

Question and Accepted Answer Recommendation System with a Generic Knowledge Mining Visualization Tool



Shariful Islam Arafat - 011132012

Md. Shakil Hossain - 011132017

SM Al Hossain Imam - 011132125

Mahmudul Hasan - 011132078

Department of Computer Science and Engineering

United International University

A thesis submitted for the degree of
BSc in Computer Science & Engineering

January 2018

Declaration

We, Sk Md. Shariful Islam Arafat, Md. Shakil Hossain, SM AL Hossain Imam and Mahmudul Hasan, declare that this thesis titled, Question and Accepted Answer Recommendation System with a Generic Knowledge Mining Visualization Tool and the work presented in it are our own. We confirm that:

- This work was done wholly or mainly while in candidature for a BSc degree at United International University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at United International University or any other institution, this has been clearly stated.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely our own work.
- We have acknowledged all main sources of help.
- Where the thesis is based on work done by ourself jointly with others, we have made clear exactly what was done by others and what we have contributed myself.

Signed and Date: _____

Signed and Date: _____

(Sk Md. Shariful Islam Arafat)

(Md Shakil Hossain)

Signed and Date: _____

Signed and Date: _____

(SM AL Hossain Imam)

(Mahmudul Hasan)

Certificate

I do hereby declare that the research works embodied in this thesis entitled Question and Accepted Answer Recommendation System with a Generic Knowledge Mining Visualization Tool is the outcome of an original work carried out by Sk. Md. Shariful Islam Arafat, Md. Shakil Hossain, SM Al Hossian Imam and Mahmudul Hasan under my supervision.

I further certify that the dissertation meets the requirements and the standard for the degree of BSc in Computer Science and Engineering.

Signed: _____

Date: _____

(Md. Mofijul Islam)
Department of Computer Science and Engineering,
University of Dhaka,
Dhaka-1000, Bangladesh.

Abstract

With the increasing software developer community, questions answering (QA) sites, such as StackOverflow, Quora, Yahoo Answer etc. have been gaining its popularity. Hence, in recent years, millions of users are posting in StackOverflow. With the increase of users their question/answer a copious amount of data is generated, which attracts both the research community to utilize this information for extracting knowledge and the industry for developing the knowledge based systems.

Since it is really difficult to find a post that relates ones expertise area in order to gain some extra reputation or bounty points, we have built a personalized recommender system named **PRISM** that will recommend posts to a user after login based on their expertise. We have also implemented our recommendation system web application which is currently deployed in our localhost.

Again it takes an enormous amount of effort to find out the suitable answer of a question. Propitiously, StackOverflow allows their community members to label an answer as an accepted answer. However, in the most of the questions answers are not marked as accepted answers. Therefore, there is a need to build a recommender system which can accurately suggest the most suitable answer to the questions. Contrary to the existing systems, in this work, we have utilized the textual features of the answers comments with the other metadata of the answers to build a recommender system for predicting the accepted answer called **RAiTA**. We have also deployed our recommendation system web application, which is publicly accessible at <http://210.4.73.237:8888/>

Visualization of data, pattern mining from datasets and analyzing data drift for the different features are three highly used applications of machine learning and data science fields. A generic web-based tool integrated with such features will provide prodigious support for pre-processing the dataset and thus extracting accurate information. In this work, we propose such a data visualization tool, named **VIM** which is publicly accessible at <http://210.4.73.237:9999/>

In memory of our family, friends and our parents.

Published Papers

Work relating to the research presented in this thesis has been published/ submitted by the author in the following peer-reviewed journals and conferences:

1. **Md. Mofijul Islam**, Sk. Shariful Islam Arafat, Md Shakil Hossain, Md Mahmudur Rahman, S M Al-Hossain Imam, Md. Mahmudul Hasan, Swakkhar Shatabda, Tamanna Islam Juthi, **"RAiTA: Recommending Accepted Answer using Textual Metadata"**, accepted in Springer IEMIS-2017. Application
2. **Md Shakil Hossain**, Sk. Shariful Islam Arafat, S M Al-Hossain Imam, Md. Mahmudul Hasan, Md. Mofijul Islam, Sanjay Saha, Swakkhar Shatabda, Tamanna Islam Juthi, **"VIM: A Big Data Analytics Tool for Data Visualization and Knowledge Mining"**, IEEE WIECON-ECE-2017. Application

Acknowledgements

First and foremost would like to thank the almighty Allah, without his blessings it would never have been possible to complete our thesis successfully.

We are really grateful to our supervisor **Md. Mofijul Islam**, Lecturer, Department of Computer Science and Engineering, University of Dhaka. His imparting knowledge and expertise, consistent guidance and ample time spent has helped us in bringing our study into success.

We are also thankful to **Dr. Swakkhar Shatabda**, Asst. Professor and Undergraduate Program Coordinator, Department of CSE, United International University, **Sanjay Saha**, Assistant Professor, Department of CSE, University of Asia Pacific and **Md. Mahmudur Rahman**, Lecturer, Department of CSE, United International University for advising us.

Last but not the least, none of this could have possible without the support of our beloved family.

Contents

List of Figures	x
List of Tables	xi
1 Introduction	1
1.1 Question answer community	1
1.1.1 Stack Exchange	1
1.1.2 Yahoo! Answers	2
1.1.3 Quora	3
1.2 Motivation	3
1.3 Challenge	3
1.4 Proposed System	4
1.4.1 Question Recommender	4
1.4.2 Accept Answer Recommendation System	4
1.4.3 Data Visualization Tool	5
1.5 Organization of the Thesis	5
1.6 Summary	5
2 Related Work	6
2.1 Related work of Question Recommendation System	6
2.2 Related work of Accepted Answer Prediction System	7
2.3 Related work of A Big Data Analytics Tool for Visualization and Knowledge Mining	8
3 Question Recommendation System	11
3.1 Question Recommender for Expert (PRISM)	11
3.1.1 Feature Selection	11
3.1.2 Feature Finalization	12
3.1.3 Feature List	12
3.1.4 Expert User Prediction Model	13
3.1.5 Recommender System	13
3.2 Application Implementation	14
3.2.1 Application System Architecture	14

3.2.2	Web Application	15
4	Accepted Answer Prediction System	17
4.1	Recommending Accepted Answer using Textual Metadata (RAiTA) . .	17
4.1.1	Feature Selection	17
4.1.2	Feature Finalization	18
4.1.3	Feature List	18
4.2	Application Implementation	19
4.2.1	Application System Architecture	19
4.2.2	Web Application	20
5	A Big Data Analytics Tool for Visualization and Knowledge Mining	22
5.1	Big Data Visualization and Knowledge Mining (VIM)	22
5.1.1	Key Features	22
5.1.2	Application System Architecture	23
5.1.3	Web Application	24
6	Performance and Evaluation	29
6.1	Performance evaluation of PRISM	29
6.1.1	Dataset Description	29
6.1.2	Performance Evaluation Criteria	30
6.1.3	Classifier and Feature Selection and Evaluation	30
6.2	Performance evaluation of RAiTA	31
6.2.1	Dataset Description	31
6.2.2	Performance Evaluation Criteria	32
6.2.3	Effectiveness of Sentiments in Accepted Answer	33
6.2.4	Classifier and Feature Selection and Evaluation	33
7	Conclusion	36
7.1	Summary of Research	36
7.1.1	Question Recommendation System	36
7.1.2	Recommending Accepted Answer using Textual Metadata	36
7.1.3	Recommending Accepted Answer using Textual Metadata	37
7.2	Discussion	37
7.3	Future Works	37
	Bibliography	39

List of Figures

1.1	Question by hour	2
1.2	Answer by hour	2
3.1	Application System Architecture of PRISM	15
3.2	Home Page of PRISM	16
3.3	Recommended Questions	16
4.1	Application System Architecture of RAIiTA	20
4.2	Home Page of RAIiTA	21
4.3	Recommended Answers of RAIiTA	21
5.1	Application System Architecture of VIM	23
5.2	Flow Chart of VIM	24
5.3	Home Page of VIM	25
5.4	Visualization for Data Distribution of (ProgrammingHobby) Feature . .	26
5.5	Feature Extraction of Some Selected Features	26
5.6	Frequently Mined Pattern Rule Set	27
5.7	Drift Analysis of VIM for the Particular Dataset	28
6.1	MCC Equation	32
6.2	ROC Curve	33
6.3	ROC curves for different classifier	35

List of Tables

3.1	Feature List of Question Recommendation System	13
3.2	System Specification of Our Web Applications	14
4.1	Feature List of Accepted Answer Prediction	19
4.2	System Specification of Our Web Applications	19
5.1	System Specification of VIM	25
6.1	Dataset Details	30
6.2	Results of Different Classifier Analysis for PRISM	31
6.3	Dataset Details	31
6.4	Results of the Sentimental Analysis	33
6.5	Results of Different Classifier Analysis for RAiTA	34
6.6	Results of Different No. of Features for RAiTA	34

Chapter 1

Introduction

Internet has made this world a village to ease the communication and sharing knowledge and information with one another. As a consequence, question answering (QA) sites are nowadays growing their popularity rapidly. A number of question answering sites, such as StackOverflow, Quora, Yahoo Answer, have already established their reputation up to a limit that people are engaging more and more on those sites to get help with their questions' answer.

1.1 Question answer community

Question answer community is such a platform where anyone can ask questions about their problems and experts can give a solution to that particular question based on their expertise. There are lots of user coming with their problems and experts are always ready to help them with a good solution. A huge number of the user submitting questions and answers every day. Among users and experts, a good relation and community have been creating day by day. This kind of community is very helpful for responding to question for particular industries or sector. These communities also helpful for specialists for exchanging their answer and thought on a regular basis. There are lots of question-answer community websites. Example: Stack Exchange, Yahoo! Answers, Quora etc.

