KweriME: A Q&A based model which predicts the accepted answers of questions in CQA sites

Chapter 1

Introduction

In today's age everyday humans are encountering problems for which they need a best and accurate answer. This is where a Question and Answer based site such as Quora, Yahoo!Answers, Stack Overflow etc comes into picture. They are important sources of content creation providing an entire knowledge base for general questions or domain specific questions.

Such growing contents are posing several challenges which open up new opportunities for researchers to analyze their field of interest. Mostly, there are two types of users involved in these sites one being the asker, who asks or posts questions on these sites and waits for the question to be answered and second being the answerer who answers these questions which is posted by the asker. Usually multiple answerers could provide the answer, thus it becomes difficult to select the best answer.

Since multiple answers are available to the asker on these sites, it is the need of the hour that a best answer must be selected amongst them to provide ease to the asker thus, ensuring a proper and optimized working environment for both the asker as well as the answerer.

1.1 Purpose

- KweriME is a Q&A based model which predicts accepted answers of various questions in a CQA sites. Seeking information on the internet has become a daily part of our lives. This inspired us to build an efficient system which reduces delay as well as gives us additional information on the content.
- Most of the Q&A systems experience a delay which we try to solve by forwarding the questions to experts thus, ensuring improved answer quality and also incorporate an anti-spammer control to filter out irrelevant and advertising verbiage to optimize user interaction .
- Instead of going through multiple questions, the asker will be satisfied with the best and optimal answer, thus, saving time and energy of the asker.

1.2 Scope

In our day to day life we learn new things which helps to know other things better.KweriME is a community based Q&A model where best answer is predicted from set of answers and also provides wide variety of information. We formulate our work by using four research questions which are as follows:

- Who provides answer to the question on the stack overflow?
- How much expertise does the answerer has in:(i)Same domain (ii) Different domain?
- To what extent the answer is relevant to the question?
- How the best answer is selected amongst the set of answers?

1.3 Literature Survey

Topical interest and Recommending the best answer are the talk of the town which has attracted many researchers interest.

- In [1], the activeness of users has been explored in CQA.They have shown how badges and reputation scores are related to the activeness in different forums based on statistical analysis. Yuhua Lin et al discussed on clustering the users of Stack Overflow into four clusters namely naive, surpassing, experts and out shiners based on characteristics accounting various metrics by using machine learning algorithm in order to predict the users' activities.
- In [2], Tirath Prasad Sahu et al. has worked on the goal of uncovering topic interest, main discussion topics and technology trends over time with the help of statistical topic modelling and also they have worked on the goal of uncovering topic interest, main discussion topics and technology trends over time with the help of statistical topic modelling. In [2], the Questions posted on Stack Overflow has been analyzed both quantitatively and qualitatively in order to improve the success of CQA.

1.4 Existing Systems

To the best of our insight, the choice of a best response for the question asked on the CQA site is done manually, which is conventional and dreary.

Previous research in Community Question Answer platforms, used features such as number of upvotes to predict the quality of answers. Few researchers conducted a survey and learned that there was a correlation between posting a high quality question and getting a high quality answer. They found that question related features such as tags, length of the question, presence of examples enhanced user understanding of the question.

While the work has provided useful insight into quality - a measure of CQA content these experimentations use only social features such as user rating and authority but ignored the textual features or content-appraisal features in their study.

Also, in few previous works, they extracted features which contain information from the questions, the answers, and the users. But there is no consideration on the relationship between the answers and the questions.

1.5 Proposed System

Motivated from the work presented by aforementioned authors, this study addresses the answer to unanswered research question - how are the best answers selected?.

- First, we use activity signatures, domain knowledge and topical similarity of the user to identify active answerers in the domain of the questions asked.
- Second, we use topical interest, topical expertise for expertise computation using topic modeling and voting scores.
- Third, we find topic relevancy to find the relationship between Q&A pairs.
- Finally, we focus on features involved in posts for predicting whether the answer to the question will be accepted or not based on different classifiers.

The proposed approach examines StackOverflow Q&A posts with no less than five responses to extract features for pattern identification using which the best answer is selected for the asked questions based on topic modeling and classifier.

To assess rightness of the proposed model, a set of parameters are utilized, for example, Receiver Operating Characteristics Area Under Curve, Precision Recall Area Under Curve, F-score, and Accuracy.

In our approach, a model is developed for choosing best answer for the question asked on the CQA site. Rather than taking data associated with question and answer alone into account as done in the manual process, this model takes both question-answer and answerers' information into account, which gives an understanding perspective into the appropriate responses given by the experts that is more liable to be chosen as the best answer.

