page ranking to satisfy the specific needs of the user.

3. Google cloud platform for big data analytics

The big data analytics is vital for a meta-search tool like the proposed IMSS-P tool to generate a most relevant page ranking order to be shown to the user to best suit the personalized requirements of a search tool user. The big data in the form of returned links by various background search engines can be easily analyzed using Hadoop-MapReduce based analytics when deployed on a cloud platform such as Google Cloud Platform. The Google Cloud Platform (GCP) is used for Hadoop based multi-node cluster setup required to implement the proposed intelligent meta Search System application. The Google cloud platform is an ideal platform for exploring various cloud services. The compute engine module allows us to create and use Virtual Machines (VM) which are virtual copies of OS servers like Linux server, Window server, etc. and let developer choose VMs with small to massive configuration in terms of CPU cores, memory, and OS image to best suit the meta-search project requirements like that of proposed IMSS-P tool. The IMSS-P tool has a multimode cluster setup of total three VM instances, one of these instances is serving the role of name node and remaining two serve the purpose of data nodes to effectively collect, store and analyze number of web links returned by the background search engines to produce the most personalized web page ranking order.

4. System design

The system design for the proposed meta-search tool has three sub-phases. The detailed description of each of the phase and simplified block diagram is shown in Fig. 2

Meta Search and Page Ranking Tool			
Select Search Engine Tabs for Intermediate Document Retrieval			
GOOGLE	YAHOO	BING	ASK
Enter Search String: HDFS and Map Reduce			
Search		Reset	
Ranking Box.....			
Rank	Web Links	Security	Response
1	https://en.wikipedia.org/wiki/Apache_Hadoop	HTTPS:	00:00:00:10ms
2	www.cloudera.com/content/cloudera/hdfs-	N/A	00:00:00:25ms
3	www.gttbm.org/software/datacom/infospher/ma	SSL	00:00:00:33ms

Fig. 1. Interface of Page Ranking Tool by Malhotra et al. (2015).

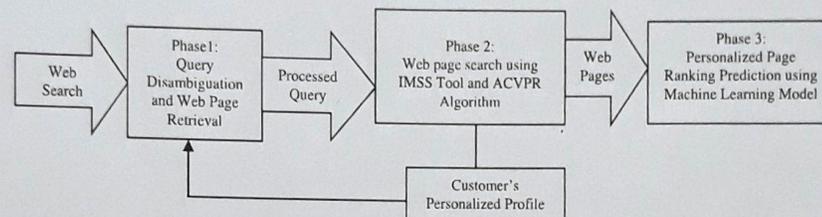


Fig. 2. System Design of Proposed IMSS - P Tool.

4.1. Phase 1: Query disambiguation and web page retrieval

In today's era of big data, even a state of art search engine is fetching output links on the top that may not be relevant to the user. Moreover, if the search query is ambiguous or incomplete, then even a popular search engine is not likely to produce the relevant result. As shown below in Fig. 3, a web search query is first processed to remove stop words and stem words to avoid their involvement while determining the relevancy of a prospective webpage. The user's recent browsing history is retrieved to determine the context of web search. For instance, after searching for a query of "Canon DSLR" on search tool, if a user inputs a partially incomplete query like "Sony" then instead of retrieving Sony Bravia television or Sony Vaio laptops on the top of the output. The tool should automatically expand the search query to a personalized and more meaningful search query, i.e., "Sony DSLR" from user's incomplete query, i.e., "Sony". Similarly, long-term preferences can be retrieved from the user's old browsing history to categorize the profile of the user and hence to enable correct and relevant expansion of web search query. The recommendation engine module can be used to build a user's profile using semantic web technology. The expanded personalized query is further passed to the number of background search engines in our meta-

search tool. The proposed tool can thus possess a good recall characteristic by retrieving massive volume of the web because of the involvement of a number of popular search engines in its background. However, as already discussed that search pages retrieved may be ranked in a biased or non-relevant order by background search engine to support paid or advertised web links. Hence top few links from each of the background search engine are passed to phase 2.

4.2. Phase 2: Web page search using ACVPR algorithm and IMSS-P tool

The proposed research work aims to design ACVPR algorithm and its implementation in the form of IMSS-P tool to determine the effectiveness and efficiency of the proposed approach. The ACVPR algorithm and IMSS-P tool are discussed below:

4.2.1. Advanced cluster vector page ranking algorithm

Advanced Cluster Vector Page Ranking Algorithm is an improvement of RV algorithm published by Malhotra et al. (2018). ACVPR is an advanced version of RV algorithm due to two main reasons (i) ACVPR is a generic search algorithm, while RV algorithm is suitable only for the search of E-Commerce websites (ii) ACVPR algorithm unlike RV algorithm incorporate machine learning based

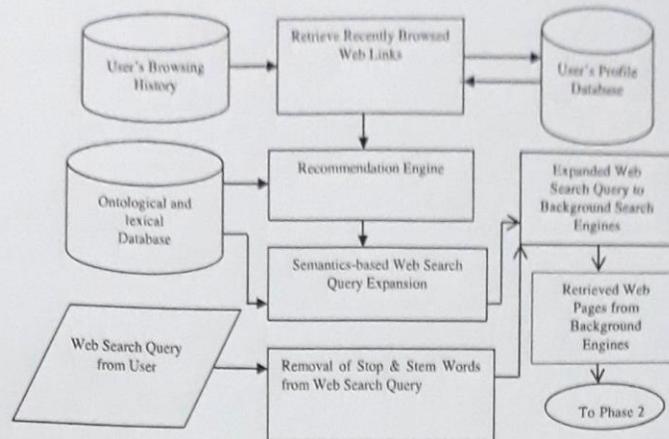


Fig. 3. Phase 1: Query Disambiguation & Web Page Retrieval for Meta Search Tool.

regression model to predict relevancy of a web page for a specific user. ACVPR algorithm is step by step discussed as follows:

- Start
- Accept a web search query from a user
- Remove stop and stem words from the query to determine keywords
- Incorporate semantic based disambiguation and personalized expansion
- Determine the weight of each of the keyword by using personalized profile database using `map()` and `reduce()` functions
- Execute a personalized search query on each of the chosen web search directory/engine incorporated within the proposed meta-search tool
- Predict relevancy of each of the returned web page from each of the search engines by generating machine learning model using `glm()` function
- Check for null deviance, residual deviance, fisher scoring and possible errors like multi-collinearity and over dispersion to determine the model effectiveness in ascertaining feedback of a prospective web page
- Remove all those web pages from the clusters for which model satisfies any of the following condition
 1. Residual deviance is more than null deviance
 2. Fisher scoring iterations are more than 8
 3. Search parameters have a substantial value of the standard error
 4. Variance inflation factor, `vif()` is more than 5
 5. over dispersion indicator is more than 0.05
- Label each of the relevant clusters with cluster ID consisting of web pages sharing hyperlinks to calculate various relevancy vectors such as `TSV[]`, `FCV[]`, etc.
- Calculate relevancy with respect to Time Spent Vector, `TSV[P]` for a candidate web page by referring to `feedback.txt` file and by calculating/assigning the average time spent by specific user on another websites with similar values of search predictors such as page loading speed etc., $TSV[P_n] = \text{avg}(TSV[P]_{1 \text{ to } n-1})$
- Calculate relevancy concerning Frequency Count Vector, `FCV[P]` by calculating average weighted contribution for the frequency of each of the keyword of a personalized search query for a candidate web page as follows:

$$FCV[P] = (W_1 * K_1 + W_2 * K_2 + \dots + W_n * K_n) / n$$

- Rank (webpage (P)) := $FCV * \beta_1 + TSV * \beta_2$, where β_1 and β_2 coefficients determine personalized preferences regarding the impact of frequency count and time spent statistic for a specific user as evident from his/her registered profile and browsing history
- Update customer personalized coefficients and `feedback.txt` after receiving feedback of above calculated and shown page ranking order

The Advanced Cluster Vector Page Ranking algorithm is designed to determine the correct ranking order of various web pages returned by the number of background search engines from phase 1 and 2 to the meta-search tool. The stop words and stem words are required to be removed from the search query to incorporate semantic based disambiguation and query expansion. The keywords of a personalized query are then weighted by referring user's personalized profile database. The weights of various keywords are stored within W_1, W_2, \dots, W_n . The personalized query is then searched over different background search engines/search directories to fetch the web pages. These web pages are then short-listed based on the comparative values of null and residual deviance, fisher scoring iterations, multi-collinearity and the value of over dispersion indicator. This process is then followed by labeling each cluster of web pages with a unique ID carrying source search engine information and consists of those web pages having hyperlinks to each other. There are two relevancy vectors required to be calculated (i) Time Spent Vector, TSV, and (ii) Frequency Count Vector, FCV. The TSV is calculated as average time spent by the specific user on various web pages with a similar value of search parameters in the past. The Frequency Count Vector, FCV is determined by the average weighted contribution of each of the keyword of a candidate web page. The rank of a candidate web page is determined by the help of personalized preference coefficients, i.e., β_1 and β_2 . Finally, the feedback obtained for current ranking order is used to update the `feedback.txt` file and personalized coefficients.

4.2.2. Intelligent meta search system – personalized tool

The IMSS-P tool is deployed on the Google cloud platform and is coded using Python programming and R statistical tool to verify

the efficiency and effectiveness of the proposed ACVPR algorithm. The IMSS-P is an advanced version of IMSS-AE tool proposed by Malhotra et al. (2018) in the sense that IMSS-P employs machine learning logistic regression to predict web user's personalized preferences and hence improves search precision. The IMSS-P is compatible with mobile phone interface and also supports biometric sign in, i.e., fingerprint scanner. Moreover, IMSS-P is generic meta-search tool and is not limited to E-Commerce search queries similar to IMSS-AE. The meta-search systems are meant to search for a query on the number of background search engines and make use of various data mining techniques to sort and merge the obtained results (Sethi et al., 2016; Gollub et al., 2018). The IMSS-P also provides advanced information on its interface such as search engine information from which a specific URL is fetched and search precision statistics. As shown in Fig. 4, IMSS-P support

two search modes (i) intelligent meta-search and (ii) advanced meta-search. The advanced mode is for a technical user and lets the user select various tabs as shown on the interface of the tool. These tabs include options to choose background search engines to be used by meta search tool, i.e., a user is free to select one or all three search engines out of Google, Bing, and Kartoo. Furthermore, options are provided to the user to select search predictors to determine the ranking order in advance mode, i.e., website response time, webpage loading speed and browsing security. The user can choose any one of these parameters to figure out the personalized page ranking order in the output of the search tool. For instance, Fig. 4 shows the output ranking order concerning page loading speed. The other features of the tool include personalized expansion of the search string, for example, user inputs partial search query, i.e., Jurassic. However, learning model

Intelligent Meta Search System- Personalized			
SIGN UP		FINGERPRINT SCANNER	
		SIGN IN	
Intelligent Meta Search		Advanced Meta Search	
Google		Bing	
		Kartoo	
Website Response Time		Web Page Loading Speed	
		Browsing Security	
Enter Web Search String: Jurassic			
CONTINUE		RESET	
Personalized String: Jurassic World: Fallen Kingdom Movie			
SUGGEST ANOTHER		SEARCH	
		CANCEL	
Web Page Ranking Output.....			
Website Rank	URL (Search Engine Initial)	Page Loading Speed	Correct Rank?
1	https://en.wikipedia.org/wiki/Jurassic_World:_Fallen_Kingdom (G)	00:00:00:25ms	CORRECT <input type="radio"/>
			WRONG <input type="radio"/>
2	http://www.jurassicworld.com/films/jurassic-world-fallen-kingdom (B)	00:00:00:42ms	CORRECT <input type="radio"/>
			WRONG <input type="radio"/>
3	https://www.cnet.com/news/jurassic-world-fallen-kingdom-review-sequel-sinks-teeth-into (G, B, K)	00:00:00:59ms	CORRECT <input type="radio"/>
			WRONG <input type="radio"/>
Search precision parameters			
PRECISION: 0.95		RECALL: 0.18	

Fig. 4. Interface of IMSS-P Tool.

checks the browsing history of the user and found that user usually likes to book movie tickets from the feedback.txt file, and hence advised latest movie released, i.e., Jurassic world: fallen kingdom movie. This personalized expansion of incomplete or ambiguous search query leads to a listing of appropriate web links in the output of various background search engines. These web links are then ranked in the order of user preference of search parameter, i.e., page loading speed and the web link with minimum page loading speed is listed on the top. Moreover, search engine information within parenthesis is also provided to show the source of the URL. For instance (G, B, K) within the last row represents that the URL was listed in the output of all the three search engines, i.e., Google, Bing, and Kartoo. The user can provide feedback regarding ranking relevancy of the web links to further improve the personalized search precision in the future search by the user. Moreover, detailed numeric statistics are also provided regarding page loading speed, search precision and search recall in the output. However, intelligent meta-search mode is implemented to provide personalized search experience to assist non-technical user while searching on the web. The intelligent mode doesn't require the user to specify background search engines and page ranking parameters. The intelligent mode will automatically use all the three search engines and uses information from feedback.txt to determine most liked page ranking parameters from various searches conducted by the user in the past. Moreover, the intelligent mode does not require the user to provide explicit feedback on the interface and can automatically determine the most suitable page ranking order to satisfy the personalized search needs of the user. The intelligent and advanced mode both implement proposed ACVPR algorithm and calculate relevancy vectors, i.e., Time Spent Vector (TSV) and Frequency Count Vector (FCV) along with ranking order parameter, e.g. page loading speed used to determine the personalized web page ranking as shown in Fig. 4.