1.1.1 Stack Exchange

Stack Exchange is created to build a network of question and answer websites based on the various field. Each site is covering a specific topic where the user can ask question and answer and earn reputation award. Then these sites are converted as stack overflow, a question-answer site based on computer programming.

Stack Overflow was created in 2008 for an open alternative of a question and answer sites. It was created to be an expert-exchange. The main feature of this site is for questions and answers of a wide range of topics in computer programming. Here one

can ask a question about programming related problems. The user can ask question and expert will give a solution. The user can up-vote or down-vote both question and answer. They can comment on question and answer and edit the text of others. In this process, the user can earn reputation point and badges. So, in web search and career jobs, search results users stack overflow profile can get huge benefits. All users granted posts is copyright under stack exchange network.

Some Question and Answer Statistics:

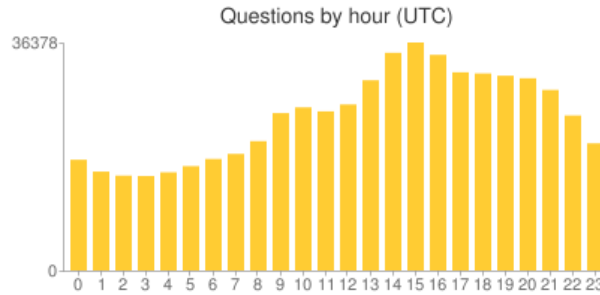


Figure 1.1: Question by hour - UTC

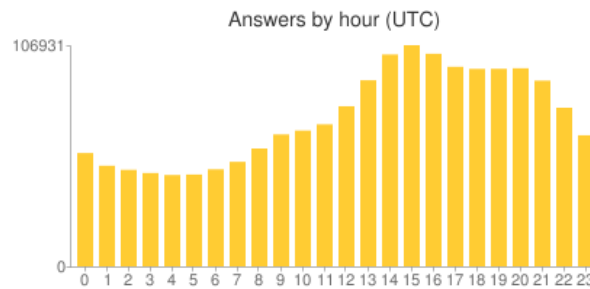


Figure 1.2: Answer by hour - UTC

There have almost 15,256,987 questions been posted yet in Stack Overflow. There are lots of websites in stack exchange network. Stack Overflow, Super User, Meta Stack Exchange, Ask Different, Ask Ubuntu, English language Exchange etc.

1.1.2 Yahoo! Answers

Yahoo! Answers is a question-answer community website. It is a knowledge market form Yahoo. Here one can ask a question that's not violet Yahoo. Questions are open for four days to answer and asker can choose the best answer. Later comment and answer can be also posted. Users need Yahoo ID with a positive score to post an answer. This is good Q & A sites for knowledge sharing. There are a level and point system for users. There is a limitation per day. The user begins in level 0 and answering 1 question they are promoted to level 1 and also get 100 free points. There is a number of limitation for every level per day for question and answer. Before 2012

user more than level 5 could give unlimited question and answer. Later on there made a limitation for every level.

1.1.3 Quora

Quora is a good Q & A site. A user can visit the website with or without registration. A user can register with the real name but name verification is not required. But the false name can be reported by community users. Users can upvote, downvote, request for extend answer or edit of other users answers. Quora is a very popular question-answer site. There have almost 14,304,529 questions been posted. Any kind of query people can ask there. Recently quora have been a useful resource for career and job. If we connect quora with our social media we can connect with people we follow. So we can share our thoughts and get feedback from the experts. Upvote and downvote help to find the best response answers so that user easily can get there best answer what they are searching for. This is the very popular site for any kind of queries nowadays. If there is any wrong information it can be reported and removed. So this is resources are trusted also.

1.2 Motivation

In these all question answer websites, a user can ask questions or answer any existing question accordingly. Moreover, they can also redirect questions to the expert users. As a result, a question may have different answers from different persons. An expert is not finding easily the question of their experts categories. They have to search for question randomly. So, sometimes all users question is not going to proper experts all the time. It would be very helpful to users and experts if there is an easier way to get the question to experts easily based on their expertise. Then they will get comfort to give an answer to the particular question based on their expertise. We feel there should be a recommender system to suggest expert questions based on their expertise.

Sometimes users submit the question and answer but they are not accepted. It would be better if the user can check the probability of their question and answer are being accepted. So, we feel there should be a tool for checking if the answer he/she will submit will be accepted. If the users get the probability of acceptance of their answer before submitting this will be very helpful for users.

When the users need the data set there are lots of unused data and its very tuff to find the features they actually need. So we feel there should be a data visualization tool for extract features they actually need.

To make this thing easier we try to come up with some solutions and tools.

1.3 Challenge

The main challenge to work on this question-answer system is collecting data. There were many scopes to work on these sites but fewer people work on these sites before. So

resource is very low. We had to find data before which is not available much and pre-process them to work on it. Then we train our system on these datasets and try to make solutions to make easier these question answer websites usage. In the dataset, there were lots of garbage values. We had to filter the dataset which was a challenging issue from lots of garbage values. We had to feature extraction from more than thousands of features. First, we select 50 features to do our operation. With 50 feature in our web tool, it was very time consuming and slow. So we select best 20 features. In our accept answer recommendation system, we use natural language processing. In the dataset, there were questions answer and also answers answer. So it was difficult to get all the answers. In question recommender system we had to find experts to recommend questions. This was a challenging part to find experts and correlation.

1.4 Proposed System

To overcome this challenges we propose three system:

- Question Recommender
- Accept Answer Recommendation System
- Data Visualization Tool

1.4.1 Question Recommender

The system we have built is called Personalized Question Recommender for Expert User. We have utilized the user profile data and Question answer related metadata to predict whether the user is expert or not. From our dataset, we filter the garbage values. Then extract the feature from the dataset. First, we select 50 feature from the dataset. With 50 feature in our web tool, it was very time consuming and slow. So we select best 20 features. Based on that result we trained a recommender system that can recommend questions to any users with their above-mentioned data. So the expert user can easily find users question based on their expertise and give solution or answer. It will be very useable and easy to the community.

1.4.2 Accept Answer Recommendation System

The system we have built is called Recommending Accepted Answer using Textual Metadata. We have utilized the textual features of the answers comments with the other metadata of the answers. In our accept answer recommendation system, we use natural language processing. In the dataset, there were questions answer and also answers answer. We process this data. This tool gives the probability of acceptable answer. So, the user can predict if their answer will be accepted or not and submit them by correcting the answer.

1.4.3 Data Visualization Tool

The system we have built is called A Big Data Analytics Tool for Data Visualization and Knowledge Mining. This system is suitable for any dataset to visualize feature distribution & pattern mining feature extraction. The major contributions of this work are building a generic and scalable data analysis tool which has a user-friendly UI (User Interface). In the dataset, there can be thousands of feature. The user can select feature visually which feature they actually need and extract them for their use. Given a set of selected features from a dataset, the user can get association rules to extract information. Users can analyse the drift of data among multiple features.

1.5 Organization of the Thesis

The rest of the work is organized as follow: Chapter 2 shows the existing related works. In Chapter 3 we present Question Recommendation System for Expert. In Chapter 4 we present the proposed accepted answer recommendation system methodologies. A Big Data Analytics Tool for Visualization and Knowledge Mining has been discussed in Chapter 5. Performance and Evaluation of the system as well as the deployed web application has been presented in Chapter 6. Finally, we conclude our work and present the future steps in Chapter 7.

1.6 Summary

There is a great community in question answer websites. A huge number of users submitting questions and answers every day. Experts are submitting their answer on others problem based on their expertise. But there is no recommendation system to suggest an expert find question and answer them based on their expertise. The user doesn't have any answer acceptance probability check recommendation system also. So, many of their answers are not accepted. So, we come up with question recommender system, accept answer recommender system and data visualization tool to make easy to question and answer to the community.

Chapter 2

Related Work

Recommending best suitable question to answer for QA sites are something that a lot of users want therefore a lot of people are already working with it. But none of them has worked to recommend a question based on user expertise. Such as: Quora, Stack Overflow, Wiki How etc. Our three working areas are followed below.

2.1 Related work of Question Recommendation System

In [1] recommends tags for a new question, given a large corpus of questions with existing tags but it only uses title and body of a question, combined into a single string and a bag of words representation was used to model the text and then used this as the feature. They didn't explore the whole other features to recommend a more accurate question. In another work [2] author discuss the importance of the business intelligence and analytics (BI& A) and the related field of big data analytics. It was basically based on survey data. Industry studies have highlighted this significant development. For example, based on a survey of over 4,000 information technology (IT) professionals from 93 countries and 25 industries, the IBM Tech Trends Report (2011) identified business analytics as one of the four major technology trends in the 2010s. Moreover, [3] has discussed the fact of our information-gathering behavior has always been to find out what other people think. In this case, they worked on a survey data. This survey covers techniques and approaches that promise to directly enable opinion-oriented information-seeking systems. Authors named it "Opinion mining and sentiment analysis".

As we can see in this paper [4] from almost 12 years back. They are discussing personalized service on the Internet, especially in Web-based learning. Generally, most personalized systems consider learner preferences, interests, and browsing behaviors in providing personalized services. Authors study proposes a personalized e-learning system based on Item Response Theory (PEL-IRT) which considers both course material difficulty and learner ability to provide individual learning paths for learners. They worked on the keywords: Distance education, Learning strategies, Intelligent tutoring

systems, Collaborative learning. Here is another work [5] which is based on survey data. This paper discusses the potential difficulties, pitfalls, and problems of the qualitative interview in IS research. Building on Goffmans seminal work on social life, the paper proposes a dramaturgical model as a useful way of conceptualizing the qualitative interview. Based on this model the authors suggest guidelines for the conduct of qualitative interviews.