Results demonstrate that the proposed model is compelling in anticipating the best answer.

1.6 Statement of the problem

To address the classic issues that dwell in the community based Q&A websites, we make use of a reputation based Q&A system which employs a category and theme based reputation management system to evaluate users willingness and capability to answer various kinds of questions. With the ample amount of data at hand, we are trying to leverage machine learning to detect patterns as well as to create a healthy environment that helps in enhancing the response latency and answer quality.

1.7 Summary

This chapter gives a brief introduction on what exactly the proposed system is. It also covered the future scope and the demand of this product in the market. Also how this product can help users on community Q&A sites to find the best possible answer for a particular question and save lots of time and energy. It even marked the main features of the device which helps in overcoming the drawbacks of the existing system.

Chapter 2

System Requirements Specifications

2.1 Software Requirements Specifications

Requirement specification is the movement of interpreting the data assembled amid investigation into prerequisite report.

Software requirements specifications are the detailed enlisting of all necessary requirements that arise in the project. The aim of having these requirements is to gain an idea of how the project is to be implemented and what is to be expected as a result of the project. The sections in this chapter deal with the various kinds of software, hardware and other functional and non functional requirements of the project. A brief description of the various users of the system is also mentioned.

2.1.1 Operating Environment

This section gives a brief about the hardware and software prerequisites for the project.

Hardware Requirements

- **Processor**: 1.6GHz or faster processor
- **RAM**: 1 GB(32 bit) or 2GB(64 bit)
- **Storage**: 250GB of available hard disk space
- Other general hardware/peripheral devices such as a mouse and keyboard for inputs and a monitor for display.

Software Requirements

- **Operating system**: Ubuntu 14.04 and above
- **Programming languages**: Python

- **Documentation**: Overleaf
- **Tools**: Spyder

2.1.2 Functional Requirements

Functional requirements are a formal way of expressing the expected services of a project. We have identified the functional requirements for our project as follows:

- The system should be able to gather data from user.
- The system should be able to check for correctness of data entered by user.
- The system should be able to predict correctly if a particular answer to a question is the accepted answer or not.
- The system should be able to have the capacity to decide the contribution of each attribute towards the decision made by the predictor.

2.1.3 Non-Functional Requirements

Non functional requirements are the various capabilities offered by the system. These have nothing to do with the expected results, but focus on how well the results are achieved.

- **Usability**: The user must be familiar with the user interfaces and should not have problems in migrating to a new system with a new environment.
- **Reliability**: The changes made by the Programmer should be visible both to the Project leader as well as the Test engineer.
- **Security** : Including bug tracking the system must provide necessary security and must secure the whole process from crashing.
- **Performance** : The system will be hosted on a single web server with a single application server in the background, hence performance becomes a major concern.
- **Portability** : This is required when the web server, which is hosting the system gets stuck due to some problems, which requires their system to be taken to another system.

- **Re-usability** : The system should be divided into such modules that it could be used as a part of another system without requiring much of work.

2.1.4 User Characteristics

There are two kinds of users involved on StackOverflow

- **Asker** – a user who posts question on a wide exhibit of topics, also hold up an answer from different users.
- **Answerer** – a user who presents reply on the inquiry posted by the asker.

Stack-Overflow enables askers to pose their queries as questions and gets numerous answers from their fellow users.

2.1.5 Applications

The model could be trained on data for different domains and platforms owing to its generic nature in dealing with identifying influential features measuring association between question and answer. This tool can be used to plan accepted answer finding strategies in question-answering platforms and save the users lots of time, energy and money and in turn lead to better productivity.

2.2 Summary

This chapter talks about the Software Requirement Specification that introduces the hardware and software requirements and the operating environment needed to run the system. It also describes the different users that interact with the application, along with its application in particular fields. It gives an outline that is needed to design, develop and test the application.

Chapter 3

High Level Design

Design is concerned with identifying software components specifying relationships components, specifying software structure and providing blue print for the document phase. A software design is description of the structure of the software to be implemented, the data which is part of the framework, the interfaces between the framework and sometimes the algorithms used. Designers do not arrive at a finished design immediately but develop the design iteratively through a number of different versions. The design process involves adding formality and detail as the design is developed with constant backtracking to revise prior designs.

A high-level design document or HLD adds the necessary details to the current project description to represent a suitable model for coding. This document includes a high level architecture diagram depicting the structure of the system. High level Design gives the overall System Design. This is very useful for the developers to understand the flow of the system.