4.3. Phase 3: Personalized page ranking prediction using machine learning model

To predict the preference of a user for a specific web page, we have developed here a machine learning model based on logistic regression. Here, the response variable to be predicted is *feedback* regarding the relevancy of ranked web link in the output of the meta-search tool by the user. The regression to be used by the proposed meta-search tool is Binomial Logistic Regression as the response variable, and *feedback* is a binary variable. The latest browsing history of the user is considered for predicting preference of the user for a new web link. The data is required to be in the .csv format as required by R statistical tool. The .csv format file will consist of the data about the following five variables:

- *Feedback* represents the relevancy response by the user for the previous web link in his browsing history and can take either of two values, i.e., Yes or No.
- *Loading* represents the web page loading experience of the user and can take either of two values, i.e., Good or Bad.
- *Response* represents the response time experience of the user and can take either of two values, i.e., Good or Bad.
- *Security* represents the security protocol feature provided by the candidate web page and can take either of two values, i.e., Yes or No.
- *Personalized* represents the usage of the feature, i.e., personalized expansion of the query by the user as available on the tool interface and can take either of two values, i.e., Yes or No.

As our response variable, i.e., *feedback* is binomial, so we will use family = binomial (link = "logit") while creating the personal-

ized search model. This syntax can be easily understood in mathematical terms as discussed below:

The natural logarithm of the odds ratio may be expressed as

$$\ln(\text{odds ratio}) = \ln\left[\frac{P}{1-P}\right] \quad (1)$$

where, P = Probability of success or probability of response, i.e., *Feedback = Yes*

$$\begin{aligned} \text{logit}(P) &= \ln\left[\frac{P(\text{Feedback} = \text{Yes})}{P(\text{Feedback} = \text{No})}\right] \\ &= C_0 + C_1 \times \text{Loading} + C_2 \times \text{Response} + C_3 \times \text{Security} \\ &\quad + C_4 \times \text{Personalized} \end{aligned} \quad (2)$$

To improve the prediction accuracy of the response variable, i.e., *feedback* by predicting the natural logarithm of the odds ratio, the probability of accurately predicting the feedback response of the web user in the proposed model may be calculated as follows:

$$\begin{aligned} \text{Probability of true feedback} &= \text{predicted odds ratio} / \\ &\quad \times (1 + \text{predicted odds ratio}) \end{aligned} \quad (3)$$

4.3.1. Steps for generating and testing machine learning model

- Reading feedback.txt file to retrieve search data
- Model generation using various search parameters
- Plotting diagnostic curves for the generated model
- Recasting model by removing non-significant search parameter
- Deviance calculation between original and recast model
- Testing for multi-collinearity and over dispersion to determine the accuracy of prediction
- Plotting diagnostic curves for recast model
- Reading feedback.txt file

To generate a logistic regression based machine learning model to predict web user personalized search preferences and hence *feedback* for a specific web link. We first need to consider his/her previous search and corresponding *feedback* data to determine his preferences. This *feedback* data is maintained in a text file within .csv format. The *feedback* file can be read using read.csv() function as discussed below:

```
feedback_data <- read.csv ("D://DheerajUOK//machinelearning//feedback.txt", header = TRUE, sep = ",")
```

As shown above, read.csv () function accepts three parameters, i.e., (i) path of the file containing user *feedback* data about previous searches on the tool, (ii) header information, i.e. whether header or column captions are there in the *feedback* file and (iii) separator information, which is comma in the case of .csv file. The *feedback.txt* information is stored within *feedback_data*, and that can be summarized to show consolidated details on various columns within the *feedback* file.

- Model generation using various search parameters

In order to generate the regression model for *feedback* prediction of prospective web link, we have used glm() function in R statistical tool with the following syntax:

```
feedback_model = glm(feedback ~ loading + response + security + personalized, data = feedback_data, family = binomial(link = "logit"))
summary(feedback_model)
```

The summary of the generated model as shown in Tables 1 and 2 represents information regarding reference value taken for each parameter, estimated contribution, standard error, and Pr, i.e., predictability value calculated using glm() function. Here, the first argument of glm() function is a response variable, i.e., *feedback* and is required to be predicted concerning the remaining parameters, i.e., *loading*, *response*, *security* and *personalized*. The

Table 1
Statistics of Various Search Parameters as Calculated by the Generalized Linear Model.

Search Parameter	Reference Value	Estimate	Std. Error	Z Value	Predictability Value
Page Loading	Good	-0.8136	0.2781	-2.925	0.003441
Response Time	Good	-0.0823	0.2643	-0.311	0.755470
Security	Yes	-1.2205	0.2784	-4.383	1.17e-05
Personalized Query	Yes	1.0193	0.2714	3.755	0.000173

Table 2
Model Deviance Statistics.

Null Deviance	373.93
Residual Deviance	331.17
Degrees of Freedom	269 (null deviance) and 265 (residual deviance)
Fisher scoring Iterations	4

estimated contribution, std. error and Z values are shown in table 1. If $Pr > 0.05$ for a specific parameter then the particular parameter is not considered as significant, for instance, as indicated above, the response variable is having $Pr = 0.755470$. Hence response variable is not significant. Further, information regarding null deviance, residual deviance, and Fisher Scoring iterations is also available within the summary of the model. A small value of Residual

deviance as compared to Null deviance shows a good model. Moreover, a Fisher scoring of fewer than eight iterations, here it is four iterations also strengthens the fact that model generated is a good model and can efficiently and correctly predict the dependence of user's feedback value over other search parameters such as response time, page loading speed, security, and personalized relevancy.

• Plotting diagnostic curves for the generated model

There are four diagnostic curves plotted for the generated model, feedback_model as shown in Figs. 5 to 8. The detailed interpretation of various diagnostic plots is being discussed after recasting feedback_model to feedback_model2.

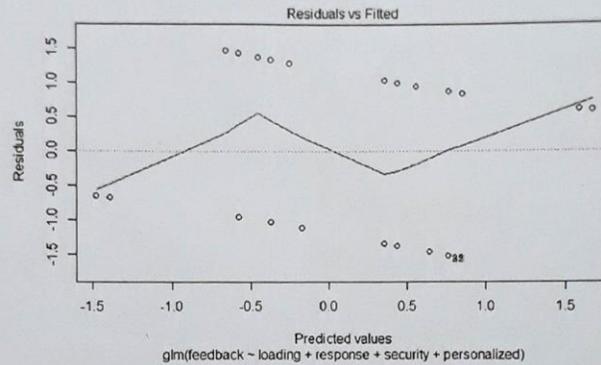


Fig. 5. Feedback_Model Diagnostic Plot - Residuals vs. Fitted.

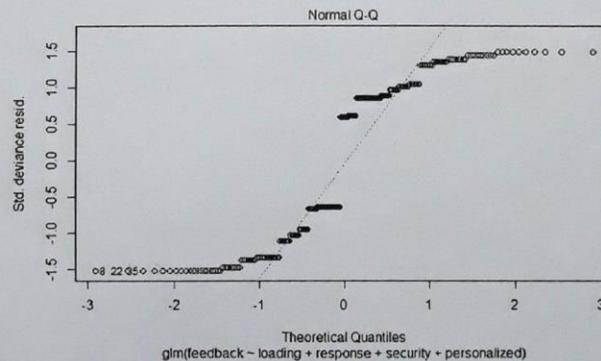


Fig. 6. Feedback_Model Diagnostic Plot - Normal Q-Q.

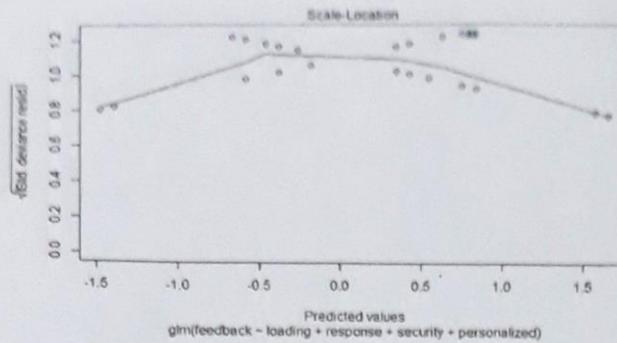


Fig. 7. Feedback_Model Diagnostic Plot - Scale-Location.

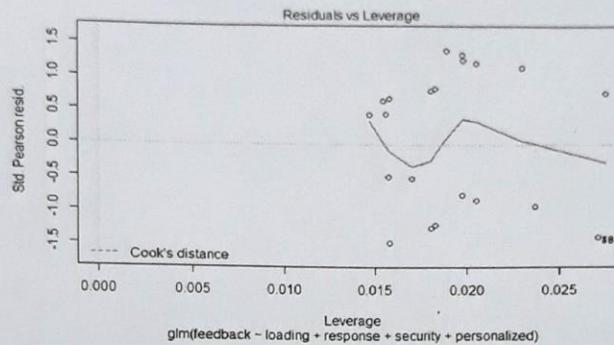


Fig. 8. Feedback_Model Diagnostic Plot - Residuals vs. Leverage.

• **Recasting model by removing non-significant search parameter**

As evident from the summary and diagnostic curves of the generated model, we may further improve the generalized linear model via recasting. The new model may be improved by eliminating search parameters having P value more than 0.05, i.e., after removal of the response parameter. The command to generate new feedback_model2 without response parameter is as follows:

```
feedback_model2 = glm(feedback ~ loading + security + personalized, data = feedback_data, family = binomial(link = "logit")) summary(feedback_model2)
```

After recasting model, statistics generated for various parameters are shown in (Tables 3 and 4).

Table 3
Statistics of Various Search Parameters for Recast Generalized Linear Model.

Search Parameter	Reference Value	Estimate	Std. Error	Z Value	Predictability Value
Page Loading	Good	-0.08146	0.2779	-2.931	0.003383
Security	Yes	-1.2214	0.2783	-4.389	1.14e-05
Personalized Query	Yes	1.0205	0.2712	3.762	0.000168

Table 4
Model Deviance Statistics for Recast Model.

Null Deviance	373.93
Residual Deviance	331.26
Degrees of Freedom	269 (null deviance) and 266 (residual deviance)
Fisher scoring Iterations	4

• **Deviance calculation between original and recast model**

The deviance between original and recast model can be calculated using anova () function as shown below:

```
anova (feedback_model, feedback_model2, "PChisq")
```

The first two arguments of anova () will be two generalized models to be compared using PChisq test. The difference between the degrees of freedom for both the models is one while calculating residual deviance. The deviance difference calculated is -0.096992. The small deviance difference between two models represents a low impact of response parameter on the generalized linear model.

• **Testing for multi-collinearity and over dispersion to determine the accuracy of prediction**

To check whether the generated model suffers from multi-collinearity or over dispersion, we need first to install the DAAG library in R studio. In order to check multi-collinearity in the recast model. We will use the vif() function with feedback_model as an

Table 5
Vif Value for Various Search Parameters used while Generating the Model.

Search Parameter	Vif value
Loading	1.0986
Security	1.0924
Personalized	1.0063

argument. The statistics for various search parameters obtained using vif is given in Table 5.

The vif stands for Variance Inflation Factor. As shown above, vif value for various search parameters is less than 5. Hence, the model is not suffering from multicollinearity.

The command to calculate and test for overdispersion for recast personalized search model is

```
overdisp_indicator <- feedback_model2$residuals / feedback_model2$df.residual
```

As calculated in R, overdispersion indicator value is less than 0.5, hence our model is not suffering from overdispersion. So the

generated model can accurately predict the user's personalized preferences while searching the web.

• Plotting Diagnostic Curves for recast feedback_model2

After carrying out regression analysis, we plotted several diagnostic plots before and after recasting machine learning model to show and compare residuals in four different ways as shown in Figs. 5 to 8 for feedback_model and from Figs. 9 to 12 for feedback_model2. The four diagnostic curves for each of the model are as follows:

- (i) Residuals vs. Fitted Values Plot
- (ii) Normal QQ Plot-Standard Residuals vs. Theoretical Quantities
- (iii) Scale -Location plot-Standard Residuals vs. Fitted Values
- (iv) Standard Residuals vs. Leverage Plot

The first plot, i.e., Residuals v/s Fitted values shows a non-linear relationship between response and predictor variables.