2.2 Related work of Accepted Answer Prediction System

Predicting accepted answer for online question answering sites is not new in the town. There are works on it. A work to predict the score of an answer is presented in [6]. In this paper, the authors have examined different features about the content and the formatting of the questions. Mainly, the features contain the number of lines, links, tags, and paragraphs with some other factors on the polarity and the subjectivity of the questions. In addition, they have considered the number of answers, favorites, views, comments and time to the accepted answer. Moreover, [7] presents a work to predict the quality characteristics for academic question answering sites using ResearchGate as a case study. In this work, the authors have focused on the peer judgments like votes. They have developed prediction model testing with Naive Bayes, Support Vector Machine (SVM) and multiple regression models.

In [8] a work shows to predict accepted the answer in the massive open online course is done. In this work, they present a machine learning model using historical forum data to predict the accepted answer. They have designed their model considering three types of features. User features - name and details (number of posted question, number of answers given and number of accepted answers) of questioners and responders, Thread features - answers, comments and timing metadata and Content features - number of words of the question and answer, similarity measure (Jaccard similarity and KL Divergence) between question and answers. In [9], a work to find the best answer is described with the help of four categories of features - linguistic, vocabulary, meta and thread. In this work, normalized log-likelihood and Flesch-Kincaid Grade are used. They have used age and rating score as the meta-features. Moreover, Blooma et.al. [10] presents to find the best answer in Community-driven Question Answering (CQA) services. They have used three different categories of features to design their model. Social features - asker/answerer authority, user endorsement, Textual features - answer length, question and answer length ratio, number of unique words, number of nonstop word overlap and number of high-frequency words and Content-appraisal features - accuracy, completeness, presentation, and reasonableness.

In addition, Q. Tian et.al. [11] presents another work to find the best answer in CQA services using the features Answer Context, Question-answer Relationship and Answer Content. In addition, [12] has presented a recent work to find the best answer in CQA. Moreover, [13] has discussed the prediction of closed questions based on the accepted answer findings using reputation.

Tools for data analytics have been in the spotlight since the dawn of data mining era.

2.3 Related work of A Big Data Analytics Tool for Visualization and Knowledge Mining

As a result, there are numerous applications related to knowledge discovery available by this time. As we can see in this paper [14] from almost 20 years back, data mining tools have been a popular research topic from very early ages. As a consequence, now we have many data mining and visualization tools available on different platforms and also for distinctive problem areas. In a recent book [15], authors have discussed the generalized tools built in JAVA and algorithms available in knowledge discovery field. They have covered a generalization of major methods of data mining that produce knowledge representation as output.

Another survey [16] on data-intensive applications and its challenges presented a detailed insight into big data, its analytics, challenges and applications through tools and techniques. They enlisted all the major vendors of big data analytics tools and their products as well. All these applications are basically focused on individual and distinct sectors of problem sets. These problem sets include commercial applications, large cloud-based data storage, biological data analysis and visualization, social media analysis etc. In another work [17], authors proposed a tool for literature survey for scientific researchers. Their tool integrates information from online paper repositories and citation databases to provide visualization of article-level metrics. A similar approach was found in [18], where the authors proposed a web-based tool that discovers meaningful information from resources available online for learning data science, including gigantic open online courses, videos of tutorials and research talks presented at conferences, textbooks, blog posts, and standalone web pages.

2.3 Related work of A Big Data Analytics Tool for Visualization and Knowledge Mining

The education sector has promoted researchers on big data analytics tools. Another literature review paper [19] shows recent activities in the development of tools in the education sector. Apart from these applications, big data analytics have been used in various fields. This paper [20] focuses on impact and tools of big data analytics in the field of health-care. They have addressed challenges related to big data processing in health-care and did an extensive review of big data analysis processes. The authors proposed a big data analytics architecture based on distributed computing to control traffic system. This paper [21] authors proposed a recommender system that reviews the key decisions in evaluating collaborative filtering that basically based on big data. In this paper [22], we can see creators chipped away at the huge information logical theme named "Lessons from applying the orderly writing audit process inside the product building area". Fundamentally this paper mirrors the result of the developing number of observational examinations in programming building is the need to embrace methodical ways to deal with evaluating and collecting research results keeping in mind the end goal to give an adjusted and target synopsis of research prove for a specific point.

Here is another paper [23] discussing the data mining factors. This paper is based on the premises that the purpose of engineering education is to graduate engineers who can design, and that design thinking is complex. This paper [24] is about collective

2.3 Related work of A Big Data Analytics Tool for Visualization and Knowledge Mining

knowledge systems: where the social web meets the semantic web. Which is related to big data analytics and data mining. This paper is based on the authors keynote presentation given at the 5th International semantic web conference, 7 November 2006, which was a call for unifying the social and semantic webs. In addition, Latent Semantic Analysis (LSA) [25] presents another work of recommendation system. Here authors claim that LSA generates latent semantic dimensions that are either interpreted, if the researchers primary interest lies with the understanding of the thematic structure in the textual data, or used for purposes of clustering, categorisation, and predictive modelling, if the interest lies with the conversion of raw text into numerical data, as a precursor to subsequent analysis. In this thesis paper, [26] authors worked on a survey of trust in computer science and the Semantic Web. Trust is an indispensable part in numerous sorts of human collaboration, enabling individuals to act under vulnerability and with the danger of negative outcomes. For instance, trading cash for an administration, offering access to your property, and picking between clashing wellsprings of data all may use some type of trust. In software engineering, trust is a broadly utilized term whose definition contrasts among specialists and application regions. Trust is a fundamental segment of the vision for the Semantic Web, where both new issues and new utilizations of trust are being contemplated. This paper gives a review of existing confide in to explore in software engineering and the Semantic Web. Here is another work [27] of big data. The author says that what can happen on the off chance that we consolidate the best thoughts from the Social Web and Semantic Web? The Social Web is a biological community of cooperation, where esteem is made by the total of numerous individual client commitments. The Semantic Web is an environment of information, where esteem is made by the combination of organized information from numerous sources. What applications can best combine the qualities of these two methodologies, to make another level of significant worth that is both rich with human interest and controlled by all around organized data? This paper proposes a class of uses called aggregate learning frameworks, which open the "aggregate insight" of the Social Web with information portrayal and thinking systems of the Semantic Web. Here is another work [27] found which worked on Latent Semantic Analysis (LSA). Inactive Semantic Analysis (LSA), an individual from a group of methodological methodologies that offers a chance to address this hole by depicting the semantic substance in literary information as an arrangement of vectors, was spearheaded by scientists in brain science, data recovery, and bibliometrics. LSA includes a framework operation called solitary esteem disintegration, an expansion of important part investigation. LSA creates dormant semantic measurements that are either deciphered, if the specialist's essential intrigue lies in the comprehension of the topical structure in the printed information, or utilized for motivations behind grouping, arrangement, and prescient demonstrating, if the intrigue lies with the change of crude content into numerical information, as a forerunner to consequent examination. This paper audits five methodological issues that should be tended to by the analyst who will set out on LSA. Another paper [28] found recently on the recommendation system. Recommender frameworks have created in parallel with the web. They were at first in view of statistic, content-based and communitarian sep-

2.3 Related work of A Big Data Analytics Tool for Visualization and Knowledge Mining

arating. At present, these frameworks are consolidating social data. Later on, they will utilize verifiable, neighborhood and individual data from the Internet of things. This article gives an outline of recommender frameworks and additionally shared separating techniques and calculations; it likewise clarifies their advancement, gives a unique characterization to these frameworks, recognizes territories of future usage and builds up specific regions chose for past, present or future significance.

This recommender system [29] speak to client inclinations to suggest things to buy or inspect. They have turned out to be principal applications in electronic business and data get to, giving proposals that successfully prune extensive data spaces with the goal that clients are coordinated toward those things that best address their issues and inclinations. An assortment of procedures has been proposed for performing suggestion, including content-based, community oriented, learning based and different strategies. To enhance execution, these techniques have at times been consolidated in cross breed recommenders. This paper studies the scene of genuine and conceivable cross breed recommenders, and presents a novel half-breed, EntreeC, a framework that consolidates information based proposal and community oriented separating to suggest eateries. Further, we demonstrate that semantic evaluations acquired from the learning based piece of the framework improve the adequacy of collective separating. In this paper [30] authors claim that their study speculations identified with the utilization of customized content administrations and their impact on client fulfillment. Three noteworthy speculations have been distinguished data over-burden, uses and delights, and client association. The data over-burden hypothesis suggests that client fulfillment increments when the prescribed substance fits client interests (i.e., the proposal exactness increments). The utilizations and delights hypothesis demonstrates that inspirations for data get to influence client fulfillment. The client inclusion hypothesis infers that clients incline toward content prescribed by a procedure in which they have the unequivocal contribution. In this examination, an exploration demonstrate was proposed to coordinate these speculations and two tests were led to look at the hypothetical connections. Our discoveries show that data over-burden and uses and delights are two noteworthy hypotheses for clarifying client fulfillment with customized administrations. Customized administrations can diminish data over-burden and, consequently, increment client fulfillment, however, their belongings might be directed by the inspiration for data get to. This investigation distinguishes speculations identified with the utilization of customized content administrations and their impact on client fulfillment. Three noteworthy speculations have been distinguished data over-burden uses and delights, and client association.

Chapter 3

Question Recommendation System

*Question Answering (QA) service provides facilities to ask questions and get answers from others users in online communities. StackOverflow (StO) is the largest and the most trusted online QA service nowadays. A large number of people in the world have trust in this service. It has also become a great medium of job circulation. Therefor all the users are looking for reputation points all the time to highlight their profile. But unfortunately, there is no personalized recommendation system for them to earn some extra bounty points. Therefore, we have designed a recommender system model (**PRISM**) to recommend questions to users based on their expertize area.*

3.1 Question Recommender for Expert (PRISM)

The system we have built is called Personalized Question Recommender for Expert User. We have utilized the user profile data and Question & answer related metadata to predict whether the user is expert or not. Based on that result we trained a recommender system that can recommend questions to any users with their above-mentioned data.