- **Preliminary design:** In the preliminary stages of a software development, the need is to size the project and to identify those parts of the project that might be risky or time consuming.
- **Design overview:** As the project proceeds, the need is to provide an overview of how the various sub-systems and components of the system fit together. In both cases the high-level design should be a complete view of the entire system, breaking it down into smaller parts that are more easily understood.

3.1 Design Considerations

There are numerous aspects to consider in the design of a piece of software. The significance of each should reflect the goals the software is trying to accomplish. Some of the aspects like compatibility, reliability, maintainability, usability, security etc. ought to be fulfilled by the design.

3.1.1 Assumptions and Dependencies

There are several Assumption and Dependencies which form the basis for some of the decisions taken. These are the following:

Assumptions:

- System can be able to handle for large set of inputs.
- Assuming that the algorithm is selecting the best features in order to learn the correlations amongst the data.
- The time taken by the machine depends on the input size.
- Values of pre-computed features used for training is right.
- Features taken best suits our problem.

Dependencies:

- Operating System: This System is designed to run on any operating system with NLTK, and Python compatibility support.
- Working of Software model also depends on the various factors like RAM Capacity, processor speed, and server response speed and amount of data.
- Following are the required libraries for designing and analyzing in the experiment provided by NLTK, Pip and Python : Pandas, Textblob, re, Scikit-learn, Numpy, Beautiful Soup, Textstat, Matplotlib, Seaborn.

3.1.2 Goals and Constraints

Goals:

- The Python scripts are responsible for execution of the algorithm and visualizing the data and the results.
- All the features are used and machine learning is based on the mathematical algorithms used.
- Once the scripts are executed the results are generated in form of scores.
- The various calculations that are carried out are displayed.
- The features include the similarity check based on corpus based algorithm (Term frequency-Inverse document frequency) in NLP,Readability, Answer and Answerer score, Time Difference and Polarity.

Constraints:

- The project's focus is on StackOverflow Community Question-Answering platform.
- Input data is limited to 6 features.
- Response speed will be decreased with respect to the size of input and number of features.
- The data is restricted to include those questions with atleast one answer and ones which have a marked accepted answer by the asker of the corresponding question.
- The data is also constrained in a way to include only python related questions.

3.2 System Architecture

The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces.