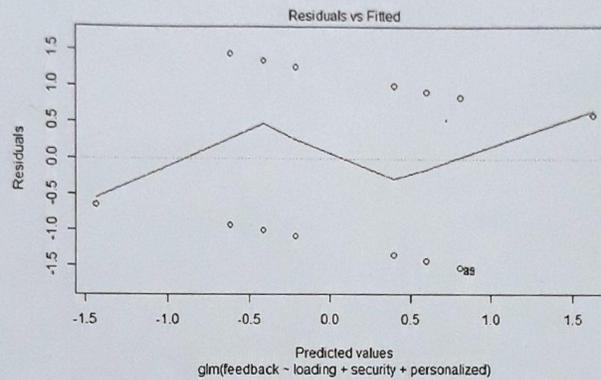


Fig. 9. Feedback_Model2 Diagnostic Plot - Residuals vs. Fitted.

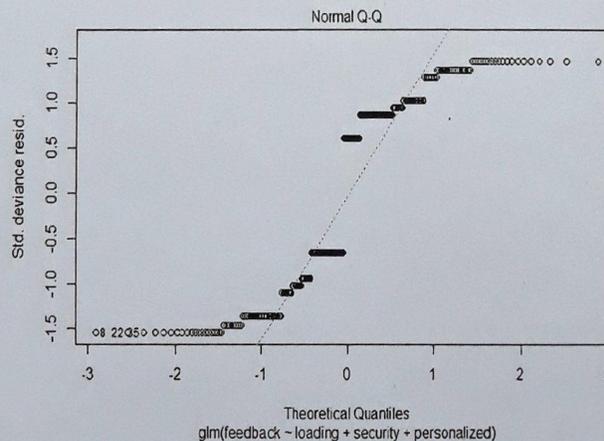


Fig. 10. Feedback_Model2 Diagnostic Plot - Normal Q-Q.

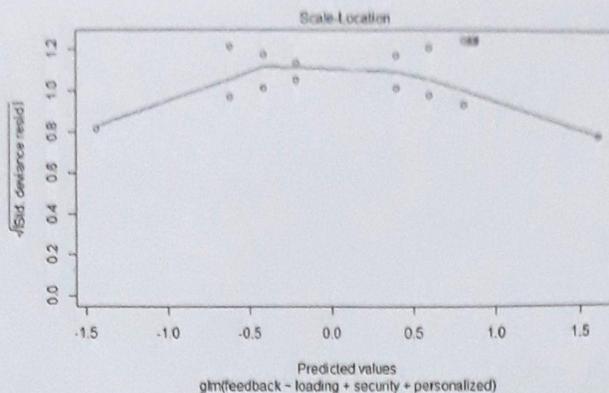


Fig. 11. Feedback_Model2 Diagnostic Plot - Scale-Location.

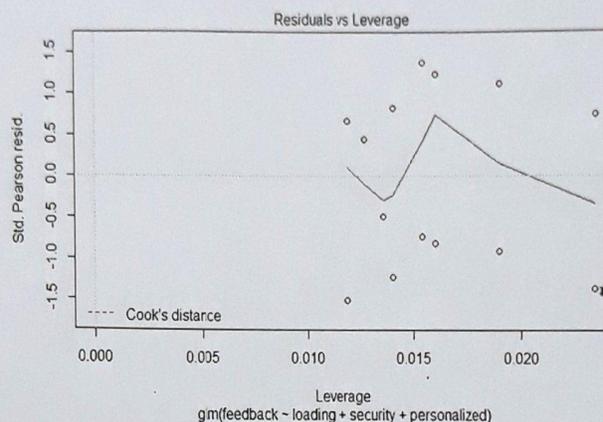


Fig. 12. Feedback_Model2 Diagnostic Plot - Residuals vs. Leverage.

The points lying at the fit line, i.e., dotted line at $y=0$ represent zero residuals while the points lying above the fit line represent positive residuals and below the fit line represent negative residuals. The smooth red non-linear curve represents an excellent fitted model for both original `feedback_model` and recasts `feedback_model2`.

The second plot, i.e., QQ plot shows whether residuals follow linear normal distribution or not. The points are shown as firmly placed near the dotted line in both cases. Hence both `feedback_model` and `feedback_model2` passes normal distribution test.

The third plot, i.e., Scale Location plot is also sometimes referred as Spread Location plot as it represents a pattern of spreading points across the range of predicted values. The ideal scale-location curve is horizontal and represents Homoscedasticity, i.e., a uniform variation of points across the expected range. However, in our case, the curve for intermediate points is Homoscedastic and for initial and final points is Heteroscedastic in nature. This red curve represents that the proposed model will work well for the intermediate number of search predictors and not with very small or large number of predictor variables and the same is true for

intermediate data observations of generated models, i.e., `feedback_model` and `feedback_model2`.

The fourth plot, i.e., Residuals v/s leverage assists in finding those observations that can potentially determine a regression line. There are a majority of observations that may be included or excluded without affecting the result of the analysis. However, a few observations can hugely impact the regression line and can change the outcome of the analysis. Whenever observations have a high value of Cook's distance scores, they can easily influence the result of the analysis. The points shown near the $y=0$ line represent all those feedback data observations having the high value of Cook's distance scores and hence cannot be excluded from the data used for regression analysis.

5. Experimental and graphical analysis

To evaluate the effectiveness of the machine learning model, meta-search tool data sets within the `feedback.txt` file are subdivided into training and testing data using various commands in R studio as discussed below:

```
feedback_data_partition <- createDataPartition(feedback_data$fe
edback, p = 0.80, list = false)
feedback_training_data <- feedback_data[ feedback_data_
partition, ]
feedback_testing_data <- feedback_data[ -feedback_data_
partition, ]
```

Here we divided a user-specific previous search feedback_data, initially provided within feedback.csv with a total of 270 observations into two parts, i.e., feedback_training_data consisting of 80% (0.80) observations, while feedback_testing_data consists of rest 20% observations. We may also use randomly generated subsets of the data sets to be used as training dataset and testing dataset for accurate evaluation of the generalized linear model. After creating the data partition, the summarized details can be verified to check the random distribution of records within training and testing data supported by the fact that the serial number of various observations included within training data is not same as that of testing data. Moreover the observations used within testing data are not sequentially picked from total observations; instead, randomly chosen observations are included. Furthermore, summarized details also verify the data partition with the 80–20 ratio, i.e., out of 270 observations, 216 observations are contained within the training data, and the remaining 54 observations are included within testing data.

5.1. Training the model

In the next step, a generalized linear model will be generated using feedback_training_data via glm () function in a similar way as described earlier while casting and recasting feedback_model and feedback_model2. Here, Null deviance of the generated model with training data is 299.717 with 216-1 = 215 degrees of freedom, while residual deviance is 258.152 with 216-1-4 = 211 degrees of freedom as the model calculates residual deviance by subtracting the number of search parameters. Moreover, Fisher scoring iterations of training data model are 7. As evident from fisher scoring and deviation statistics, model is likely to predict accurately the feedback of prospective web link by the specific web user to successfully implement personalized search system. Here the training phase is over. However, further analysis was required to verify the predictions of the generated model by using the remaining 20% of the test data.

5.2. Testing the model using confusion matrix

The procedure used for testing the model may be step by step summarized as follows:

- Use the generated training_feedback_model as described in Section 5.1 to predict the response variable for all observations within the test data.
- The predicted response variable was compared with the actual values of the response variables stored within the test data.
- After comparison, a confusion matrix was generated to determine False Negatives (FN), True Positives (TP), False Positives (FP) and True Negatives (TN). Here False Negatives stand for observations in the test data that were predicted as negative (0) but were positives(1). The True Positives stand for observations in the test data that were predicted as positive(1) and were positive(1). The False Positives stand for observations within test data that were predicted as Positive(1) but were negative(0) while True Negatives stand for observations within test data that were predicted as negative(0) and were negative (0). The accurate model will generate more True Positives and more True Negatives and no or negligible False Positives and False Negatives to verify its effectiveness in predicting accuracy.

The confusion matrix for testing data, i.e., feedback_testing_data is shown in Table 6. As shown, the number of True Positives and True Negatives are much higher than the False Positives and False Negatives. Hence the model generated is an accurately trained model. There were 49 observations (TN-20+TP-29) being accurately predicted, and only five observations were wrongly predicted (FN-2+FP-3) out of total 54 observations in test data, i.e., feedback_testing_data.

5.3. Evaluation metrics

In order to determine the prediction accuracy of the generated model, we will evaluate and plot following evaluation metrics:

- Accuracy
- Specificity
- Sensitivity
- Precision
- Recall

The mathematical expressions/ formulas used to calculate all of above evaluation metrics are as follows:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / \text{Total Observations} \quad (4)$$

$$\text{Specificity} = \text{True Negative Rate} = \text{TN} / (\text{FP} + \text{TN}) \quad (5)$$

$$\begin{aligned} \text{Sensitivity} = \text{Recall} = \text{True Positive Rate} &= \text{TP} / (\text{FN} + \text{TP}) \\ &= \text{TP} / (\text{All Positives}) \end{aligned} \quad (6)$$

$$\text{Precision} = \text{TP} / (\text{FP} + \text{TP}) \quad (7)$$

These metrics can be evaluated for confusion matrix shown in table 6 as discussed below:

Here, TP = 29, TN = 20, FN = 2 and FP = 3

Therefore, using Eqs. (4)–(7)

$$\text{Accuracy} = (29 + 20) / 54 = 0.9074 = 90.74\%$$

$$\text{Specificity} = 20 / (3 + 20) = 0.8695 = 86.95\%$$

$$\text{Sensitivity} = \text{Recall} = \text{TPR} = 29 / (2 + 29) = 0.9354 = 93.54\%$$

$$\text{Precision} = 29 / (3 + 29) = 0.9062 = 90.62\%$$

As shown above, high values of accuracy, specificity, sensitivity, precision, and recall metrics proved the prediction accuracy of the proposed ACVPR algorithm and personalized meta search tool, IMSS-P.

We can use performance () function of ROCR package to obtain $\text{TPR} = \text{TP} / (\text{FN} + \text{TP}) = \text{TP} / (\text{All Positives})$ $\text{FPR} = \text{FP} / (\text{FP} + \text{TN}) = \text{FP} / (\text{All Negatives})$ and plot TPR against FPR to obtain Receiver Operating Characteristic, i.e. ROC curve. The TPR statistic represents a number of positives which were correctly predicted by the proposed IMSS-P tool, i.e., positive feedback for a prospective web page suggested by the machine learning model to adapt to personalized requirements of the specific web user and was accepted as relevant by the user. This prediction is further supported by the web user when in his/her feedback, web page was marked as relevant. On the other side, FPR represents those positive predictions that were not finally marked as relevant by the user. The ROC curve in Fig. 13 represents that TPR is improving rapidly with respect to

Table 6
Confusion Matrix for Feedback_Testing_Data.

	FALSE	TRUE
NEGATIVES (0)	2	20
POSITIVES (1)	3	29

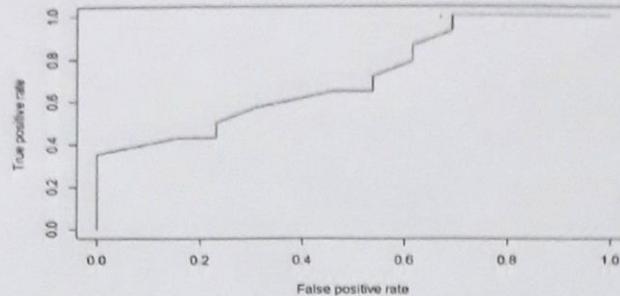


Fig. 13. TPR vs. FPR for the Proposed Regression Model.

FPR. This observation proves the capabilities of IMSS-P in correctly predicting relevant web links for the user.

The specificity represents true negative rate which means all those negative predictions by the machine learning model of proposed IMSS-P regarding a specific web link returned by the background search engine that was also marked as non-relevant by the user. On the other hand, sensitivity represents the true positive rate, i.e., high values of both sensitivity and specificity metrics support the accurate prediction by proposed ACVPR algorithm and meta-search tool. The ROC curve in Fig. 14 indicates that sensitivity falls with the increase in specificity. This curve is again supporting the accurate web page prediction by the proposed model and tool. Moreover, both sensitivity and specificity cannot simultaneously be dominant because of the binomial nature of response variable, i.e., feedback, as, either a suggested web link by the proposed model is marked as relevant/positive (1) or non-relevant/negative(0) by the web user in his/her feedback.