3.1.1 Feature Selection

StackOverflow (StO) provides the users related data and all the questions and answers with a number of attributes. The attributes contain UpVote, DownVote, Answer Acceptance Percentage etc. It also contains answer related information as count, score, creation date, comment count, comment score etc. From all the information, we have used two categories of features to design our prediction model.

- **User Profile Data** - it contains user reputation, no of badges, views, total bounty. It also contains name, location, age as well as other personal information

3.1 Question Recommender for Expert (PRISM)

that can be useful for training our model. It is basically the data that are related to users and useful for our model training for expert prediction.

- **Question & Answer Related Metadata** - contains question & answer related data that a particular user has answered or asked. Like upvote, downvote, answer acceptance percentage, no of comments, no of posts etc. These are the data that are related to questions & answers that belong to any particular user to train our model. These data have a great impact on our performance of model which we have discussed elaborately in Chapter 6.

3.1.2 Feature Finalization

To design the prediction model, we have tested Random Forest (RF), Logistic Regression (LR), AdaBoost (AdaB), Support Vector Machine (SVM), Decision Tree (DT) & Gaussian Nave Bayes (GNB). On the other hand, Recursive Feature Elimination (RFE) is used to finalize the features from our feature set. After an extensive experimentation, which is presented in Chapter 6, we have selected 20 attributes from the StO provided attributes list to examine our work. Feature finalization is completed with the following steps.

- **User Features Extraction:** Filtered out the user related attribute from the StO provided dataset and added into the feature set to be examined. This is the basic feature selection criteria for our system that we have come up with by research.
- **Thread Features Extraction** - all the important attributes related to the thread according to our research are filtered out and added into the feature set to be tested. Thread features are the ones that are related to the questions & answers but not any textual data. These are really important as they carry a real impact to train our model.
- **Feature Elimination** - in this step, we have used sklearn.feature selection [18] library to select the features according to the Recursive Feature Elimination method. After the elimination, the feature set is updated.

With the finalized feature set we have tested our prediction model. After that the best result along with others have been shown in Chapter 6.

3.1.3 Feature List

After going through all the feature selection procedures we have come up with only 20 features that are really handy for training our model. The list of feature is given below.

With the finalized feature set we have tested our prediction model. After that, the best result along with others have been shown in Chapter 6.

3.1 Question Recommender for Expert (PRISM)

Feature Type	Feature Name
User Profile Data	Accepted Percentage
User Profile Data	Num of Badge
User Profile Data	Views
User Profile Data	Num of Post
User Profile Data	Total Bounty
User Profile Data	Total Score
Question & Answer Related Metadata	Up Votes
Question & Answer Related Metadata	Down Votes
Question & Answer Related Metadata	Num of Comment
Question & Answer Related Metadata	11 Tags

Table 3.1: Feature List of **Question Recommendation System**

3.1.4 Expert User Prediction Model

For expert prediction model for answer recommendation, we have used Decision Tree (DT) algorithm to train our model with 20 features after feature elimination to get the prediction probability of 50 class variable. After training a **scikit-learn** model, it is desirable to have a way to persist the model for future use without having to retrain. We have used **joblib** [20] library of **scikit-learn** to save/export our trained models so that we can use them further to predict another dataset with that model. GraphLab allows us to import & export the recommender model we have generated therefore, we didnt have to use any external library for using our model in any other system.

We have discussed the results details in Chapter 6.

3.1.5 Recommender System

Using the prediction probability result as input for our recommender system we have got the recommended questions for a user. For recommendation, we have used the Graphlab [31] recommender system. In our case we have used **Collaborative Filtering Model** in our recommendation engine.

Lets start by understanding the basics of a collaborative filtering algorithm. The core idea works in 2 steps:

- Find similar items by using a similarity metric
- For a user, recommend the items most similar to the items (s)he already likes.

To give a high-level overview, this is done by making an **item-item matrix** in which we keep a record of the pair of items which were rated together.

In this case, an item is a tag. Once we have the matrix, we use it to determine the best recommendations for a user based on the tag he has the probability of being expert. We

have experimented 3 types of item similarity metrics supported by Graphlab. These are:

- **Jaccard Similarity:**

- Similarity is based on the number of users which have rated item A and B divided by the number of users who have rated either A or B
- It is typically used where we don't have a numeric rating but just a boolean value like a product being bought or an add being clicked

- **Cosine Similarity:**

- Similarity is the cosine of the angle between the 2 vectors of the item vectors of A and B
- Closer the vectors, smaller will be the angle and larger the cosine

- **Pearson Similarity**

- Similarity is the pearson coefficient between the two vectors.

In our case for building recommender model we have used **Cosine Similarity** with the input of our previously got prediction probability result from Decision Tree (DT) classifier of GraphLab library.

3.2 Application Implementation

Personalized Question Recommender for Expert User is a fully featured web application. We have used technologies that are listed below.

Front-end	HTML5, CSS (Bootstrap Framework)
Back-end	Python (Django Framework)
Web Server	Apache 2
Database	MySQL

Table 3.2: System Specification of Our Web Applications

3.2.1 Application System Architecture

We developed the PRISM recommendation system as a three-layer modular architecture, which is depicted in Fig. 3.1, so that we can easily modify and enhance the system performance. This three-layer architecture enables us to separate the recommendation engine layer from the data and user layer. As a result, we can easily modify the recommendation engine layer without modifying the other layer.

3.2 Application Implementation

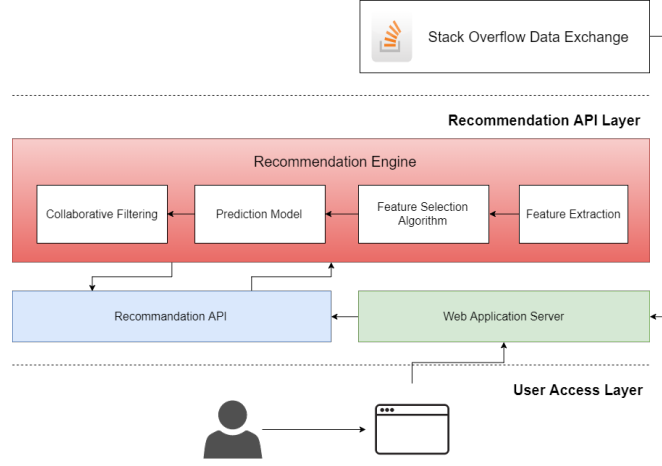


Figure 3.1: Application System Architecture of PRISM -

In the StackOverflow Data Exchange Layer, we utilize the StackOverflow data exchange API to extract the user profile data and question and answer related metadata, such as upvote, downvote, answer acceptance percentage, no of comments, no of posts etc, it is fed to the prediction system engine for generating the suitable questions.

In the Recommendation Engine Layer, when Recommendation API manager calls Recommendation engine, Feature Extraction module is called to extract the necessary feature data from the StackOverflow data exchange. Then, Feature Extraction module fed the extract data to the feature selection algorithm to find out the best possible feature from the data. After getting the selected features, Recommendation Engine employs the pre-trained Prediction Algorithm for identifying the best possible expert area. Finally, the output of the prediction model is used as the input of our recommendation system model to recommend the best possible answer for a particular user. Then the list of questions is returned through the Recommendation API.

In this web app, there are basically three modules involved to recommend the questions. First, feature extraction modules, which extract the related features of the given question from the StackOverflow data exchange system [32]. Secondly, the recommendation engine uses these extracted features and employ the pre-trained model to predict users expert area. And finally, recommend question for that expert area of that user.

3.2.2 Web Application

The user interface of this recommendation system is very easy to use. To access and use this tool users just need to log in with their Stackoverflow account, which is depicted in Fig. 3.2. After successfully sign in with their Stackoverflow account, our system gets the user's some information from the account and to the recommended question, this application sends all the necessary information to our system. After that our system

3.2 Application Implementation

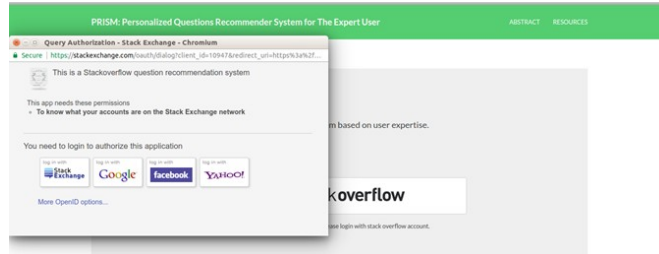


Figure 3.2: Home Page of PRISM -

extract these features and return the questions which have all the information whether the questions should users need to show or not.

Then system sends back all the questions and this application recommended the question and finally show all the questions, which is presented in Fig. 3.3.

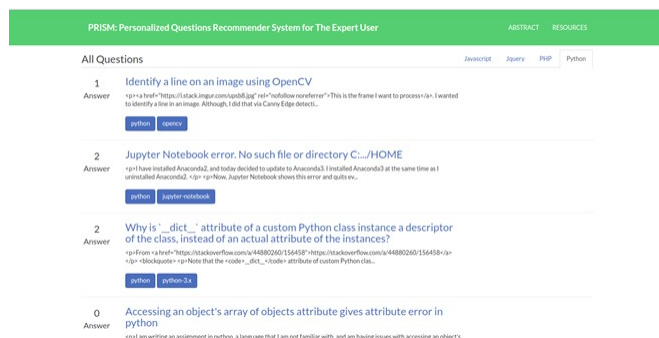


Figure 3.3: Recommended Questions -

Chapter 4

Accepted Answer Prediction System

Question Answering (QA) service provides facilities to ask questions and get answers from others users in online communities. StackOverflow (StO) is the largest and the most trusted online QA service nowadays. A large number of people in the world have trust in this service. As a large community, it has a large number of users as well as questions. Around 62.8% of the questions in StO have marked the accepted answer. As a result, about 37.2% questions are remaining without any accepted answer marking. So, we have designed a prediction model (RAiTA) to predict the accepted answers.