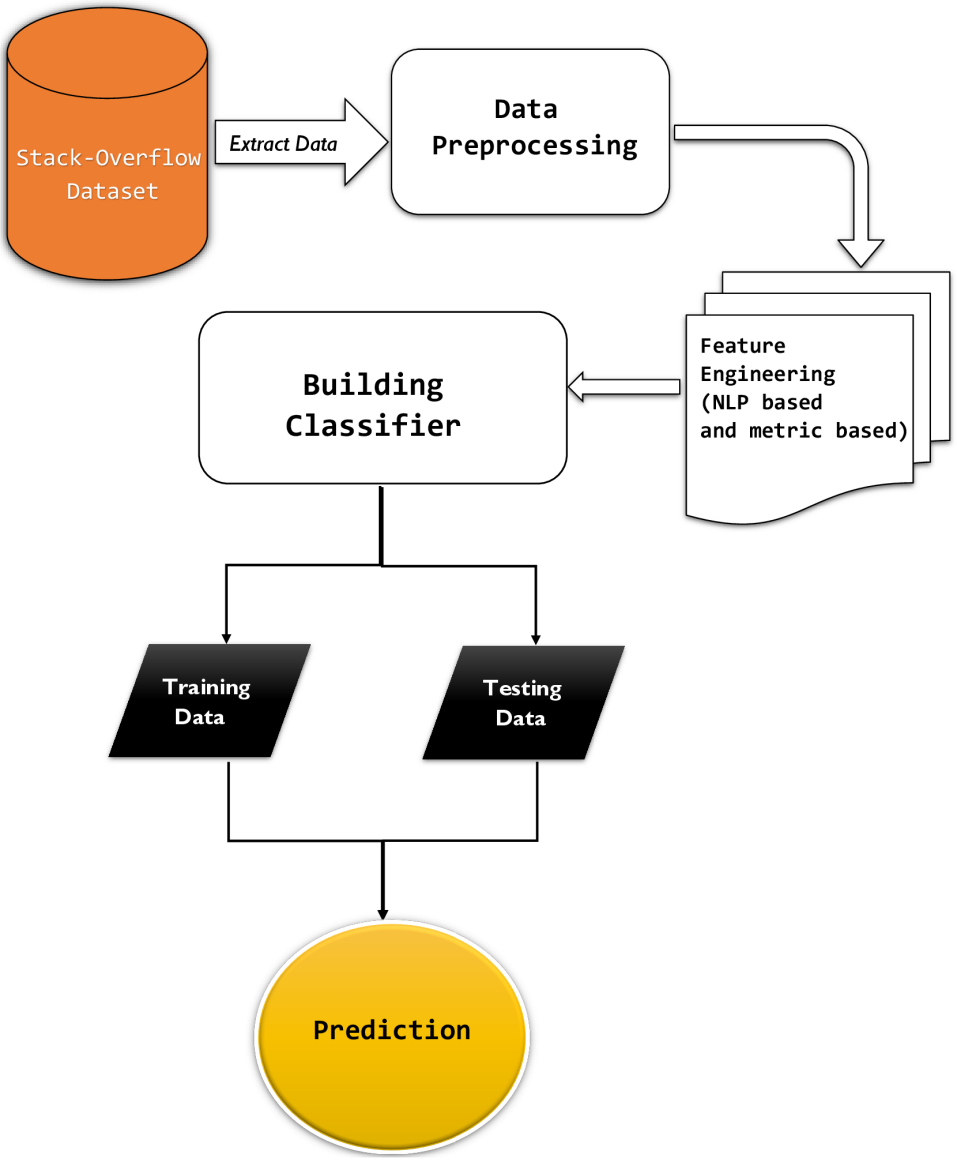

Figure 3.1: System Architecture

3.3 Data Flow Diagram

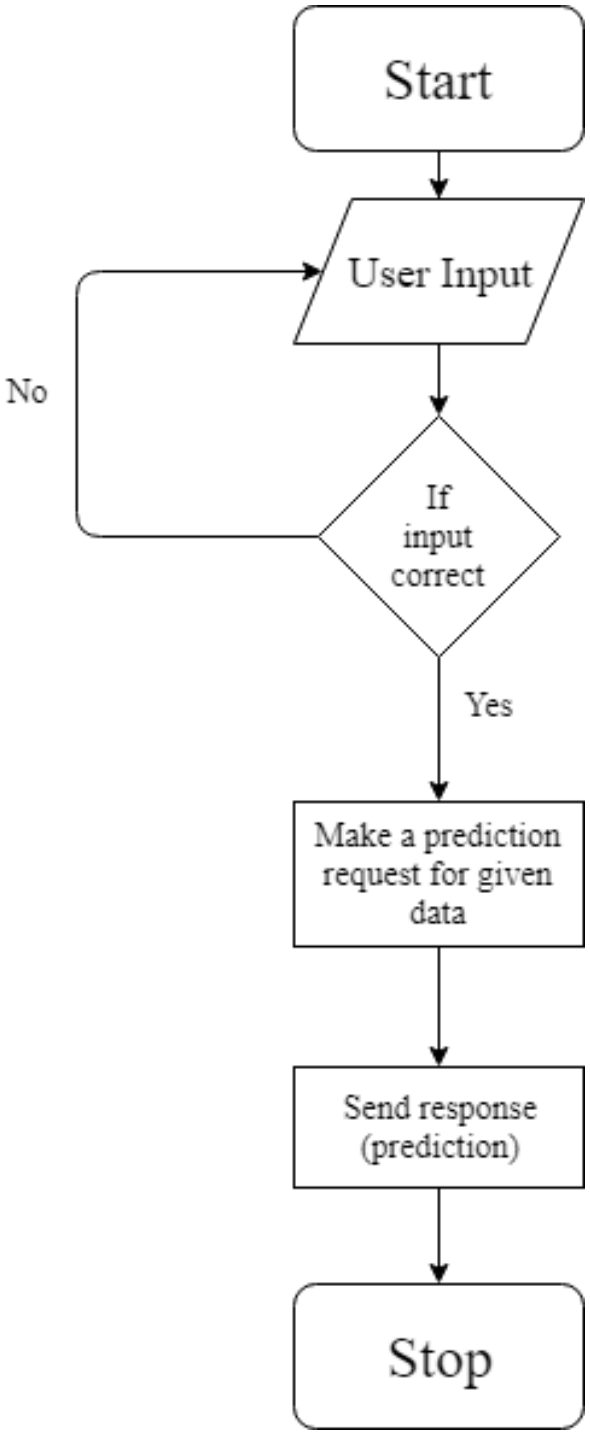

Figure 3.2: Data Flow Diagram

3.4 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

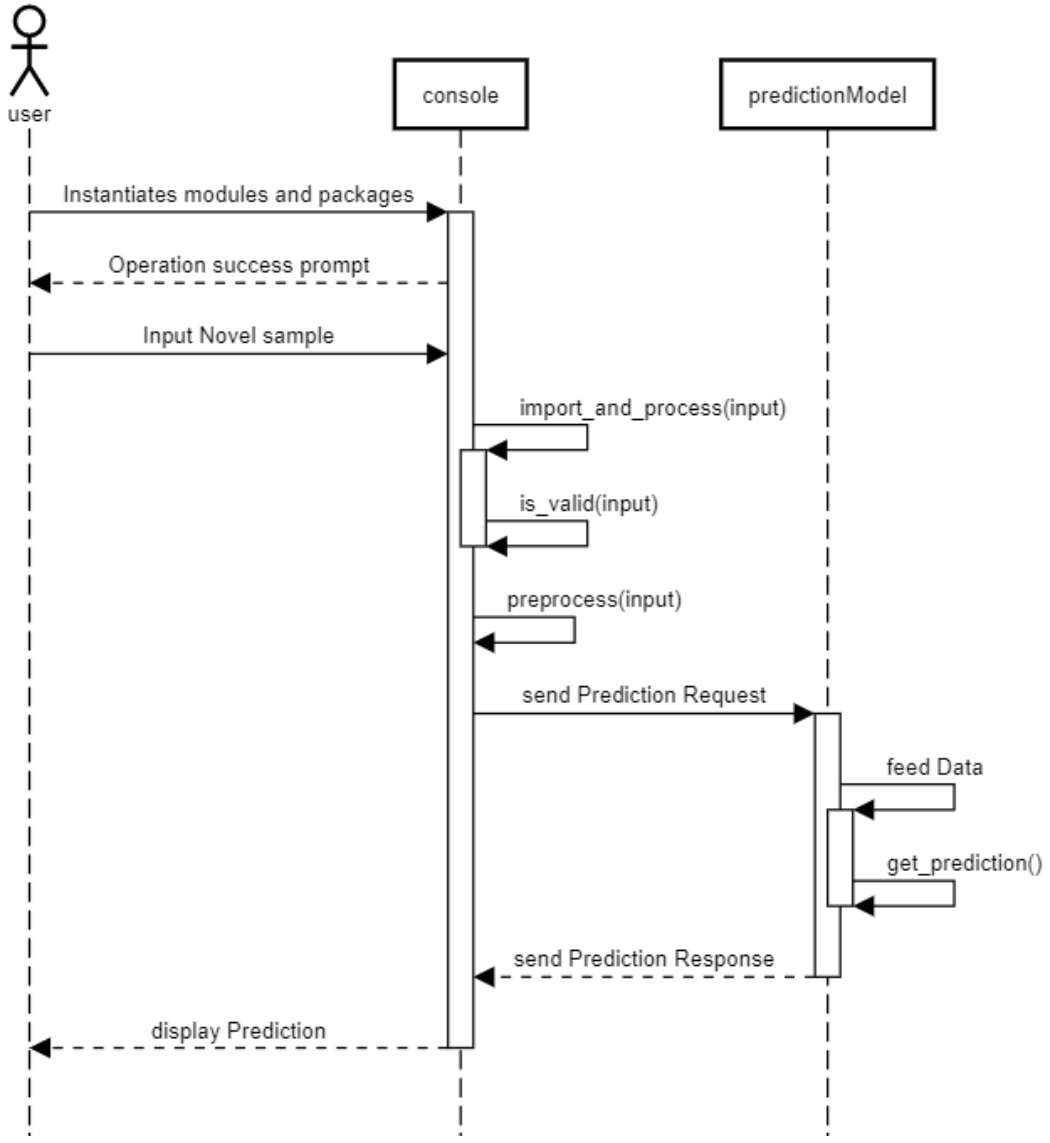

Figure 3.3: Sequence Diagram

3.5 Summary

High Level Design provides an overview of an entire system, identifying all its elements at some level. This contrasts with Low-level Design which exposes the detailed design of these elements. In this chapter the goals and constraints, the assumptions and dependencies of the project are discussed. The system architecture and the steps for processing are explored. The data flow in the KWERIME is explained and the communication between various entities in KWERIME is understood.

Chapter 4

Detailed Design

4.1 Purpose

The purpose of the design is to plan our system to meet the requirements specified at the start. In the detailed design we see what is the input data for each model, how the model implementation is carried out and how the output is interpreted.

4.1.1 Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put the collected data in to a usable form.

The design of input focuses on controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple.