In web search domain, precision metric represent relevance of returned web links in the output while recall represents the comprehensiveness of the search result. For instance, a web search tool returns 10 relevant web links out of total 50 links in its output. However, it misses to return any of the remaining 60 more relevant web links then the precision of the search tool is $10/50 = 1/5$. How-

ever, recall of the search tool is $10 / (10 + 60) = 1/7$. The Fig. 15 represents a plot between precision and recall for the proposed IMSS-P tool. Here, precision is initially constant; however at larger values of recall, precision starts falling. This relationship represents that when the total number of relevant links returned by background search engines are less than proposed meta-search tool can quickly identify the relevant links and identify their correct ranking because of the natural contrast between most relevant and most non-relevant. However, when the recall is more, i.e., large numbers of relevant links are returned by background search engine to proposed meta-search tool then contrast between various returned web links is less obvious and hence it would be little tricky for learning model to identify the correct rank of each of the relevant web link. This problem can be efficiently sorted by identifying and incorporating more personalized search parameters to maintain a good contrast between various web links and hence to quickly decide the ranking order among relevant web links.

5.4. Search parameter comparison of IMSS_P with popular search engines

The superiority of the proposed IMSS-P tool and ACVPR algorithm over traditional search engines was verified by human

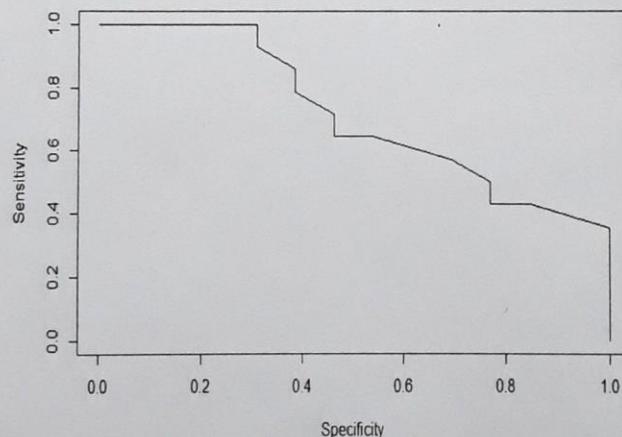


Fig. 14. Sensitivity vs. Specificity for the Proposed Regression Model.

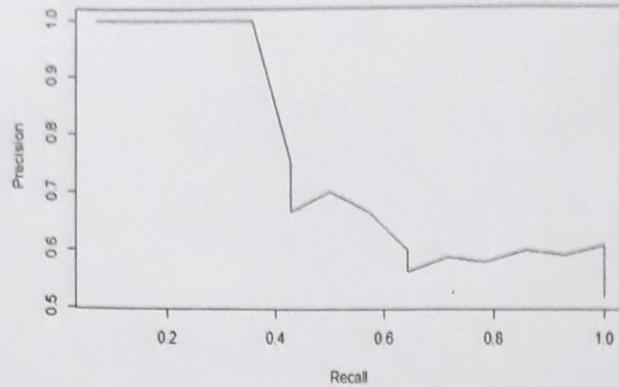


Fig. 15. Precision vs. Recall for the Proposed Regression Model.

volunteers. There is a total of thirty human volunteers employed for precision comparison of proposed IMSS-P with two popular search engines, Google and Bing plus a meta-search engine, Kartoo. Out of total thirty volunteers, twenty are females and rest ten are males within the age group of 25 years to 55 years with a minimum ten years of experience to carry out web search and browsing. They were asked to access the IMSS-P application on Google Cloud Platform using the external IP address of the tool. After completing the signup process, volunteers were asked to repeat at least ten times following steps on IMSS-P tool to allow the tool to learn their individual personalized preferences and to build their feedback.csv file. The queries searched by volunteers were also searched on Google, Bing, and Kartoo to rank these search tools by determining various search precision parameters.

- Firstly, volunteers were asked to search a partial or ambiguous search query on all the search tools under consideration, for instance, search for an incomplete query like *Jurassic* rather than *Jurassic Park Movie tickets* or *Jurassic Waterpark timings*.
- Secondly, volunteers were asked to assign weight to various search precision parameters in between 1(worst) to 5(best) to top five web links produced in the output of the considered search engines.

- The precision data obtained is then normalized by recursively applying following equation on each precision parameter for each of the candidate web page returned by the background search engine to IMSS-P using following equation, Normalized Value (NV) is given by:

$$NV = \frac{(\text{Maximum value of parameter}) - (\text{Measured value of parameter})}{(\text{Maximum Value} - \text{Minimum Value})}$$

- The overall precision of a candidate web page is then obtained using the weighted summation of the normalized value of each of the precision parameter
- Finally, the precision of the background search tool/engine is then calculated by the average of all weighted summations as reported by all the thirty volunteers.

The bar plot shown in Fig. 16 is comparing the precision of various search parameters represent that the human volunteer's judgment proves that the precision of IMSS-P is better than Google, Bing, and Kartoo. The precision of each of the parameter, i.e., Response Time, Page Updation and Personalized Relevancy of IMSS-P dominate over to that of Google, Bing, and Kartoo. This

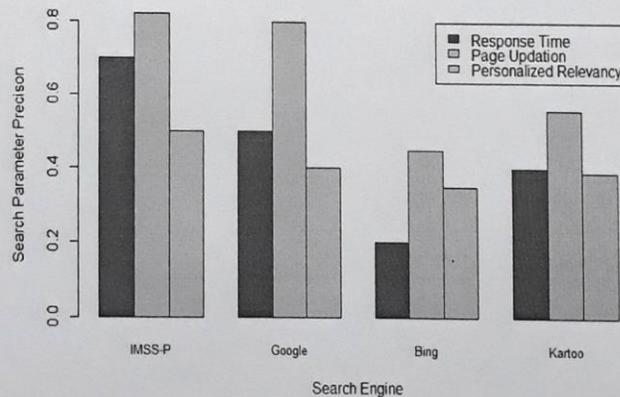


Fig. 16. Search Precision Comparison between IMSS-P with Popular Search Engines.

observation, in turn, proves the effectiveness and efficiency of the proposed ACVPR algorithm and logistic regression based machine learning model to merge the output links from all three professional search engines, i.e., Google, Bing, and Kartoo. Hence IMSS-P is utilizing the strength of all of its three background engines and offering the benefits of all of these under a single search platform by merging the top links of each of the competitive search engines to satisfy the personalized needs of the user.

6. Conclusion and future work

This research work proposes ACVPR algorithm and architecture of a meta-search system, IMSS-P. This search tool is backed by intelligent and advanced technologies like semantic web and Hadoop2 based big data analytics to predict meta keywords to expand the user search query and to handle a massive number of web links returned by background search engines. The IMSS-P tool is implemented using proposed machine learning enabled Advanced Cluster Vector Page Ranking Algorithm (ACVPR). This meta-search system can suitably expand a partial or ambiguous search query from a user to satisfy the personalized needs of the web user. The tool can easily predict the personalized relevancy of a prospective web link returned by backend search engines, i.e., Google, Bing, and Kartoo for a specific user. The tool can also remove those web links from its final output which are not relevant as evident from user personalized profile, browsing history and the same is decided by the machine learning model based on logistic regression. The proposed logistic regression model undergoes extensive training and testing and hence can well predict the binomial response variable, i.e., feedback of a user about a prospective web link. The capabilities of the proposed model are verified by generating a confusion matrix which in turn assisted in evaluating various metrics like specificity, sensitivity, TPR, FPR, precision, and recall. The graphical ROC plots between TPR v/s FPR, specificity v/s sensitivity and precision v/s recall establish the effectiveness and efficiency of the proposed logistic machine learning model, ACVPR algorithm, and IMSS-P tool. The diagnostic curves for both feedback_model and feedback_model2 represent the successful generation of machine learning model for accurate prediction of relevant web links. Moreover, human volunteers judgment regarding the better precision of various predictor variables of IMSS-P when compared with popular search engines like Google, Bing and Kartoo also establish the fact that proposed machine learning model, IMSS-P tool, and ACVPR algorithm can well satisfy the personalized search needs of the user. The major application of the proposed research work is personalized meta-search application to assist end user while browsing the web. The practical significance for various audiences of the proposed study is discussed below:

Significance for End User: The end user of the proposed personalized meta-search system is a web user wishing to locate a relevant URL to satisfy his personalized web search requirements. The proposed IMSS-P tool can assist the end user in searching and ranking web links in the personalized order. These web links produced in the output of the proposed tool are free from biased ranking. The conventional search engines usually show advertised/paid and hence irrelevant web links on the top of their search output. Thus, IMSS-P saves a lot of time and energy of the user spent in searching a relevant web link as compared to traditional search engines. The personalized proposed tool is a meta-search tool which in turn will combine relevant top links from different search engines and hence recall of the result will be better than recall achieved by conventional search engines.

Significance for Website Owners: The website owners will be motivated to build user-friendly websites rather than search

engine friendly websites. The site that has the potential to satisfy the personalized needs of the user will automatically be listed among the top links in search engine output without any fear of biased ranking or wrong ranking support to paid incompetent web link by the search engine. This assurance, in turn, will motivate online businesses for positive competition.

Significance for Researchers & Developers: The proposed research work will motivate researchers and developers to design and develop various meta-search applications by using the potential of machine learning based big data analytics and hence to improve the experience of the end user on the web by incorporating more and more powerful personalized search algorithms.

In future, Advanced Cluster Vector page Ranking Algorithm (ACVPR) and IMSS-P tool can be further refined to perform an image-based personalized web search, i.e., search for useful & personalized web links using images. The face recognition based web search can be used to find helpful web links or to locate parents and address of a lost child on WWW or social media using his/her image. The proposed research work can also be enriched by incorporating domain-based search tabs on the interface of IMSS-P tool. The domain-specific search tabs may include tabs specialized for e-commerce websites, airline websites to compare and contrast a particular product/ticket offering from many online businesses. This feature can assist customers in easy searching a specific webpage to satisfy his/her personalized requirements without manually requiring to visit many websites to compare the offerings.

References

- Adamopoulos, P., 2014. On discovering non-obvious recommendations: Using unexpectedness and neighborhood selection methods in collaborative filtering systems. *Proceedings of the 7th ACM international conference on Web search and data mining*, ACM, 655-660.
- Ahmad, M.W., Doja, M.N., Ahmad, T., 2017. Enumerative feature subset based ranking system for learning to rank in presence of implicit user feedback. *J. King Saud Univ. - Comput. Inf. Sci.* Elsevier.
- Alam, M., Sadaf, K., 2015. Labeling of web search result clusters using heuristic search and frequent itemset. *Procedia Comput. Sci.* Elsevier, 216-222.
- Aoki, Y., Koshijima, R., Toyama, M., 2015. Automatic determination of hyperlink destination in web index. In: *Proceedings of the 19th International Database Engineering & Applications Symposium*, pp. 206-207.
- Boudajenek, M.R., Hacid, H., Bouzeghoub, M., Vakali, A., 2016. Persador: personalized social document representation for improving web search. *Inf. Sci.* Elsevier 369, 614-633.
- Cacheda, F., Carneiro, V., Fernández, D., Formoso, V., 2011. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Trans. Web (TWEB)* 5 (1). <https://doi.org/10.1145/1921591.1921593>.
- Chawla, S., 2018. Web page recommender system using hybrid of genetic algorithm and trust for personalized web search. *J. Inf. Tech. Res. (JITR)* 11 (2), 110-127.
- Ferretti, S., Mirri, S., Prandi, C., Salomoni, P., 2016. Automatic web content personalization through reinforcement learning. *J. Syst. Softw.* Elsevier 121, 157-169.
- Gollub, T., Genc, E., Lipka, N., Stein, B., 2018. Pseudo descriptions for meta-data retrieval. In: *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*, ACM, pp. 139-146.
- Gomez-Nieto, E., San Roman, F., Pagliosa, P., Casaca, W., Helou, E.S., de Oliveira, M.C.F., Nonato, L.G., 2014. Similarity preserving snippet-based visualization of web search results. *IEEE Trans. Visual. Comput. Graphics* 20 (3), 457-470.
- Gupta, D., Singh, S.K., Malhotra, D., Verma, N., 2017. EPRT-An ingenious approach for e-commerce website ranking. *Int. J. Comput. Intell. Res.* 13 (6), 1471-1482.
- Kuppusamy, K.S., Aghila, G., 2014. CaSePer: An efficient model for personalized web page change detection based on segmentation. *J. King Saud Univ.-Comput. Inf. Sci.* Elsevier 26 (1), 19-27.
- Limbu, D.K., Connor, A., Pears, R., MacDonell, S., 2006. Contextual relevance feedback in web information retrieval. In: *Proceedings of the 1st International Conference on Information Interaction in Context*, ACM, pp. 138-143.
- Malhotra, D., 2014. Intelligent web mining to ameliorate web page rank using back-propagation neural network. In: *Confluence the Next Generation Information Technology Summit (Confluence)*, 2014 5th International Conference. IEEE, pp. 77-81.
- Malhotra, D., Verma, N., 2013. An ingenious pattern matching approach to ameliorate web page rank. *Int. J. Comput. Appl.* 65 (24), 33-39.
- Malhotra, D., Malhotra, M., Rishi, O.P., 2015. An innovative approach of web page ranking using Hadoop- and Map Reduce-based cloud framework. *Proceedings*

observation, in turn, proves the effectiveness and efficiency of the proposed ACVPR algorithm and logistic regression based machine learning model to merge the output links from all three professional search engines, i.e., Google, Bing, and Kartoo. Hence IMSS-P is utilizing the strength of all of its three background engines and offering the benefits of all of these under a single search platform by merging the top links of each of the competitive search engines to satisfy the personalized needs of the user.