4.1 Recommending Accepted Answer using Textual Metadata (RAiTA)

The system we have built is called Recommending Accepted Answer using Textual Metadata. We have utilized the textual features of the answers comments with the other metadata of the answers.

4.1.1 Feature Selection

StackOverflow (StO) provides the comments on all the questions and answers with a number of attributes. The attributes contain question details with id, score, view count, comment count etc. It also contains answer related information as count, score, creation date, comment count, comment score etc. From all the information, we have used two categories of features to design our prediction model.

- **Thread features** - it contains question title, tags, details, number of responder with response details, question score with favorite count, answer rating, number of commenter with comment score. It also contains view counts as well as timing metadata like question creation time, response time, answer creation time etc.

4.1 Recommending Accepted Answer using Textual Metadata (RAiTA)

Thread features are the ones that are related to the questions & answers but not any textual data. These are really important as they carry a real impact to train our model.

- **Textual Metadata** - this is a finding from the textual data. It represents the sentiment of the responder as well as commenter whether they are positive, negative or neutral with their comments and responses. It only contains the textual data related to the questions & the answers and from these data, we have encountered a new feature which is considered as sentiment and that feature bears our nobility of this stated work.

4.1.2 Feature Finalization

To design the prediction model, we have tested Random Forest (RF), Logistic Regression (LR), AdaBoost (AdaB), Support Vector Machine (SVM), Decision Tree (DT) & Gaussian Nave Bayes (GNB). On the other hand, Recursive Feature Elimination (RFE) is used to finalize the features from our feature set. After an extensive experimentation, which is presented in 6, we have selected 23 attributes from the Stack Overflow (StO) provided attributes list to examine our work. Feature finalization is completed with the following steps.

- **Thread Features Extraction** - all the important attributes related to the thread according to our research are filtered out and added into the feature set to be tested. That was the initial selection criteria of our feature list.
- **Textual Metadata Analysis** - after finalizing the textual features like question body, answer body & comment body, using nltk library [19] for each of these data, we got the sentiment results. The library analyzed the data and returned us the sentiment of the texts as either positive or negative or neutral. Finally, this information is also used as a feature which improves the accuracy of our model while training. We have discussed this elaborately later in Chapter 6.
- **Feature Elimination** - in this step, we have used sklearn feature selection [18] library to select the features according to the Recursive Feature Elimination method. After the elimination, the feature set is updated.

4.1.3 Feature List

After going through all the feature selection procedures we have come up with only 10 features that are really handy for training our model. The list of feature is given below.

With the finalized feature set we have tested our prediction model. After that, the best result along with others has been shown in Chapter 6.

4.2 Application Implementation

Feature Type	Feature Name
Thread Feature	Question Score
Thread Feature	Question Comment Count
Thread Feature	Question Favourite Count
Thread Feature	Question View Count
Thread Feature	Answer Count
Thread Feature	Answer Score
Thread Feature	Answer Comment Count
Thread Feature	Comment Score
Thread Feature	Is Accepted
Textual Metadata	Sentiment

Table 4.1: Feature List of **Accepted Answer Prediction**

4.2 Application Implementation

A web application for the proposed accepted answer recommendation system, RAI_{TA}, is implemented with the technology mentioned below.

Front-end	HTML5, CSS (Bootstrap Framework)
Back-end	Python (Django Framework)
Web Server	Apache 2
Database	MySQL

Table 4.2: System Specification of Our Web Applications

This system is deployed and publicly accessible at <http://210.4.73.237:8888/>

We have also deployed our system as Facebook Messenger Bot Application, which is accessible at <https://www.facebook.com/RAITABOT/>

4.2.1 Application System Architecture

We developed the RAI_{TA} recommendation system as a three-layer modular architecture, which is depicted in Fig. 4.1, so that we can easily modify and enhance the system performance. This three-layer architecture enables us to separate the recommendation engine layer from the data and user layer. As a result, we can easily modify the recommendation engine layer without modifying the other layer.

In the StackOverflow Data Exchange Layer, we utilize the StackOverflow data exchange API to extract the questions and answer related data, such as question ID, list of answers and their comments. After extracting the requested question data, it is fed to the recommendation system engine for generating the suitable answers.

In the Recommendation Engine Layer, when Recommendation API manager calls

4.2 Application Implementation

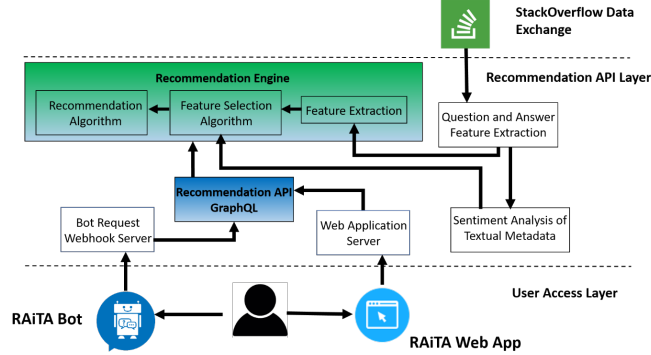


Figure 4.1: Application System Architecture of RAiTA -

Recommendation engine, Feature Extraction module is called to extract the necessary feature data from the StackOverflow data exchange. After extracting the feature data using Question and Answer Feature Extraction module, sentiment Analysis of Textual Metadata is employed to identify the sentiment of the answers comments using NLTK library. Finally, Feature Extraction module fed the extract and processed sentiment data to the feature selection algorithm to find out the best possible feature from the data. After getting the selected features, Recommendation Engine employs the pre-trained Recommendation Algorithm for identifying the best possible answer. Then the list of accepted answer is returned through the GraphQL Recommendation API.

To simplify the user access, we develop two applications, RAiTA Bot, and web application. Both the application access the recommendation engine through their corresponding application server, while the application server only calls the recommendation API module to extract the recommendation engine output. As a result, user access layer is completely separate from the recommendation engine, which enables us to easily modify the recommendation approach without modifying the user application.

In this web app, there are basically two modules involved to predict the accepted answer. First, feature extraction modules, which extract the related features of the given question from the StackOverflow data exchange system [32]. Secondly, the recommendation engine uses these extracted features and employ the pre-trained model to sort answers to that question. Moreover, in each answer, we showed the likelihood percentage of being an accepted answer. RAiTAs full web application is developed using the Python Django framework.

4.2.2 Web Application

The user interface of this recommendation system is very easy to use. Users just need to enter the StackOverflow question link in the search box, which is depicted in Fig. 4.2.

After clicking the predict button our system will sort the answers to this question based on the appropriateness, which is presented in Fig. 4.3.

4.2 Application Implementation

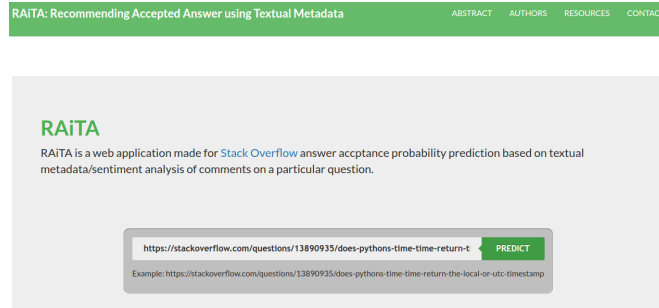


Figure 4.2: Home Page of RAIITA -

In this web app, there are basically two modules involved to predict the accepted answer. First, feature extraction modules, which extract the related features of the given question from the StackOverflow data exchange system [32]. Secondly, the

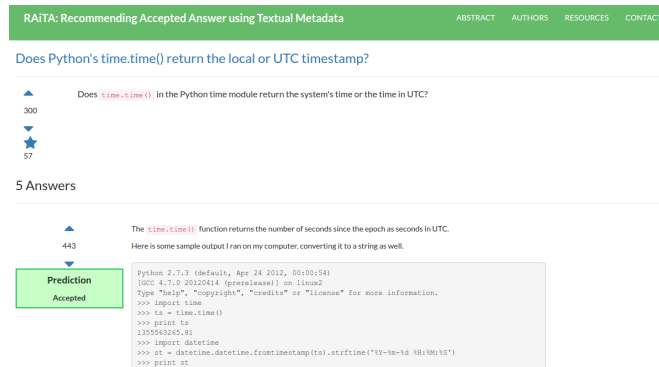


Figure 4.3: Recommended Answers of RAIITA -

recommendation engine uses these extracted features and employ the pre-trained model to sort answers to that question. Moreover, in each answer, we showed the likelihood percentage of being an accepted answer. RAIITA's full web application is developed using the Python Django framework.

Chapter 5

A Big Data Analytics Tool for Visualization and Knowledge Mining

With the increase of using Question Answering (QA) sites a huge amount of data is generated every year which can be useful for both research communities for utilizing this information to extract knowledge and IT industries for developing knowledge-based systems. Visualization of data, pattern mining from datasets and analyzing data drift for the different features are three highly used applications of machine learning and data science fields. A generic web-based tool integrated with such features will provide prodigious support for pre-processing the dataset and thus extracting accurate information. We have built such tool called VIM, which is a web-based comprehensive tool for generic data visualization, data pre-processing and mining suitable knowledge with drift analysis of data.

5.1 Big Data Visualization and Knowledge Mining (VIM)

The system we have built is called A Big Data Analytics Tool for Data Visualization and Knowledge Mining. This system is suitable for any dataset to visualize feature distribution and pattern mining and feature extraction. We have developed our system using Python Django framework and GraphLab library which is deployed to make publicly accessible at <http://210.4.73.237:9999/>.