The KAGGLE dataset and SEDE(Stack Exchange Data Explorer) data are taken as input for training data. Using those datasets the machine is trained and using machine learning algorithms the model classifies the answers as accepted answer or not based on various contextual and StackOverflow metric features.

The parameters that are taken as input from the datasets should consider following things:

- Which fields to be selected to get the required results?
- Methods to compute the selected parameters.

4.1.2 Output Design

A quality output is one, which meets the requirements as per SRS and presents the information clearly.
The output form of an information system should accomplish one or more of the following objectives.

- Future Scope

- It should show the predicted score

The final outcome is classified as the predicted score.

4.2 Modules

The project is categorized into 5 modules and they are as follows:

4.2.1 Feature Extraction

It starts from an initial set of measured data and builds derived values (features) intended to be informative and non- redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction. Features that are being used in the project are Readability, Answer and Answerer score, Topical Similarity(NLP), Time Difference and Polarity.

4.2.2 Parameter Computation

This mainly involved removing placeholders for proper nouns, preparing the text, tokenizing and stripping essays of punctuation. Intrinsic variables in an essay like the style and fluency cannot be directly measured-they have to be approximated with measurable quantities like the sentence and word length, length of essay etc

4.2.3 Designing Learning Model

We model a prediction system as a binary classifier, which classifies an answer as an accepted answer or not an accepted answer. We next use classification algorithms based on Bayes rule such as Naive Bayes and Decision tree to predict acceptability of the answer by the asker or community. We also implement a dummy classifier(Baseline) to provide a basis for comparison with other classifiers.Based on the features that were extracted, we classify the answers as best or not.This is suggested by figure 4.1 which indicates type 1 as being accepted as the best answer and type 2 as being rejected.

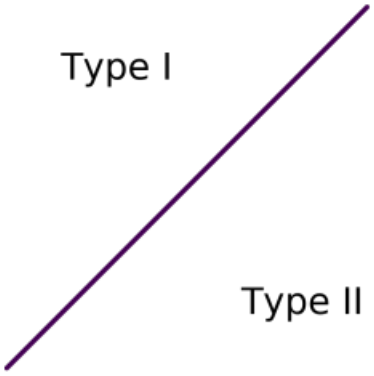

Figure 4.1: Classification

4.2.4 Output

The classification results received from the classifier module are converted into a confusion matrix and the corresponding confusion matrix is plotted. Along with the matrix the class labels for each sample in the test dataset are also displayed. The computed class labels are compared with the actual class labels and the accuracy of the classifier is calculated along with various other performance evaluation metrics such as precision, recall, f-score, Receiver operating curve-Area under curve.

4.2.5 Testing

In this phase, we check if our system conforms to the requirements and how correctly the system fulfills those requirements. In practice, some requirements will have to be deleted, or compromised, or satisfied partially. The system cannot fully satisfy all requirements. But, there are critical requirements of a system which must be satisfied.

4.3 Summary

- Detailed design comprises of the developing specification and procedures for data preparation.
- Input design is important to avoid errors in the data input process.
- Output design determines how the information is to be displayed for immediate need.

Chapter 5

Implementation

Implementation is the procedure that needs to be followed in order to convert any idea into a proper working system. It is considered the most important and critical part of any project. Technically, implementation involves the usage of an algorithm, making of a system component and developing of a software through the use of various programming languages, tools and frameworks that are readily available to us. In implementation proper care should be taken while choosing the framework, tools and programming language so that the end system is free of errors and faults.

5.1 Programming Language Selection

Python is a high level programming language that is being widely used by programmers all across the world. One of the main features of Python is that it is a general purpose programming language which helps the programmers to implement it in any application domain. It is free and open source and relies on readability of code.

There are many features provided by Python such as object-oriented, portability, extensibility and experimental programming. The automatic memory management supports numerous number of libraries along with non-static typing mechanism. It has light syntax and indentation is significant.

Some of the functionalities provided by Python include graphical user interfaces, web frameworks, multimedia, databases, networking, automation, system administration, scientific computing, text processing, image processing etc. It uses a command line interpreter which takes input from user and gives result based on the input received from the user in a sequential manner. Thus, making it user friendly, compatible and easier to use for any kind of application.

5.2 Platform Selection

5.2.1 Linux(Ubuntu 16.10)

Linux is one of popular version of UNIX operating System. It is open source as its source code is freely available. It is free to use. Linux was designed considering UNIX compatibility. Its function-

ality list is quite similar to that of UNIX.