6. Conclusion and future work

This research work proposes ACVPR algorithm and architecture of a meta-search system, IMSS-P. This search tool is backed by intelligent and advanced technologies like semantic web and Hadoop2 based big data analytics to predict meta keywords to expand the user search query and to handle a massive number of web links returned by background search engines. The IMSS-P tool is implemented using proposed machine learning enabled Advanced Cluster Vector Page Ranking Algorithm (ACVPR). This meta-search system can suitably expand a partial or ambiguous search query from a user to satisfy the personalized needs of the web user. The tool can easily predict the personalized relevancy of a prospective web link returned by backend search engines, i.e., Google, Bing, and Kartoo for a specific user. The tool can also remove those web links from its final output which are not relevant as evident from user personalized profile, browsing history and the same is decided by the machine learning model based on logistic regression. The proposed logistic regression model undergoes extensive training and testing and hence can well predict the binomial response variable, i.e., feedback of a user about a prospective web link. The capabilities of the proposed model are verified by generating a confusion matrix which in turn assisted in evaluating various metrics like specificity, sensitivity, TPR, FPR, precision, and recall. The graphical ROC plots between TPR v/s FPR, specificity v/s sensitivity and precision v/s recall establish the effectiveness and efficiency of the proposed logistic machine learning model, ACVPR algorithm, and IMSS-P tool. The diagnostic curves for both feedback_model and feedback_model2 represent the successful generation of machine learning model for accurate prediction of relevant web links. Moreover, human volunteers judgment regarding the better precision of various predictor variables of IMSS-P when compared with popular search engines like Google, Bing and Kartoo also establish the fact that proposed machine learning model, IMSS-P tool, and ACVPR algorithm can well satisfy the personalized search needs of the user. The major application of the proposed research work is personalized meta-search application to assist end user while browsing the web. The practical significance for various audiences of the proposed study is discussed below:

Significance for End User: The end user of the proposed personalized meta-search system is a web user wishing to locate a relevant URL to satisfy his personalized web search requirements. The proposed IMSS-P tool can assist the end user in searching and ranking web links in the personalized order. These web links produced in the output of the proposed tool are free from biased ranking. The conventional search engines usually show advertised/paid and hence irrelevant web links on the top of their search output. Thus, IMSS-P saves a lot of time and energy of the user spent in searching a relevant web link as compared to traditional search engines. The personalized proposed tool is a meta-search tool which in turn will combine relevant top links from different search engines and hence recall of the result will be better than recall achieved by conventional search engines.

Significance for Website Owners: The website owners will be motivated to build user-friendly websites rather than search

engine friendly websites. The site that has the potential to satisfy the personalized needs of the user will automatically be listed among the top links in search engine output without any fear of biased ranking or wrong ranking support to paid incompetent web link by the search engine. This assurance, in turn, will motivate online businesses for positive competition.

Significance for Researchers & Developers: The proposed research work will motivate researchers and developers to design and develop various meta-search applications by using the potential of machine learning based big data analytics and hence to improve the experience of the end user on the web by incorporating more and more powerful personalized search algorithms.

In future, Advanced Cluster Vector page Ranking Algorithm (ACVPR) and IMSS-P tool can be further refined to perform an image-based personalized web search, i.e., search for useful & personalized web links using images. The face recognition based web search can be used to find helpful web links or to locate parents and address of a lost child on WWW or social media using his/her image. The proposed research work can also be enriched by incorporating domain-based search tabs on the interface of IMSS-P tool. The domain-specific search tabs may include tabs specialized for e-commerce websites, airline websites to compare and contrast a particular product/ticket offering from many online businesses. This feature can assist customers in easy searching a specific webpage to satisfy his/her personalized requirements without manually requiring to visit many websites to compare the offerings.

References

- Adamopoulos, P., 2014. On discovering non-obvious recommendations: Using unexpectedness and neighborhood selection methods in collaborative filtering systems. *Proceedings of the 7th ACM international conference on Web search and data mining*, ACM, 655-660.
- Ahmad, M.W., Doja, M.N., Ahmad, T., 2017. Enumerative feature subset based ranking system for learning to rank in presence of implicit user feedback. *J. King Saud Univ. - Comput. Inf. Sci.* Elsevier.
- Alam, M., Sadaf, K., 2015. Labeling of web search result clusters using heuristic search and frequent itemset. *Procedia Comput. Sci.* Elsevier, 216-222.
- Aoki, Y., Koshijima, R., Toyama, M., 2015. Automatic determination of hyperlink destination in web index. In: *Proceedings of the 19th International Database Engineering & Applications Symposium*, pp. 206-207.
- Boudajenek, M.R., Hacid, H., Bouzeghoub, M., Vakali, A., 2016. Persador: personalized social document representation for improving web search. *Inf. Sci.* Elsevier 369, 614-633.
- Cacheda, F., Carneiro, V., Fernández, D., Formoso, V., 2011. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Trans. Web (TWEB)* 5 (1). <https://doi.org/10.1145/1921591.1921593>.
- Chawla, S., 2018. Web page recommender system using hybrid of genetic algorithm and trust for personalized web search. *J. Inf. Tech. Res. (JITR)* 11 (2), 110-127.
- Ferretti, S., Mirri, S., Prandi, C., Salomoni, P., 2016. Automatic web content personalization through reinforcement learning. *J. Syst. Softw.* Elsevier 121, 157-169.
- Gollub, T., Genc, E., Lipka, N., Stein, B., 2018. Pseudo descriptions for meta-data retrieval. In: *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*. ACM, pp. 139-146.
- Gomez-Nieto, E., San Roman, F., Pagliosa, P., Casaca, W., Helou, E.S., de Oliveira, M.C.F., Nonato, L.G., 2014. Similarity preserving snippet-based visualization of web search results. *IEEE Trans. Visual. Comput. Graphics* 20 (3), 457-470.
- Gupta, D., Singh, S.K., Malhotra, D., Verma, N., 2017. EPRT-An ingenious approach for e-commerce website ranking. *Int. J. Comput. Intell. Res.* 13 (6), 1471-1482.
- Kuppusamy, K.S., Aghila, G., 2014. CaSePer: An efficient model for personalized web page change detection based on segmentation. *J. King Saud Univ.-Comput. Inf. Sci.* Elsevier 26 (1), 19-27.
- Limbu, D.K., Connor, A., Pears, R., MacDonell, S., 2006. Contextual relevance feedback in web information retrieval. In: *Proceedings of the 1st International Conference on Information Interaction in Context*. ACM, pp. 138-143.
- Malhotra, D., 2014. Intelligent web mining to ameliorate web page rank using back-propagation neural network. In: *Confluence the Next Generation Information Technology Summit (Confluence)*, 2014 5th International Conference. IEEE, pp. 77-81.
- Malhotra, D., Verma, N., 2013. An ingenious pattern matching approach to ameliorate web page rank. *Int. J. Comput. Appl.* 65 (24), 33-39.
- Malhotra, D., Malhotra, M., Rishi, O.P., 2015. An innovative approach of web page ranking using Hadoop- and Map Reduce-based cloud framework. *Proceedings*

- of *Advances in Intelligent Systems and Computing*, vol. 634. CSI Springer, pp. 421–427.
- Malhotra, D., Rishi, O.P., 2016. IMSS-E: An intelligent approach to design of adaptive meta search system for e-commerce website ranking. In: *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*. ACM. <https://doi.org/10.1145/2979773.2979782>.
- Malhotra, D., Rishi, O.P., 2017a. IMSS: a novel approach to design of adaptive search system using second generation big data analytics. In: *Proceedings of International Conference on Communication and Networks*. Springer, pp. 189–198.
- Malhotra, D., Verma, N., Rishi, O.P., Singh, J., 2017. Intelligent big data analytics: adaptive e-commerce website ranking using apriori hadoop-bdas-based cloud framework. *Maximizing Business Performance and Efficiency Through Intelligent Systems*, IGI Global, pp. 50–72.
- Malhotra, D., Rishi, O.P., 2018. An intelligent approach to design of E-Commerce metasearch and ranking system using next-generation big data analytics. *J. King Saud Univ.-Comput. Inf. Sci.*, Elsevier.
- Salonen, V., Karjalainen, H., 2016. Web personalization: the state of the art and future avenues for research and practice. *Telematics Inf.*, Elsevier 33 (4), 1088–1104.
- Sethi, S., Malhotra, D., Verma, N., 2016. Data mining: current applications & trends. *Int. J. Innovations Eng. Technol.* 6 (4), 586–589.
- Sugiyama, K., Hatano, K., Yoshikawa, M., 2004. Adaptive web search based on user profile constructed without any effort from users. In: *Proceedings of the 13th International Conference on World Wide Web*. ACM, pp. 675–684.
- Tanajaisankit, P., Watrous-deVersterre, L., Song, M., 2012. Personalized query expansion in the QJC system. In: *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*. ACM, pp. 259–262.
- Verma, N., Malhotra, D., Malhotra, M., Singh, J., 2015. E-commerce website ranking using semantic web mining and neural computing. *Procedia Comput. Sci.*, Elsevier, 42–51.
- Verma, N., Singh, J., 2017a. An intelligent approach to big data analytics for sustainable retail environment using Apriori-MapReduce framework. *Ind Manage Data Syst.*, Emerald 117 (7), 1503–1520.
- Verma, N., Singh, J., 2017b. A comprehensive review from sequential association computing to Hadoop MapReduce parallel computing in a retail scenario. *J. Manage Anal.*, Taylor and Francis. <https://doi.org/10.1080/23270012.2017.1373261>.
- Wang, S., Tan, Y., 2018. Efficient algorithms for finding approximate heavy hitters in personalized page ranks. In: *Proceedings of the 2018 International Conference on Management of Data*. ACM, pp. 1113–1127.
- Wasil, M., Kant, V., 2015. A particle swarm approach to collaborative filtering based recommender systems through fuzzy features. *Procedia Comput. Sci.*, Elsevier 54, 440–448.
- Xiang, B., Jiang, D., Pei, J., Sun, X., Chen, F., Li, H., 2010. Context-aware ranking in web search. In: *Proceedings of the 33rd International ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 451–458.
- Zhou, D., Zhao, W., Wu, X., Lawless, S., Liu, J., 2018. An iterative method for personalized results adaptation in cross-language search. *Inf. Sci.*, Elsevier 430, 200–215.

IMSS: A Novel Approach to Design of Adaptive Search System Using Second Generation Big Data Analytics

Dheeraj Malhotra and O.P. Rishi

Abstract In this present era of Big Data, different search engine users have different information requirements at different intervals of time. Thus, search results should be adapted to user's requirements [1, 2]. In this research work, we propose a novel approach to adaptive web search augmented with capabilities of carrying out Big Data Analytics using second generation HDFS. Moreover, unlike conventional personalization techniques, the proposed approach does not require additional efforts from user such as reporting feedback/ratings etc. The proposed system can be implemented in the form of Intelligent Meta Search System (IMSS Tool) to overcome the problem of irrelevant web page retrieval faced by user of generic search engines. An extensive experimental evaluation shows that the average ranking precision of adaptive IMSS tool improves with trial runs when compared with a popular search engine.

Keywords Second generation HDFS · Personalized search · Big data search system · Meta search engine · Intelligent meta search system (IMSS) tool · Adaptive web search

1 Introduction

Adaptive search when supported by HDFS-Cloud framework leads to easy and efficient analysis of Big Data available on WWW to retrieve useful personalized page ranking patterns. Search engines are known to retrieve far larger information

D. Malhotra (✉) · O.P. Rishi
Department of CSI, University of Kota, Kota, Rajasthan 324005, India
e-mail: Dheerajmalhotra@gmail.com

O.P. Rishi
e-mail: Omprakashrishi@yahoo.com

© Springer Nature Singapore Pte Ltd. 2017
N. Modi et al. (eds.), *Proceedings of International Conference on Communication and Networks*, Advances in Intelligent Systems and Computing 508,
DOI 10.1007/978-981-10-2750-5_20

but still no search engine can index more than about 16 % of index able web [3, 4]. The issue is not just only the volume but is also the relevancy with respect to user's information needs [1, 2]. When the same query is searched by different users, even a state of art search engine returns the same result, irrespective of the user submitting the query. For example, if a user is tech savvy and usually searches for laptop/mobiles then an incomplete query search like *Blackberry* should return documents related to *Blackberry mobiles* by intermediately expanding the query rather than returning the documents of some fruit. There are various types of conventional personalized search systems as discussed in literature. However these search systems fail to satisfy the user personalized requirements without having explicit ratings/feedback from user. Moreover such systems can't handle second generation Big Data as they not just require scalability, partial failure support etc. but also need to support multiple analytic methods on varied data types, as well as the ability to respond in near real time.