5.1.1 Key Features

The major contributions of this work are building a generic and scalable data analysis tool which has a very user-friendly UI (User Interface). The three vital features of our proposed system are enlisted below

5.1 Big Data Visualization and Knowledge Mining (VIM)

- **Visualization:** With this feature, users can observe the data distribution of any features of a dataset. When the feature of a dataset is a numeric value, then a line chart will represent the data. On the other hand, in case of nominal data, bar charts are implemented to show the distribution. Therefore, VIM automatically identify which graphical visualization will be appropriate for the data feature.
- **Pattern Mining:** Given a set of selected features from a dataset, users can generate association rules to extract information.
- **Drift Analysis:** From a given dataset users can analyze the drift of data among multiple features. For demonstration purpose, in this work, we only consider the survey data of StackOverflow (<https://www.stackoverflow.com>), however, it can be easily extended to other datasets as well.

5.1.2 Application System Architecture

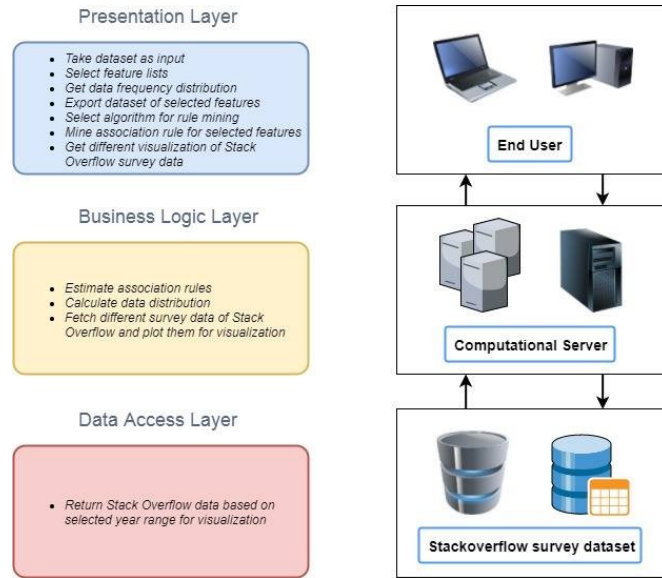


Figure 5.1: Application System Architecture of VIM -

Our proposed data visualization, analysis and knowledge mining tool, VIM, is built upon a three-tier architecture as depicted in Fig. 5.1 Presentation Layer, Business Logic Layer and Data Access Layer.

- **Presentation Layer:** Being the topmost layer of the system, presentation layer works as the interface between the whole system and its users. Users can upload their datasets through this interface and also get their expected outputs here. The functionality of this layer includes uploading datasets, selecting features for drift analysis and association rules, selecting algorithms for rule mining, showing

5.1 Big Data Visualization and Knowledge Mining (VIM)

different representations of data visualization, exporting outputs and few other functions as well.

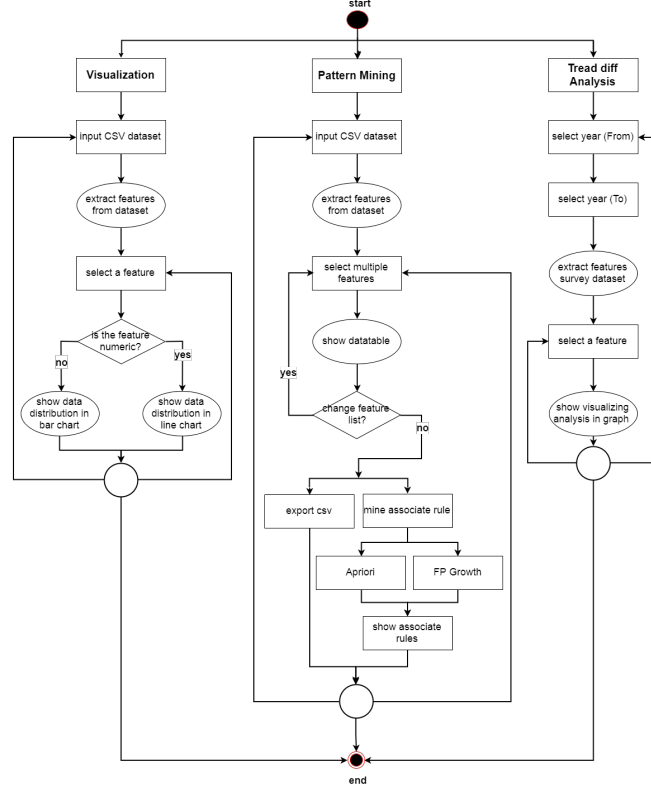


Figure 5.2: Flow Chart of VIM -

- **Business Logic Layer:** In this part of the system, all the calculations and algorithms are run to generate rules, visualization charts and trend lines from a given dataset. For example, this layer employs GraphLab tools to determine the data distribution of features and send back to the presentation layer. Moreover, data preprocessing and extraction are also performed in this portion of the VIM system.
- **Data Access Layer:** Reading and processing of the selected features are done here in this layer. Imported datasets and processed data are temporarily stored here for further analysis.

5.1.3 Web Application

The user interface of VIM is very simple and easy to user. The system specification of the tool is given below.

The key features of VIM are described below.

5.1 Big Data Visualization and Knowledge Mining (VIM)

Table 5.1: System Specification of VIM

Front End	HTML5, CSS39(Bootstrap)
Back End	Python (Django Framework)
Extracting Data	Graph Lab
Distribution	
Web Server	Apache 2
Database Server	MySQL

- **Visualization:** A dataset in CSV format has to be uploaded using the uploading interface of the visualization system, which is depicted in Fig. 5.3. After that, the knowledge mining module, extract the column names as features from the uploaded file. Features will be attached with a radio button, i.e. only one feature distribution can be observed at a time. When a particular feature is selected, our system employs the GraphLab library to generate the data distribution graph automatically for that feature.

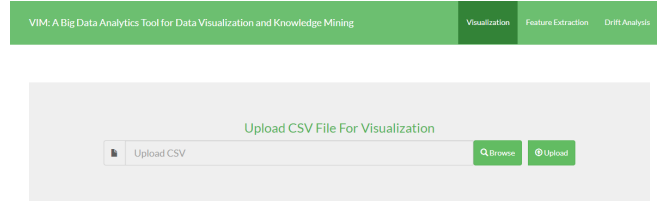


Figure 5.3: Home Page of VIM -

Moreover, our system automatically determines, which type of graphs will be suitable to visualize the data distribution of that selected feature. For example, for numerical feature data our system will choose line graph and the bar graph will be chosen for the categorical data. In Fig. 5.4, the user selects the ProgramHobby feature for data distribution visualization. After selecting the ProgramHobby feature, VIM system automatically determines the type of data, which is a categorical feature, as a result, the bar graph is produced.

- **Feature Extraction for Data Pre-processing:** The same way as in visualization, a CSV file has to be uploaded from which column names will be extracted as features of the uploaded dataset. However, in the feature extraction, features will be attached with checkbox buttons instead of radio buttons, this feature is depicted in Fig. 5.5.

Hence, a user can select multiple features from the feature set and after clicking

5.1 Big Data Visualization and Knowledge Mining (VIM)

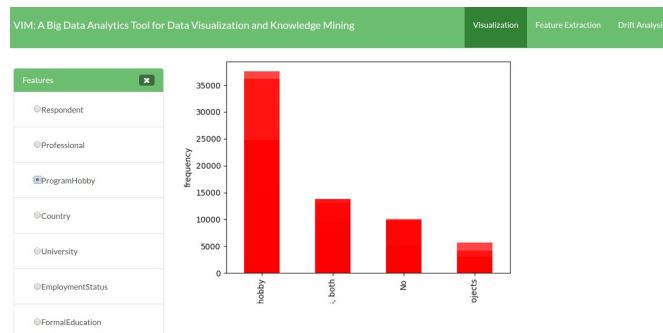


Figure 5.4: Visualization for Data Distribution of (ProgrammingHobby) Feature -

The screenshot shows the VIM tool interface with the 'Feature Extraction' tab selected. On the left, a 'Features' list includes outlook, temperature, humidity, windy, and play. The main area displays a table of feature extraction results. The table has columns for humidity, temperature, windy, outlook, and play. The rows represent individual data points.

humidity	temperature	windy	outlook	play
85	85	False	sunny	no
90	80	True	sunny	no
86	83	False	overcast	yes
96	70	False	rainy	yes
80	68	False	rainy	yes
70	65	True	rainy	no
65	64	True	overcast	yes
95	72	False	sunny	no
70	69	False	sunny	yes
80	75	False	rainy	yes
70	75	True	sunny	yes
90	72	True	overcast	yes
75	81	False	overcast	yes
91	71	True	rainy	no

Figure 5.5: Feature Extraction of Some Selected Features -

5.1 Big Data Visualization and Knowledge Mining (VIM)

the Process Button, which is available in the features table, the user can generate the frequent pattern set. This feature will help the user to find out the importance each of the feature, i.e. using frequent pattern mining we can also extract the features set.

In Fig. 5.6, we can observe that the extracted frequent pattern is listed in the below of the table. Moreover, after finding the pivotal features group set we can export the processed dataset. We just need to select the desired feature group set and click the Export as CSV to export processed dataset in CSV format, which is depicted in Fig. 5.5.



Figure 5.6: Frequently Mined Pattern Rule Set -

- Draft Analysis:** In drift analysis, a user can compare and identify the trend of survey data among different features. For the demonstration purpose, we use only the past six years StackOverflow survey data. A user can select any year from the options list. When the option is selected, it will extract the features from the StackOverflow survey data. The features will be available with radio buttons. The user has to select two features from the selected features list. When the features are selected, it will generate a graph based on the features from that particular years. The x-axis indicates the attributes of the class and the Y-axis indicates the frequency of the attributes. Demonstrations of the drift analysis for a dataset is depicted in fig. 5.7.