Use of Virtual Environment

Virtual environments have the advantage that they never install the required dependencies system wide so we have a better control over the environment in which our application is running. We can choose only to install the required libraries and packages and keep the environment clean.

Virtual environment is also important as we may have multiple applications on one system with conflicting requirements.

5.2.2 Flask

Flask is a Python framework, based on Werkzeug, Jinja2 and inspired by Sinatra Ruby framework, available under BSD license. Some of the important features of Flask are:

- built-in development server and fast debugger
- integrated support for unit testing
- RESTful request dispatching
- Jinja2 template
- support for secure cookies (client side sessions)
- WSGI 1.0 compliant
- Unicode based

5.3 Libraries Required

- **scikit-learn**: scikit learn is a wide python library which practices machine learning, cross validation of data, and preprocessing of data. It is built on NumPy and SciPy.
- **NumPy**: NumPy's main object is an multidimensional array, it holds an array data structure. NumPy is a core python library which contains a collection of tools and techniques. One of these tools is multi dimensional object.

- **Maplotlib**: It is a library extensively used for visualizing the data.
- **SciPy**: It is a library which contains functions to perform more complex calculations, particularly scientific computations.
- **Pandas**: Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.
- **Beautiful Soup**: Beautiful Soup is a Python package for parsing HTML and XML documents. It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping.
- **Text Blob**: TextBlob is a Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.
- **Text Stat**: Python package to calculate statistics from text to determine readability, complexity and grade level of a particular corpus.

5.4 Methodology

Following are the steps that we have followed to implement the methodology:

5.4.1 Data Collection

We used Stack Exchange Data Explorer for running arbitrary queries against public data from the Stack Exchange network and also gathered data dumps of random timeframe from StackExchange archives.

The dataset consists of posts from Stackoverflow website which can be obtained in .csv format from StackExchange archives. We have merged, transformed, and preprocessed our data to fill in missing values and to deal with categorical variables.

At present, Stackoverflow consists of over 15M questions and 24M answers with over 64M comments. Owing to the huge dataset to deal with, we decided to take a subset of these posts around 60,000 questions which were tagged as python or python-2.x or python-3.x.

5.4.2 Data Visualization

Exploratory Data Analysis (EDA) is an approach to analyze the data set with respect to their characteristics and present it in the form of visual methods through the use of various kinds of graphs.

- The libraries that we used for EDA implementation are pandas and seaborn.
- Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.

5.4.3 Feature Engineering

It is performed based on extensive literature survey in terms of research questions. We formulate the following four research questions:

- **Who provide answers to the question on stack overflow**
  In StackOverflow, there are many questions from various topics related to programming. There are about 10M questions, 17M answers, 4.5M users and 42K tags in the StackOverflow till May 31, 2015. We try to identify answerer based on the metrics such as activeness and the domain knowledge of a particular user in a specific domain
- **How much expertise does the answerer has in (i) Same domain (ii) Dierent domain of the question?**
  The knowledge of the answerer can be derived from the metrics such as number of up-votes, reputation and percentage of accepted answer.
  As the knowledge of the answerer increases, the expertise level also increases, consequently the respective answers are likely to be accepted.Associated features are :
  - **ANSWERER SCORE**: a predefined attribute in StackOverflow dependent upon number of upvotes.
  - **REPUTATION**: a predefined attribute in StackOverflow dependent upon user activity and expertise.
  - **ACCEPTED ANSWER**: the percentage of accepted answer to the total number of answers given by a particular user.

- **To what extent the answer is relevant to the question asked?**
  We find the compatibility of the answer with the question using various topics to meet the satisfaction level of the asker.
  We calculate the relevancy of the answer given by each answerer to the question asked:
  TOPIC RELEVANCY or **TOPICAL SIMILARITY**: makes use of term frequency (TF) and inverse document frequency (IDF) which indicates how important a word is to a document.

- **How the BEST answer is accepted amongst a set of answers?**
  We compile the answer of above three research questions, which acts as the baseline in order to answer this research question.
  Based on a posts feature vector that we have extracted, the answer is classified as accepted or not. We employed NaiveBayes and Decision tree for the task of binary classification.
  Apart from this we try to implement other contextual features such as:

  – **TIME SPAN OF ANSWER**: how much time it took for a particular answer to be posted for a question.

  – **ANSWER SCORE**: calculated by difference between number of upvotes and downvotes.

  – **SENTIMENT ANALYSIS** (for comments in the measure of polarity): takes into consideration whether a comment is positive or negative statement to imply favourability towards the answer given.

  – **READABILITY**: makes use of *Flesch reading ease* score where higher score indicates that the answer posted is structured well and neat and can be read easily and vice versa.