2 Contribution from the Study

To the best of our knowledge, this proposed research work is the first formal attempt to design and development of adaptive search system using intelligent big data analytics and is also deployable on cloud framework. Various contributions of the proposed approach may be summarized as follows:

- The user effort for providing explicit ratings/feedback in order to use personalized search system will no longer be required.
- The proposed system will overcome the limitations of traditional mining approaches to extract useful web search and page ranking patterns from Big Databases of Search engines by providing features like Scalability, Partial Failure Support etc.
- The proposed research work discusses the design of future ready intelligent search tool i.e. *IMSS* which can well satisfy the requirements of next generation Big Data Search System such as Real time response, support of multiple analytic engines.

3 System Design

The proposed system will follow modular approach as shown in Fig. 1. Here we first accept user search query and expand the same to intermediate query based on user's preferences obtained from his search history [5-7]. Proposed system will build user profile using user's long term and short term preferences derived from browsing history of n days ago and of current day of usage respectively. Meta key word recommender is used to derive Meta keywords of search from extracted

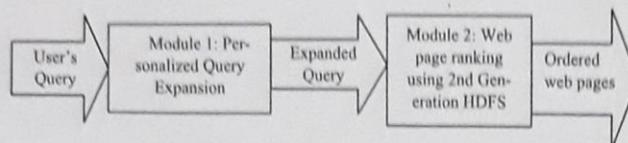


Fig. 1 Simplified design of proposed system

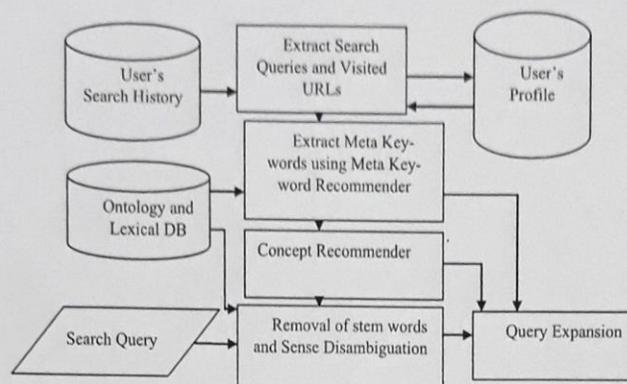


Fig. 2 Design of Module 1—personalized query expansion/modification

URLs. Similarly, Concept Recommender and Word Sense disambiguation processes are used for expanding user query into non ambiguous and more meaningful query as shown in Fig. 2. Module 2 is used for ranking of web pages obtained from backend search engines. HDFS Map() and Reduce() approach is used to calculate content relevancy vector; other relevancy vectors such as semantic relevancy vector (SRV) to determine the semantic closeness of user query with respect to web document under consideration, similarly Time Relevancy Vector is based on importance given by previous user of same web page. The detailed functionality of module 2 to determine weighted rank of candidate web page is shown in Fig. 3.

4 Second Generation HDFS and Map Reduce

There are two significant trends of Second Generation Big data Systems [2, 8] that are responsible for choosing second generation HDFS as a preferable deployment framework in proposed approach. (i) There is rapid growth in network bandwidth as

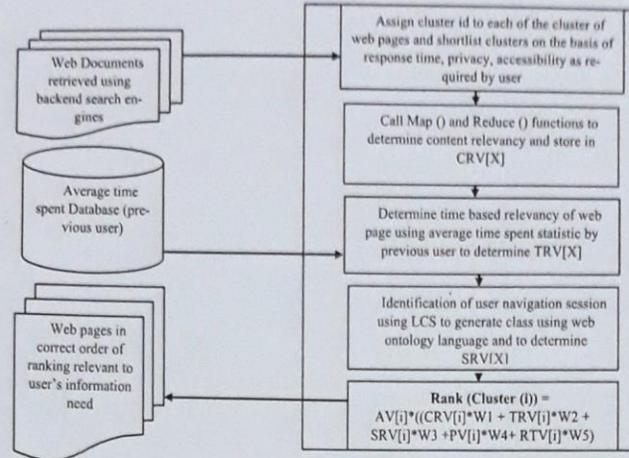


Fig. 3 Design of Module 2—web page ranking using HDFS based cloud framework

compared to hard drive bandwidth (ii) Development of In-memory computation models is urgently required to allow intermediate results to be kept in memory and hence reduces overhead of iterative analytics as suffered by conventional HDFS [9, 10].

HDFS is now adapted as long term store from which applications read their initial data and write their final results. The data layer is divided into sub layers for consistent storage and for intermediate objects separately to handle second generation of Big Data as shown in Fig. 4. In our proposed System, Map function will accept cluster ID as key and cluster log as second argument to tokenize each of web link entry in cluster log, obtained from back end search engine used by IMSS tool, to count individual occurrence of each of the keyword of search query. Extract () function is used to generate elements in list one at a time. Reduce function is coded to aggregate over all the occurrence of each keyword as provided by Map ()

Analytics Engine 1	Analytics Engine 2	Analytics Engine n	Map Reduce	Data Warehouse -SQL	Streaming
Scheduling of Resources			Intermediate & Global Memory Scheduling		
Data Storage			HDFS Data Store		

Fig. 4 HDFS deployment framework for second generation big data system

function [11] to determine keyword frequency in each of the web document and hence to determine the content relevancy vector. Map and Reduce code to be used by **Proposed System** is as follows:

```
Map (Int ID, String Log){
    List<String> X = tokenize (log)
    For each Token in X (// Token - Link extracted
                        //from back end search engine
    Extract ((String) KWL, (Int) 1) // KWL - Keyword list
    })
Reduce (String Token, List <Int> count)
    Int F = 0
    For each word in KWL {
        F = F + 1
    }
    //F- Frequency count of each keyword
    extract((string) token, (Int) F)
```

5 Intelligent Meta Search System

In order to evaluate the proposed research design, *IMSS* tool using HDFS framework for analytics of second generation of Big Data is implemented using ASP.NET framework. The interface of *IMSS* tool is shown in Fig. 5. After Sign In, the inter-face of tool may allow user to select some or all of the four popular search engines like Google, Yahoo, ASK and Bing, for the purpose of intermediate web pages retrieval and search box allow user to specify search string. After clicking the Search button, tool will assign personalized rank to some of the top web links retrieved from back end search engines based on the calculation of various ranking vectors such as AV, SRV, CRV, TRV, RTV. The tool will return web links in the order of their ranking along with statistic of selected advanced search criterion. However *Take Me Fast* tab will not allow selecting any of the search criteria and will give result directly on the basis of user's history of browsing patterns stored in user's contextual database, which could be retrieved using his/her profile.

6 Comparative Precision Analyses—*IMSS* Tool V/S Google

In order to evaluate the effectiveness of our proposed approach, we recruited 10 human volunteers with age varied from 20 to 50 years with minimum of 5 years web search experience. 6 of them were males, 4 were females. They are asked to bring their personal laptops with installed *IMSS* tool followed by initial profile sign

Intelligent Meta Search System			
Create New User Profile	User ID: Dheeraj@UOK	Password: *****	
Select Search Engine Tabs for Intermediate Document Retrieval			
GOOGLE	YAHOO	BING	ASK
<i>Take Me Fast</i> (Personalized Search)		<u>Advanced Search</u> (Select Criteria)	
Response Time	Loading	Security	Page Freshness
<input type="text" value="Enter Search String: HDFS and Map Reduce"/>			
Search		Reset	
Rank	Web Links	Security	Response
1	https://en.wikipedia.org/wiki/Apache_Hadoop	https:	00:00:00:10ms
2	www.gt.ibm.org/software/datacom/infosphere/hadoop/mapreduce	SSL	00:00:00:33ms

Fig. 5 Interface of IMSS tool

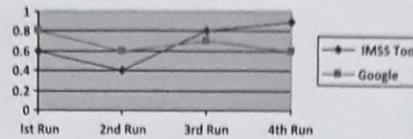
up process on tool, we followed following steps and asked volunteers to repeat the process for at least 4 trial runs one by one on Tool and Google:

1. In the first step, we asked volunteers to search an intentional incomplete query, for example a query like *Black Berry* rather than *Black Berry Mobiles* or *Black Berry Fruit*.
2. In the second step we asked volunteers to give points from 0(worst) to 5(best) to various precision parameters such as personalized page relevancy, page freshness, page size and response time to the top 10 links with respect to their shown rank in output of IMSS and Google.
3. After collecting data from each of the volunteer, we normalized the value of various precision parameters using expression:

$$Q_{\{ab\}} = (\text{HIG}(P_{\{ab\}}) - P_{\{ab\}}) / (\text{HIG}(P_{\{ab\}}) - \text{LOW}(P_{\{ab\}}))$$

where, P_{ab} = Value of b_{th} Parameter of a_{th} web page; Q_{ab} = Normalized value of b_{th} Parameter of a_{th} web page; LOW, HIG = Lowest and Highest value of each of the parameter of precision.

Fig. 6 Personalized precision comparison of IMSS tool with Google for Query "BlackBerry"



- In the next step, we calculated the overall weighted precision of each web page retrieved by each volunteer as $N_a = \sum W_b \cdot Q_{ab}$, where, N_a = weighted precision of a_{th} web page; W_b = Weight assigned to b_{th} parameter by volunteer, usually $0 \leq W_b \leq 1$
- Finally we determined overall precision by calculating average of all the weighted precisions as obtained from volunteers, Precision = AVG (N_a).

6.1 Observation

The graphical analysis in Fig. 6 shows that during first trial Run, precision of Google is reported as high; however with increase in number of trial runs, average precision of Tool improves slowly over Google. This is due to the fact that that Tool will build user profile and by employing personalized search can better satisfy the user for incomplete or ambiguous queries; On the other side, generic search engines try to interpret the query with all possible meanings without considering the preferences of user who searched for query and hence fails to achieve high value of personalized search precision.

7 Conclusion and Future Work

This research work present a HDFS based adaptive search framework for analytics of second generation of Big Data through implementation of IMSS Tool. The effectiveness of proposed approach is justified by experimental evaluation and comparison of personalized precision of IMSS tool over Google. The proposed approach can be applied to retail transactional or E Commerce website database as such transactional databases are also growing in the scale of Terabytes on daily basis and hence they require second generation Big data analytics system to mine useful customer buying patterns rather than conventional data mining techniques. The proposed system design can be enhanced by incorporating other advanced technologies such as Back Propagation Neural Networks, SVM etc. to further improve the precision of tool.

References

1. Wasid, M., Kant, V.: A Particle Swarm Approach to Collaborative Filtering based Recommender Systems through Fuzzy Features. In: *Procedia Computer Science, IMCIP*, Vol. 54, pp. 440–448, Science Direct, Elsevier, Bangalore, India, August 21–23 (2015).
2. Gebara, F., Hofstee, H., Nowka, K.: *Second Generation Big Data Systems*, pp. 36–41, Cover Feature Outlook, IEEE Computer Society (2015).
3. Shou, G., Bai, H., Chan, k., Chen, G.: Supporting privacy protection in personalized web search. In: *IEEE transactions on knowledge and data engineering*, Vol. 26, No 2, pp. 453–467. IEEE (2014).
4. Kuppusamy, K.S., Aghila, G.: CaSePer: An Efficient Model for Personalized Web Page Change Detection Based on Segmentation. Vol. 26, pp. 19–27, *Journal of King Saud University*, Elsevier (2013).
5. Verma, N., Malhotra, D., Malhotra, M., Singh, J.: E-commerce website ranking using semantic web mining and neural computing. In: *International Conference on Advanced Computing Technologies and Applications*, Elsevier *Procedia Computer Science*, Vol. 45, pp. 42–51. Elsevier, Mumbai, India, March 26–27 (2015).
6. Malhotra, D.: Intelligent Web Mining to Ameliorate Web Page Rank using Back Propagation Neural Network. In: *5th International Conference, Confluence: The Next generation information Technology Summit*, pp. 77–81, IEEE Xplore, UP, India, September 25–26 (2014).
7. Malhotra, D., Verma, N.: An ingenious Pattern Matching Approach to Ameliorate Web Page Rank. Vol. 65, No 24, pp. 33–39, *International Journal of Computer Applications*, FCS, New York, USA (2013).
8. Khurana, A.: Bringing Big Data Systems to the Cloud. pp. 72–75, *What's trending? Column*, IEEE Computer Society (2014).
9. Tesai, C., Lai, C., Chao, H., Vasilakos, A.: Big Data Analytics: A Survey. 2:21, pp. 1–32, *Journal of Big Data*, SPRINGER (2015).
10. Singh, A., Velez, H.: Hierarchical Multi-Log Cloud-Based Search Engine. In: *8th IEEE International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 212–219. IEEE CPS, Birmingham, UK, July 2–4 (2014).
11. Son, J., Ryu, H., Yi, S., Chung, Y.: SSFile: A novel column-store for efficient data analysis in Hadoop-based distributed systems, Vol. 316, pp. 68–86. *Elsevier Information Sciences*, September 20 (2015).