5.1 Big Data Visualization and Knowledge Mining (VIM)



Figure 5.7: Drift Analysis of VIM for the Particular Dataset -

Chapter 6

Performance and Evaluation

In our work, we have used different measures to evaluate the performances of the prediction model. Definitions of the measures are given here with respect to Positivity (P) and Negativity (N). Both the characteristics are divided into two parts - False (False Positive, FP and False Negative, FN) and True (True Positive, TP and True Negative, TN).

Confusion Matrix (CM): In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

It is a special kind of contingency table, with two dimensions ("actual" and "predicted"), and identical sets of "classes" in both dimensions (each combination of dimension and class is a variable in the contingency table).

6.1 Performance evaluation of PRISM

Experimenting with a number of the different classifiers, our evaluation shows that we have better results when we use Random Forest algorithm for the expert prediction model. We have evaluated the work using different measures described in earlier in Chapter 3. The datasets and all the results are given below.

6.1.1 Dataset Description

We have collected our datasets from StackExchange Data explorer [32] and then pre-processed those datasets for further usage. Dataset description of accepted answer prediction model is given below.

No of Instances	49992
No of Unique Users	49992
No of Unique Tags	50
Attributes	69

Table 6.1: Dataset Details

6.1.2 Performance Evaluation Criteria

For our answer recommendation system we have adopted different criteria to measure performance which is pretty much familiar to machine learning.

- **Accuracy (Acc):** Accuracy is the closeness of a measured value to a standard value. It is generally measured in percentages. Accuracy can be measured as:

$$Acc = TP + TN / (TP + TN + FP + FN)$$

Accuracy is basically the primary scale of measuring the performance of any model or classifier. Other than that there are few more parameters as well to measure the performance of trained model.

- **Precision (Pre):** Recall is also called positive predictive value which is the fraction of relevant instances among the retrieved instances, while recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. It refers to positive predictive value. It is defined as:

$$Pre = TP / (TP + FP)$$

- **Recall:** Recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. It refers to true positive rate or sensitivity. It is defined as:

$$Recall = TP / (TP + FN)$$

- **F1-Score:** The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall.

6.1.3 Classifier and Feature Selection and Evaluation

In this work, we had a lot of options but only Random Forest (RF), Logistic Regression (LR), AdaBoost (AdB), Support Vector Machine (SVM), Decision Tree (DT) & Gaussian Naive Bayes (GNB) are tested to find the best suited classifier as well as feature set. We have used 10-Fold cross validation on each classifier and 20 feature set to get the best result for our model.

The experimental result for expert prediction model is shown in Table 6.2.

6.2 Performance evaluation of RAITA

Classifier	Acc	Pre	Recall	F1-Score
SVM(rbf)	0.902	0.88	0.90	0.89
SVM(sigmoid)	0.873	0.86	0.88	0.87
Logistic Regression	0.903	0.88	0.90	0.89
Random Forest	0.937	0.91	0.94	0.92
AdaBoost	0.208	0.17	0.20	0.13
Decision Tree	0.963	0.96	0.96	0.96
Gaussian Naive Bayes	0.632	0.77	0.69	0.67

**All the simulation is done under 10-Fold cross validation*

Table 6.2: Results of Different Classifier Analysis for **PRISM**

After analysing Table 6.2, we can see that **Decision Tree classifier, Random Forest & Logistic Regression** achieved good accuracy, Precision, Recall & F1-Score value as 0.963, 0.96, 0.96 and 0.96 & 0.937, 0.91, 0.94 and 0.92 & 0.903, 0.88, 0.9 and 0.89 respectfully. The other classifiers achieve significant accuracy too but since Decision Tree was the best of all therefore, we have considered Decision Tree algorithm to train our model for implementing PRISM application.

6.2 Performance evaluation of RAITA

Experimenting with a number of different classifier, our evaluation shows that we have better results when we use Random Forest algorithm for expert prediction model. We have evaluated the work using different measures described in earlier of Chapter 3. The datasets and all the results are given below.

6.2.1 Dataset Description

We have collected our datasets from StackExchange Dataexplorer [32] and then pre-processed those datasets and merged those according to our interest for further usage. Dataset description for accepted answer prediction model is given below.

No of Instances	49992
No of Unique Questions	20449
No of Unique Answers	23191
No of Accepted Answers	12843
Unique Questions without Accepted Answers	7606
Attributes	23

Table 6.3: Dataset Details

6.2.2 Performance Evaluation Criteria

For our question recommendation system we have adopted different criteria to measure performance which is pretty much familiar to machine learning.

- **Accuracy (Acc):** Accuracy is the closeness of a measured value to a standard value. It is generally measured in percentages. Accuracy can be measured as:

$$Acc = TP + TN / (TP + TN + FP + FN)$$

- **Sensitivity (Sen):** Sensitivity is also called the true positive rate which measures the proportion of positives that are correctly identified as such (e.g. the percentage of sick people who are correctly identified as having the condition). It refers to the true positive rate. It can be calculated as:

$$Sen = TP / (TP + FN)$$

- **Specificity (Spec):** Specificity is also called the true negative rate which measures the proportion of negatives that are correctly identified as such (e.g. the percentage of healthy people who are correctly identified as not having the condition). It refers to the true negative rate. It is defined as:

$$Spec = TN / (TN + FP)$$

- **Matthews Correlation Coefficient (MCC):** The Matthews correlation coefficient is used in machine learning as a measure of the quality of binary (two class) classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient between the observed and predicted binary classifications. It is important to measure the quality of our binary classifier. It can be calculated directly using positive and negative values of the model. It is defined as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Figure 6.1: MCC Equation -

- **Area Under the Curve (AUC):** Receiver Operating Characteristics also known as ROC Curves are important to visualize the accuracy of a model. This is a FP-TP curve. The area under the curves defined as AUC value of a model. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection in machine learning. The false-positive rate is also known as the fall-out or probability of false alarm and

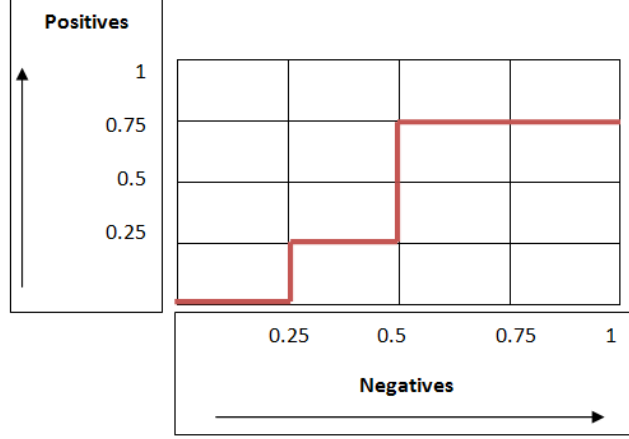


Figure 6.2: ROC Curve -

can be calculated as $(1 - \text{specificity})$. We can compute the **area under curve (AUC)** from ROC plots. In our Fig. 6.2, that is essentially the area below the red line. A perfect classifier will have an AUC of 1. AUC is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming positive ranks higher than negative).

6.2.3 Effectiveness of Sentiments in Accepted Answer

The textual metadata as sentiment is very effective in this work. We have tested the effect of sentiments a number of times in our prediction model. The summary of the experiment without feature optimization & 10 Fold cross validation is shown in 6.4. Analysing the results in Table 6.4, we see that the model works better with 89.7% accuracy, AUC and MCC values 0.886 and 0.774 respectfully when we use both features & sentiments.

Feature Set	Acc	Sen	Spec	AUC	MCC
Features & Sentiment	0.897	0.848	0.923	0.886	0.774
Only Features	0.89	0.836	0.919	0.878	0.758
Only Sentiment	0.651	0.024	0.995	0.509	0.08

**All the simulation is done under 10-Fold cross validation*

Table 6.4: Results of the Sentimental Analysis

6.2.4 Classifier and Feature Selection and Evaluation

In this work, Random Forest (RF), Logistic Regression (LR), AdaBoost (AdB), Support Vector Machine (SVM), Decision Tree (DT) & Gaussian Naive Bayes (GNB) are tested

6.2 Performance evaluation of RAiT_A

to find the best suited classifier as well as feature set. We have used 10 fold cross validation for our experiments.

The experimental result for accepted answer prediction model is shown in Table 6.5.

Classifier	Acc	Sen	Spec	AUC	MCC
SVM(rbf)	0.648	0.005	1	0.503	0.057
SVM(sigmoid)	0.651	0.048	0.981	0.514	0.03
Logistic Regression	0.652	0.051	0.981	0.516	0.09
Random Forest	0.897	0.848	0.923	0.886	0.774
AdaBoost	0.692	0.436	0.832	0.634	0.291
Decision Tree	0.889	0.849	0.911	0.88	0.758
Gaussian Naive Bayes	0.408	0.949	0.111	0.53	0.1

**All the simulation is done under 10-Fold cross validation*

Table 6.5: Results of Different Classifier Analysis for RAiT_A

After analysing the Table 6.5, we see that both **Random Forest classifier** & **Decision Tree** achieves the best accuracy, AUC and MCC value as 0.897, 0.886 and 0.774 & 0.889, 0.88 and 0.758 respectfully. The other classifiers achieve significant accuracy too. But since Random Forest is slightly better than Decision Tree we have considered Random Forest algorithm to finally train our model for further application. The achievement in AUC value is visually represented in Fig. 6.3.

Classifier of Acc(%)	RF	LR	AdB	SVM
No of Features 11	88.17	66.03	69.69	83.77
No of Features 10	88.45	66.12	69.68	82.50
No of Features 9	88.09	66.11	69.60	81.31

**All the simulation is done under 10-Fold cross validation*

Table 6.6: Results of Different No. of Features for RAiT_A

From Table 6.6, we see that Random Forest with 10 features achieves the best accuracy of 88.45%. The other classifiers achieve significant accuracy too.