IMSS-E: An Intelligent Approach to Design of Adaptive Meta Search System for E Commerce Website Ranking

Dheeraj Malhotra
University of Kota
Kota, Rajasthan, India
+91-9560375531
dheerajmalhotra@ymail.com

O.P. Rishi
University of Kota
Kota, Rajasthan, India
+91-9414258030
omprakashrishi@yahoo.com

ABSTRACT

With the continuous increase in frequent E Commerce users, online businesses must have more customer friendly websites to better satisfy the personalized requirements of online customer and hence improve their market share over competition; Different customers have different purchase requirements at different intervals of time and hence new strategies are often required to be deployed by online retailers in order to identify the current purchase requirements of customer. In this research work, we propose design of a tool called Intelligent Meta Search System for E-commerce (IMSS-E), which can be used to blend benefits of Apriori based Map Reduce framework supported by Intelligent technologies like back propagation neural network and semantic web with B2C E-commerce to assist the online user to easily search and rank various E Commerce websites which can better satisfy his/her personalized online purchase requirement. An extensive experimental evaluation shows that IMSS-E can better satisfy the personalized search requirements of E Commerce users than conventional meta search engines.

Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; H.3.4 [Information Storage and Retrieval]: Systems and Software—*Performance evaluation (effectiveness)*; I.2.6 Learning: Connectionism and Neural Nets.

General Terms

Algorithm, Performance, Experimentation

Keywords

E-commerce personalized website ranking, Intelligent Meta Search Engine, Map Reduce and Apriori, back propagation neural network, IMSS- E, adaptive search.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

AICTE '16, August 12-13, 2016, Bikaner, India
© 2016 ACM. ISBN 978-1-4503-4213-1/16/08. \$15.00
DOI: <http://dx.doi.org/10.1145/2979779.2979782>

1. INTRODUCTION

In India, Ecommerce is the fastest growing sector. According to Forrester Research Inc. [10] on Online Retail 2011 to 2016, Electronic commerce market in India has recorded magnificent growth of more than 400 % in last four years and is expected to grow more rapidly i.e. almost five-fold by December 2016. As a result, companies are not taking chance to satisfy the personalized purchase requirements of customers and hence to fetch good revenues and branding of their business.

As we all know that WWW is dynamic in content and huge in size .Moreover web doesn't possess any catalogue like feature, so, most of E Commerce users are dependent on search engines like Google, ASK, Yahoo etc. to search for relevant E Commerce website for online purchase of a specific product. As discussed in literature none of the search engine can index more than 16% of web [9]. The issue is not just the volume but is also the relevancy with respect to customer requirements, if the query is incomplete or ambiguous then search engines return large number of links in the search output as they tend to return links by interpreting all possible meanings of query for e.g. a customer searching incomplete query like "Samsung" is not always interested in buying a mobile phone as company also deal with other popular electronic products, so search system should intermediately expand the query to complete/ non ambiguous query so as to adapt to current requirements of customer and rank output links correctly by keeping track of customer's long term/short term requirements. There are various types of conventional personalized search systems as discussed in literature time to time however they are not efficient enough to satisfy the requisites for processing modern generation of Big Data available on WWW. In this research work, system design of an Intelligent E Commerce website ranking tool i.e. IMSS-E is proposed. This tool can calculate the relevant rank of various E Commerce websites to better satisfy the personalized purchase requirements of E Commerce customer. The remaining part of the paper is organized as follows. In Section 2, we discuss various types of conventional search systems category wise under Literature review. Section 3 discusses contribution from the proposed study. Section 4, 5, 6 discusses detailed system design, map reduce algorithms for parallel apriori mining and association rules extraction using web dictionary approach. Section 7, 8, 9 discusses interface of proposed IMSS-E tool, its comparison with professional and popular meta search engines and experimental evaluation with graphical analysis. Section 10 concludes the paper with a discussion on plans for future work followed by some of the important references.

2. LITERATURE REVIEW

Adaptive search when supported by intelligent technologies in a cloud framework leads to easy & efficient analysis of Big Data available on WWW to retrieve useful E Commerce website ranking patterns. There are various types of conventional personalized search systems:

2.1 Review of Search Systems based on Link Personalization

E Commerce applications generally use link personalization to assist the customer by recommending links that are more relevant to user based on implicit feedback obtained through buying pattern history and explicit feedback obtained through ratings. It is usually assumed that users who gave similar ratings to similar products have similar preferences and accordingly site recommends links to user that are most popular in his category/cluster of users. E Commerce websites like Flipkart, E Bay etc. follow the implicit and explicit feedback to determine the relevant recommendations for their customers. Dheeraj Malhotra et al. [1] proposed semantic and neural based SNEC algorithm and E Commerce website priority tool for better evaluation of product search queries and to obtain correct ranking of links. The proposed tool may be used to obtain website competitive rank and hence helps in optimization of the structure of E Commerce website. However the tool may further be personalized if provided with various capabilities such as Page Loading Speed, Security Comparison, and Ease of Navigation etc to rank the websites as required by online customer.

2.2 Review of Search Systems based on Content Personalization

Content personalization on web means to present different content to different customers on same E Commerce web site. Kazunari Sugiyama et al. [2] discussed that web sites/portals like My Yahoo present the information to users in which they may be interested. User may explicitly mention the modules of his choice on such websites that may include sports news, fashion updates, weather etc. Users may create their own page layout as per their requirement on content personalized web sites. However such systems possess limitations like effort from user is required as such systems are essentially dependent on user inputs. Moreover these sites cannot adapt well with change in user needs unless user explicitly changes his previously registered preferences. K.S. Kuppusamy et al. [3] proposed a general purpose personalized web page change detection model "CaSePer" to assist the users who frequently visit a web page and are more interested in knowing the recent changes rather than seeing the entire content of the web page. This model need to be adapted as a personalized search system and also the experimental effectiveness of such a search system is required to be evaluated.

2.3 Review of Search Systems based on Recommender System

In this era of Big Data, we get the overwhelmed feeling of continuously growing information on the web. Recommender systems have been emerged to deal with this problem of 'information overload'. Mohammed Wasid et al. [4] discussed that Recommender Systems (RS) assist users by recommending songs, movies etc. to utilize their free time, E Commerce websites to make purchase decision, financial, matrimonial services and even to whom to date. They proposed particle swarm optimization to find priorities of different users and generates

more accurate and personalized recommendations for users. They discussed four filtering techniques can be employed by RS i.e. collaborative filtering, content based filtering, hybrid filtering and demographic filtering techniques. Panagiotis Adamopoulos [5] proposed improvement of K nearest Neighbors approach i.e. Probabilistic Neighborhood method to overcome the common problems of collaborative filtering. They also implemented the concept of *unexpectedness* in Recommender systems for specifying and satisfying the expectations of user.

2.4 Review of Search Systems based on Cloud and Intelligent Technologies

Ajitpal Singh et al. [7] discussed the design of simple search engine Simha to efficiently search over various cloud platforms for unstructured and structured data using Elastic search engine at backend. They also explained the importance of carefully designed Extraction, Transform and Load processes while indexing large data sets which may otherwise lead to system exceptions. Dheeraj Malhotra [9] well explained that huge size and interference by SEO leads to difficulty in extracting relevant information from web via search engines. However back propagation neural network can well learn from errors by implementation of supervised learning and hence can be trained to improve search engine page ranking process.

3. Contribution from the Study

To the best of our knowledge, as procured from literature, this research work is the first formal attempt to design of adaptive search system using Intelligent Map Reduce framework supported by apriori mining, semantic web and back propagation neural network for E Commerce environment. Various contributions of the research study may be summarized as follows:

- The customer or online retailer effort for providing explicit feedback or ratings in order to build his/her profile and hence to use adaptive search system for ranking of E Commerce websites will no longer be required.
- The proposed research work discusses the design of intelligent meta search system for E Commerce i.e. *IMSS-E*, which can be used to derive useful personalized website ranking patterns from modern Big Databases and is capable to provide partial failure support, scalability, real time response etc as required by next generation of Big Data Ecommerce systems.

4. System Design

The proposed system design consists of following three phases and is shown in Figure 2:

Phase 1 accepts Customer product search query and disambiguate the incomplete or ambiguous query by using his/her profile. It will first remove the stem words like A, An, The etc. followed by query expansion by using various keywords provided by customer during Sign Up/ Registration process and previously made search queries and corresponding visited URLs to determine the customer preferences and current requirements. In this phase meta keywords and ontological concepts are extracted using recommender system to intermediately expand the search query to more meaningful query.

Phase 2 accepts expanded search query from phase 1 and will search the expanded query on number of backend Meta Search Engines such as Dogpile, Mamma, Kartoo and Meta Crawler to retrieve various E Commerce websites. In this phase retrieved websites are shortlisted by using various criteria as mentioned by customer through the interface of IMSS-E tool such as Page Loading Speed, Transaction Security, Page Freshness, Response Time followed by determination of Website Relevancy (WR) in terms of content relevancy by calculating frequency of hits for various association rules generated from keywords of search query using Apriori mining algorithm and Map Reduce framework, Time spent by previous user of same website is also taken into consideration to determine WR, Further Semantic Relevancy (SR) is calculated using Web Ontology Language and Longest Common Subsequence (LCS) to determine proximity between customer search requirement and candidate website.

Phase 3 determine rank of each of the candidate website by using Supervised Back Propagation Neural Network, this phase will first normalize all the inputs from Phase 2 i.e. WR, SR, Bias (B) and assign random weights V1, V2 and W1, W2 to intermediate synapses between input- hidden layer and hidden - output layer respectively followed by training of network using linear activation function, this phase will calculate the error rate and will adjust the weights accordingly in backward stage till the error rate is reduced than tolerance and finally determine the output at hidden layer and output layer using sigmoidal function and summation function respectively.

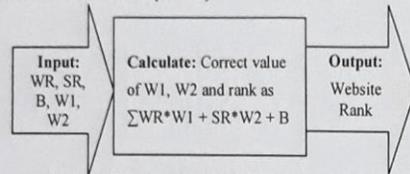


Figure1. Simplified Neural Design of Proposed System

5. Parallel Apriori Mining and Map Reduce

In order to determine content relevancy of an E Commerce web site with respect to customer product search query, we used Parallel Apriori Association Mining with in Map Reduce paradigm to determine the frequency of co-occurrence of various keywords of product search query using web dictionary of candidate E Commerce website, whose rank is to be determined. Apriori is so named as it uses prior knowledge that all non empty subsets of frequent item set must also be frequent [10]. The motivation for using parallel Apriori on map reduce platform is to save resources like CPU time, memory and execution time, which is being wasted in sequential form of mining while scanning Big data available on web. Moreover multiple transactions counting occurrence of candidate item sets can be executed in parallel rather than executing them in sequential order. Map Reduce is a parallel and distributed programming model. The advantage of Map Reduce framework is that it abstracts all the issues related with parallel and distributed processing from programmer e.g. programmer need not to worry about how to split a process into equal sized tasks so that no task may complete its execution before other task, similarly combining of results does not require programmer's attention. Here programmer need to write just the code of two functions i.e. map and reduce which operate on Key, Value pairs.