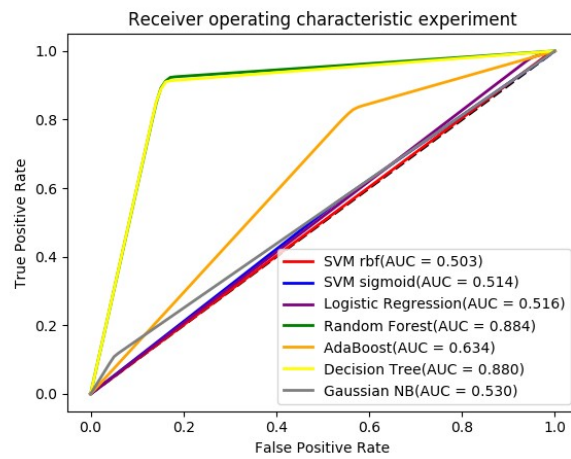


Figure 6.3: ROC curves for different classifier -

Chapter 7

Conclusion

In this chapter, we summarize the results presented in this thesis and state few directions for future works.

7.1 Summary of Research

We developed three different systems based one single terminology and that is "Big data analysis". Big Data brings new opportunities to modern society and challenges to data scientists. On the one hand, Big Data holds great promises for discovering subtle population patterns and heterogeneities that are not possible with small-scale data. On the other hand, the massive sample size and high dimensionality of Big Data introduce unique computational and statistical challenges, including scalability and storage bottleneck, noise accumulation, spurious correlation, incidental endogeneity and measurement errors. These challenges are distinguished and require new computational and statistical paradigm. We implemented different kinds of algorithms such as Ada Boost, K-Nearest Neighbour, Association Rule Mining etc. Finally, this paper concludes the following:

7.1.1 Question Recommendation System

To design this system, we have to go through users profile, question & answer related metadata to predict whether the user is expert or not. To predict, we have tested Random Forest, Logistic Regression, Decision Tree etc. Then we have trained our recommender system based on the predicted results. Finally, to recommend questions, recommender system gives back the result in where users are expert. For recommendation we have used **Collaborative Filtering Model** in our recommendation engine.

7.1.2 Recommending Accepted Answer using Textual Metadata

For this system, we have to go through every questions, tags, user's response details, question scores, favorite count, answer rating etc to predict whether the answer will be

accepted or not. We also have to go through every question answers for understanding the sentiment of their response. For understanding the sentiment, we have used Natural Language Processing tools. To predict, we have tested Random Forest, Logistic Regression, Decision Tree etc. Then we have trained our recommender system based on the predicted results.

7.1.3 Recommending Accepted Answer using Textual Metadata

To suitable for any dataset to visualize feature distribution, pattern mining and feature extraction, we have to come up a strategy. For visualization, we have to find out whether the data value is numeric or nominal. For numeric we have used the line chart to visualize data and for nominal we have used the bar chart. For pattern mining, we generate association rule mining to extract information. And last, for drift analysis, we only consider the survey data of StackOverflow data. However, it can be easily extended to other datasets as well.

7.2 Discussion

As the popularity of Stackoverflow is increasing day by day, more people will rely on it to give and to get their answers. So, it is the time to implement personalized recommender system for questions/posts to users and confirm the acceptance of the answers for each and every question using an automated system. Using our systems everyone will be beneficial in every aspect. It uses users & answers metadata information with sentiment analysis to predict.

Visualization of Big Data gives an intuitive perception of any system. In addition to that, pattern mining helps to identify the important features. Moreover, Drift analysis paves the way to analyze the changes in the features data distribution. Therefore, building a tool that serves these purposes in a comprehensive way will surely come handy for building knowledge-based system and the research purpose.

However, our fully functional web applications **PRISM**, **RAiTA** & **VIM** has some scope to enrich with more fancy features and to improve for the faster response which we will be focusing in our upcoming updates.

7.3 Future Works

- We will try to incorporate the textual metadata of questions for recommendations.
- For the new users, we will intend to make a system where users can find their desire field to work with.
- In addition, the success of finding the accepted answer in StackOverflow has interested us to extend the system to design an adaptive recommender system for other QA services like Quora, Yahoo, ResearchGate etc.

- We will focus on building a system where user's answer will be accepted or not based on their answer before answering any question. It'll be two field system. In the first field, users will submit the question and in other's one, they will submit their answer. It'll then predict whether the answer to that question will be acceptable or not.
- For new users, we intend to build a system, where users can find the answer type or the probable answer to the questions. And then users can write down their own answers so that the probability of accepting their answer can increase.
- We intend to extend VIM so that our system can help users to automatically process data by eliminating noisy features.

Bibliography

- [1] S. D. Manindra Kumar Moharana, University of California, pid: A53068263 from <http://cse.ucsd.edu>. 6
- [2] H. Alharthi, D. Outioua, and O. Baysal, “Predicting questions’ scores on stack overflow,” in *CrowdSourcing in Software Engineering (CSI-SE), 2016 IEEE/ACM 3rd International Workshop on*. IEEE, 2016, pp. 1–7. 6
- [3] L. Li, D. He, W. Jeng, S. Goodwin, and C. Zhang, “Answer quality characteristics and prediction on an academic q&a site: A case study on researchgate,” in *Proceedings of the 24th international conference on world wide web*. ACM, 2015, pp. 1453–1458. 6
- [4] M. Jenders, R. Krestel, and F. Naumann, “Which answer is best?: Predicting accepted answers in mooc forums,” in *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 679–684. 6
- [5] F. Calefato, F. Lanubile, and N. Novielli, “Moving to stack overflow: Best-answer prediction in legacy developer forums,” in *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, 2016, p. 13. 7
- [6] M. J. Blooma, A. Y.-K. Chua, and D. H.-L. Goh, “Selection of the best answer in cqa services,” in *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*. IEEE, 2010, pp. 534–539. 7
- [7] Q. Tian, P. Zhang, and B. Li, “Towards predicting the best answers in community-based question-answering services.” in *ICWSM*, 2013. 7
- [8] D. Elalfy, W. Gad, and R. Ismail, “A hybrid model to predict best answers in question answering communities,” *Egyptian Informatics Journal*, 2017. 7
- [9] C. G. E. Lezina and A. M. Kuznetsov, “Predict closed questions on stackoverflow,” 2013. 7
- [10] M. Goebel and L. Gruenwald, “A survey of data mining and knowledge discovery software tools,” *ACM SIGKDD explorations newsletter*, vol. 1, no. 1, pp. 20–33, 1999. 7

- [11] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016. 7
- [12] C. P. Chen and C.-Y. Zhang, “Data-intensive applications, challenges, techniques and technologies: A survey on big data,” *Information Sciences*, vol. 275, pp. 314–347, 2014. 7
- [13] Y. Liu, Z. Huang, Y. Yan, and Y. Chen, “Science navigation map: An interactive data mining tool for literature analysis,” in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 591–596. 7
- [14] J. L. Ambite, L. Fierro, F. Geigl, J. Gordon, G. A. Burns, K. Lerman, and J. D. Van Horn, “Bd2k erudite: the educational resource discovery index for data science,” in *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 1203–1211. 8
- [15] K. Sin and L. Muthu, “Application of big data in education data mining and learning analytics—a literature review.” *ICTACT journal on soft computing*, vol. 5, no. 4, 2015. 8
- [16] Y. Wang, L. Kung, W. Y. C. Wang, and C. G. Cegielski, “An integrated big data analytics-enabled transformation model: Application to health care,” *Information & Management*, 2017. 8
- [17] S. Amini, I. Gerostathopoulos, and C. Prehofer, “Big data analytics architecture for real-time traffic control,” in *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on*. IEEE, 2017, pp. 710–715. 8
- [18] “Recursive feature elimination,” http://scikit-learn.org/stable/auto_examples/feature_selection/plot_rfe_digits.html. 8, 12, 18
- [19] “Nltk is a leading platform for building python programs to work with human language data,” <http://www.nltk.org/>. 8, 18
- [20] “Model persistence,” http://scikit-learn.org/stable/modules/model_persistence.html. 8, 13
- [21] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004. 8
- [22] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, “Lessons from applying the systematic literature review process within the software engineering domain,” *Journal of systems and software*, vol. 80, no. 4, pp. 571–583, 2007. 8

- [23] S. Tufféry, *Data mining and statistics for decision making*. Wiley Chichester, 2011, vol. 2. 8
- [24] T. Gruber, “Collective knowledge systems: Where the social web meets the semantic web,” *Web semantics: science, services and agents on the World Wide Web*, vol. 6, no. 1, pp. 4–13, 2008. 8
- [25] T. K. Landauer, P. W. Foltz, and D. Laham, “An introduction to latent semantic analysis,” *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998. 9
- [26] D. Artz and Y. Gil, “A survey of trust in computer science and the semantic web,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 58–71, 2007. 9
- [27] N. Evangelopoulos, X. Zhang, and V. R. Prybutok, “Latent semantic analysis: five methodological recommendations,” *European Journal of Information Systems*, vol. 21, no. 1, pp. 70–86, 2012. 9
- [28] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowledge-based systems*, vol. 46, pp. 109–132, 2013. 9
- [29] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002. 10
- [30] T.-P. Liang, H.-J. Lai, and Y.-C. Ku, “Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings,” *Journal of Management Information Systems*, vol. 23, no. 3, pp. 45–70, 2006. 10
- [31] “Graphlab, a machine learning modeling tool for developers and data scientists,” 2019, [Accessed 2019-09-28]. 13
- [32] “Stackexchange data explorer for stackoverflow,” [<https://data.stackexchange.com/stackoverflow/> Accessed 2019-09-28]. 15, 20, 21, 29, 31