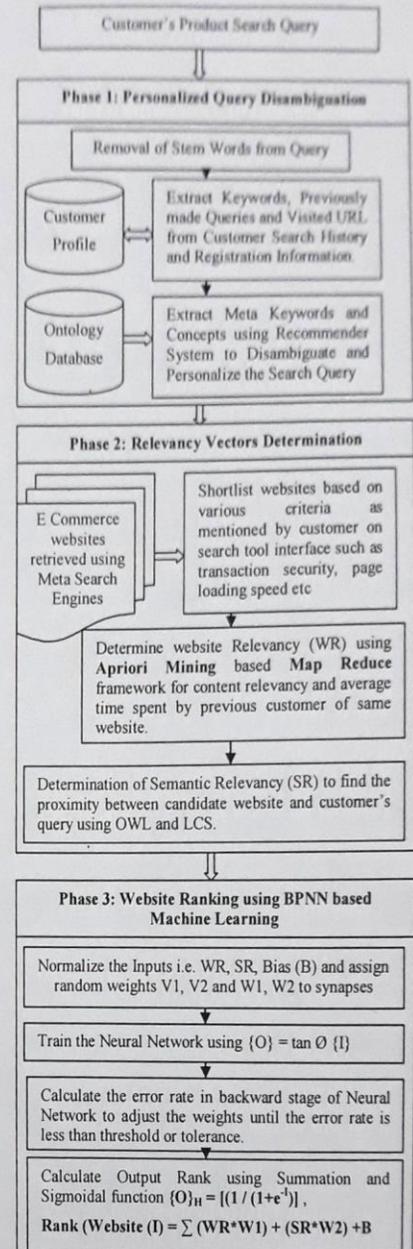


Figure2. System Design

5.1 Map () and Reduce () for Apriori Mining

Map function identifies the candidate item sets of given size C_1 in maximum allowed iterations of Apriori (max), using parallel implementation of Apriori association mining. Reduce function accepts output of map function as input and is used for summation of count occurrences and assist in calculation of frequent item sets L_k . These functions are coded to implement parallel form of Apriori as follows:

```

Map () for Implementation of Parallel Apriori Mining
Input: (Key  $x$  = Key Offset, Value  $x$ )
Output: List (Key  $y$  = Candidate Item set,
              Value  $y$  = 1 for each single key occurrence)

Algorithm:
Map (Key  $x$ , Value  $x$ )
If (max  $\geq$  1)
    For each Candidate Item Set  $C_1$ 
        If  $C_1$  is a subset of Value  $x$ 
            Return ( $C_1$ , 1)
        End If
    End For
End If
End.
    
```

```

Reduce () for Implementation of Parallel Apriori Mining
Input: (Key  $y$  = Candidate Item Set, List (Value  $y$  =
              Individual Occurrence of Key))
Output: (Key  $z$  = Key  $y$ , Value  $z$  = Total Occurrence
              of Key)

Algorithm:
Reduce (Key  $y$ , List (Value  $y$ ))
Set Count = 0
For each Value in Value  $y$ 
    Count = Count + Value  $y$ 
End For
Return (Key, Count)
End.
    
```

6. Association Rules Extraction

Here, we will first implement web dictionary for each candidate E Commerce web site which in turn will assist to determine strong association rules in between various keywords of search query. In order to generate web dictionary, we will first calculate minimum (MIN) and maximum length (MAX) of each keyword of search

query and then by allowing only those words from candidate web page having length in between MIN and MAX to get place in web dictionary. The detailed discussion of web dictionary implementation may be referred in our earlier published work [8]. In order to find frequency of co-occurrence of various keywords of customer's product search query, we will employ apriori association mining discussed in section 5 on the web dictionary of candidate web page, to be ranked. In order to find the suitable rank of candidate web site, we will compare the frequency of hits in web dictionary of each of the candidate E Commerce web site for association rules comprising of all the keywords of search query. Let us take an example of a query "Samsung S7 Mobile". Here, different item sets for analysis are keywords of search query i.e. Samsung, S7 and Mobile, sample frequent item sets L_1 , L_2 and L_3 generated for a candidate web page using web dictionary approach are shown in Table 1, Table 2 and Table 3 respectively.

Table 1 L_1 Item Sets

Item Set	Hits
Samsung	56
S7	35
Mobile	95

Table 2 L_2 Item Sets

Item Set	Hits
Samsung \rightarrow S7	22
S7 \rightarrow Mobile	30
Samsung \rightarrow Mobile	45

Table 3 L_3 Item Sets

Item Set	Hits
Samsung \rightarrow S7 \rightarrow Mobile	15

Similarly, frequent item sets corresponding to each candidate web page would be calculated and Website Relevancy (WR) would be assigned in the order corresponding to number of hits for item set having all the keywords of customer search query.

7. IMSS-E Tool

In order to evaluate the effectiveness of proposed system design, IMSS-E tool i.e. Intelligent Meta Search System for E Commerce is implemented using ASP.NET framework. The interface of IMSS-E tool is shown in Figure 3. The interface of tool allow user to sign up and register his/her basic information such as age, gender, qualification, occupation etc along with optional favorite Keywords to semantically determine his/her preferences while browsing E Commerce websites. After Sign In, the inter-face of tool may allow user to select some or all of the four meta search engines i.e. Dogpile, Mamma, Kartoo and Meta Crawler for the purpose of E Commerce web sites retrieval as tool will work like meta search system and hence does not have its own crawler or web index, search box in the interface of tool will allow customer or online retailer to specify E Commerce specific search string. After clicking the Search button, tool will assign personalized rank to some of the top web sites retrieved from back end meta

search engines based on the rank calculation via implementation of all three phases discussed in system design. The tool will return web links in the order of their ranking along with statistic of selected advanced search criterion while using criteria based search tab. However Adaptive Search tab will not allow selecting any of the search criteria and will give result directly on the basis of customer history of E Commerce website browsing patterns as available in customer profile.

Intelligent Meta Search System - E Commerce			
SIGN UP	USER ID: Cus@EC...	PASSWORD: *****	
Select Meta Search Engine Tabs for Document Retrieval			
Dogpile	Mamma	Kartoo	Meta Crawler
Adaptive Search		Criteria based Search	
Response Time	Page Loading Speed	Transaction Security	Page Freshness
Enter Search String: Online Purchase of Samsung S7 Mobile			
SEARCH		RESET	
Rank	Web Links	Transaction Security	Response Time
1	https://www.samsu...ngabc.com	Https:	00:00:39 ms
2	www.xmobile.com	SSL	00:00:67 ms

Figure3. Interface of IMSS-E Tool

8. IMSS-E V/S Meta Search Engines

In Table 4, we carried out comparison between our proposed tool i.e. IMSS-E with other professional and popular online meta search engines. IMSS-E is capable to perform personalized search as well as advance criteria based search and page ranking on the basis of various criteria such as Response Time, security, page loading speed etc. as selected by user, Personalized search and advance criteria search options are missing in professional meta search engines, however IMSS - E can perform only text based search, multimedia search options such as search for image search, video search etc is not possible by proposed IMSS-E tool.

Table 4 Comparison – IMSS-E and Meta Search Engine

Meta Search Engine	Multimedia Search Options	Advanced Criteria Search	Personalized Search
Dogpile	YES	NO	NO
Mamma	NO	NO	NO
Kartoo	NO	NO	NO
Meta Crawler	YES	YES	NO
IMSS-E	NO	YES	YES

9. Experiment and Graphical Analysis

In order to assess the efficiency of our proposed tool and hence effectiveness of the proposed approach, we recruited 20 volunteers with age varied from 18 years to 40 years with minimum of 3 years experience of carrying out E Commerce transactions. 11 of them were males, 9 were females. They are asked to install IMSS-E tool on their personal laptops and first complete the initial registration process using interface of tool, we asked volunteers to compare the performance of tool with respect to a professional meta search engine, repeating the following process for at least 5 trial runs on each of IMSS-E Tool and a professional meta search engine, Dogpile:

- Firstly, we asked volunteers to search an ambiguous product search query, for example a query like *Samsung purchase* rather than *Samsung Mobiles Purchase*.
- Secondly, we asked volunteers to give points from 0(worst) to 10(best) to various precision parameters of listed links to E Commerce websites such as page relevancy, response time to the top 20 links with respect to their displayed rank in output of IMSS-E and Dogpile.
- In third step, we normalized the value of various precision parameters obtained from volunteers using expression: $Q_{ab} = \frac{(MAX(P_{ab}) - P_{abr})}{(MAX(P_{ab}) - MIN(P_{ab}))}$ Where, P_{ab} = Value of b Parameter of a web page; Q_{ab} = Normalized value of b Parameter of a page MIN, MAX = Minimum and Maximum value
- In the fourth step, overall weighted precision of each web page retrieved is determined as $PRE_{Ca} = \sum w_b \cdot Q_{ab}$ Where, PRE_{Ca} = weighted precision of web page a, w_b = Weight assigned to b parameter by user, where, $0 < w_b \leq 1$

- In the last step, overall precision in E Commerce website ranking calculation is determined by average of all the weighted precisions as obtained in previous step
Precision (IMSS-E / Dogpile) = AVERAGE (PREC_i).

The comparative graphical analysis of proposed tool and a popular meta search engine is shown in Figure 4, initially, precision of Dogpile meta search engine is reported high in first few trial run, however with increase in number of trial runs, average precision of IMSS-E improves slowly over Dogpile. This is mainly due to two reasons (i) Back propagation neural network employed in the tool will adjust the weights correctly using supervised learning by learning from errors in each trial run (ii) Secondly, tool will slowly build user profile and employ it for personalized search to better satisfy the user requirements even for ambiguous queries; On the other side, meta search engine try to interpret the query with all possible meanings without considering the personalized preference of customer interested in online purchase.

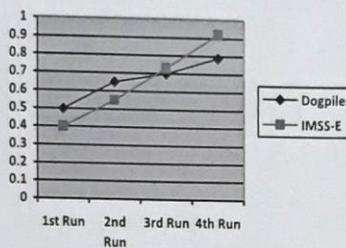


Figure4. Precision Comparison of IMSS-E and Dogpile

10. Conclusion and Future Work

This research work presents an adaptive search tool, IMSS-E, for ranking of E - Commerce websites to assist online customer to find suitable site on top while searching for a specific product as well as to assist online retailer to well structure his website to better satisfy the personalized purchase requirement of customer. The proposed research work utilizes apriori mining - map reduce based Big Data analytics framework, supported by semantic web and back propagation neural network to well adapt to personalized requirements of customer by learning from previous errors in ranking E -Commerce websites. The ranking effectiveness of proposed tool is justified by experimental evaluation, followed by comparison of personalized precision of IMSS-E tool with popular meta search engine, dogpile. The proposed tool can further be enhanced by incorporating other advanced features such as capabilities to search for multimedia content like images, videos etc, managing previous customer reviews and ratings etc to make agent recommendations to further assist the customer in easily making online purchase decision.

11. REFERENCES

- [1] Verma, N., Malhotra, D., Malhotra, M. and Singh, J. 2015. E-commerce website ranking using semantic web mining and neural computing. In *Proceedings of International Conference on Advanced Computing Technologies and Applications* (Mumbai, India, March 26-27, 2015). *Procedia Computer Science*, Science Direct. Elsevier, Vol. 45, 42-51. DOI = 10.1016/j.procs.2015.03.080
- [2] Sugiyama, K. Hatano, K. and Yoshikawa, M. 2004. Adaptive Web Search Based on User Profile Constructed without any Effort from Users. In *Proceedings of the 13th international conference on World Wide Web* (New York, USA, May 17-22, 2004), ACM, 675-684. DOI= 10.1145/988672.988764
- [3] Kuppusamy, K.S. and Aghila, G. 2014. CaSePer: An Efficient Model for Personalized Web Page Change Detection Based on Segmentation. *Journal of King Saud University*, Vol.26 (January 2014), Elsevier, 19-27. DOI = 10.1016/j.jksuci.2013.02.001
- [4] Wasid, M. and Kant, V. 2015. A Particle Swarm Approach to Collaborative Filtering based Recommender Systems through Fuzzy Features. In *Proceedings of Eleventh International Multi-Conference on Information Processing* (Bangalore, India, August 21-23, 2015). Science Direct. Elsevier, Vol.54, 440-448. DOI = 10.1016/j.procs.2015.06.051
- [5] Adamopoulos, P. 2014. On Discovering Non Obvious Recommendations: Using Unexpectedness and Neighborhood Selection Methods in Collaborative Filtering Systems. In *Proceedings of the 7th ACM international conference on Web search and data mining* (New York, USA, February 24-28, 2014), ACM, 655-665. DOI = 10.1145/2556195.2556204
- [6] Shou, G., Bai, H., Chan, k. and Chen, G. 2014. Supporting privacy protection in personalized web search. *IEEE transactions on knowledge and data engineering*, Vol. 26, (February 2014), 453-467. DOI=10.1109/TKDE.2012.201
- [7] Singh, A. and Velez, H. 2014. Hierarchical Multi-Log Cloud-Based Search Engine. In *Proceedings of International Conference on Complex, Intelligent and Software Intensive Systems*, (Birmingham, July 2-4, 2014), IEEE CPS, 212-219. DOI = 10.1109/CISIS.2014.30
- [8] Malhotra, D. and Verma, N. 2013. An Ingenious Pattern Matching Approach to Ameliorate Web Page Rank. *International Journal of Computer Applications*, Vol. 65, No 24, (New York, USA, 2013), FCS, 33-39. DOI = 10.5120/11235-6543
- [9] Malhotra, D. 2014. Intelligent Web Mining to Ameliorate Web Page Rank using Back Propagation Neural Network. In *Proceedings of 5th International Conference, Confluence: The Next generation information Technology Summit*, (Noida, India, September 25-26, 2014), IEEE, 77-81. DOI = 10.1109/CONFLUENCE.2014.6949254
- [10] Li, N. Zeng, L. He, Q. and Shi, Z. 2012. Parallel Implementation of Apriori Algorithm based on Map Reduce. In *Proceedings of 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, (Kyoto, Japan, August 8-10, 2012), IEEE, 236- 241, DOI 10.1109/SNPD.2012.31