5.4.4 Classifier modeling

We model a prediction system as a binary classifier, which classifies an answer as an accepted answer or not an accepted answer.

- Gaussian Naive Bayes
- Decision tree
- Dummy classifier

Guassian Naive Bayes: The model assumes that answers are Gaussian distributed in terms of their acceptance.

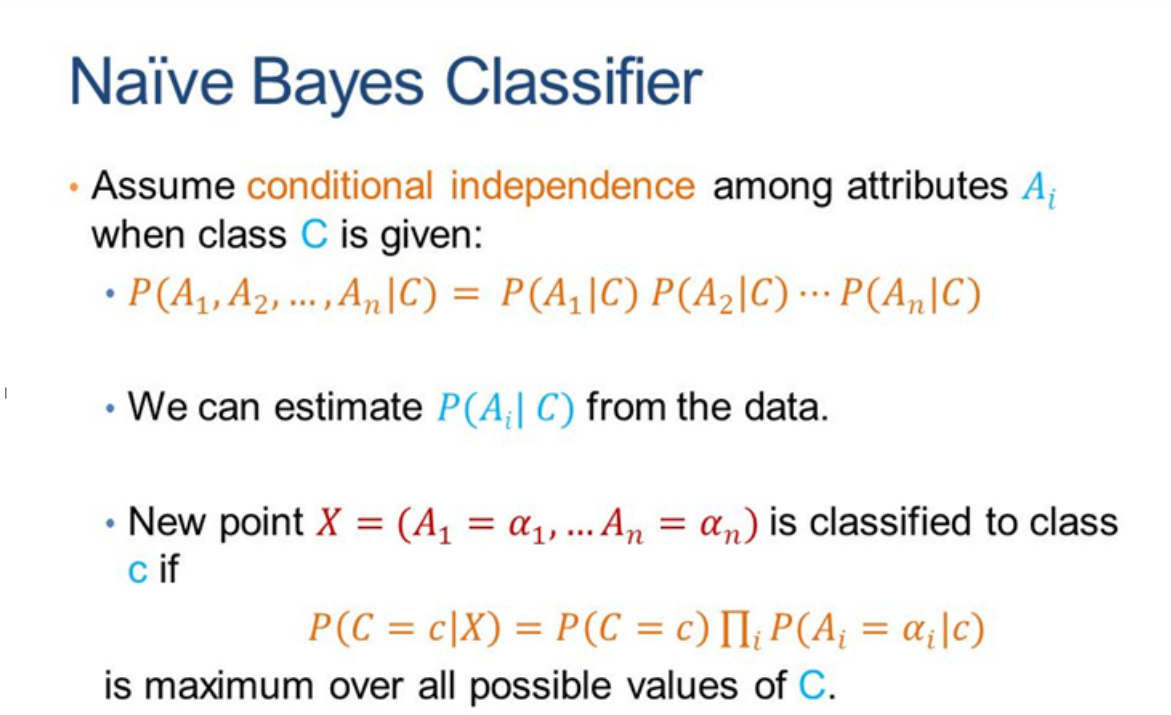

Figure 5.1: Naive Bayes Classifier

Decision tree: A decision tree is a set of rules used to classify data into categories. The key idea is that the procedure to create a decision tree is recursive. For a set (S) of observations, the following algorithm is applied:

If every observation in S is the same class or if S is very small, the tree becomes an endpoint, labeled with the most frequent class.

If S is too large and it contains more than one class, find the best* rule based on one feature (e.g. ”isweight > 150?”) to split it into subsets, one for each class.

5.4.5 Performance Evaluation Metrics

These evaluation measures are generally used to assess the performance of the classification where the dataset is imbalanced as we have.

- Accuracy
- Precision
- Recall
- F1 score
- ROC-AUC curve
- Confusion matrix

F1 score:
This is a weighted average of the true positive rate (recall) and precision. We consider F1 score as our measure of test accuracy, which considers precision and recall to compute the score.

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Figure 5.2: F-score

ROC Curve:
This is a commonly used graph that summarizes the performance of a classifier over all possible thresholds.

AUC:
This area equals the probability that a randomly chosen positive example ranks above a randomly chosen negative example.

Confusion matrix:

It is the description of the performance of the classification model on a set of test data for which the true values are known in a table format.

Confusion Matrix and ROC Curve

		Predicted Class	
		No	Yes
Observed Class	No	TN	FP
	Yes	FN	TP

TN	True Negative
FP	False Positive
FN	False Negative
TP	True Positive

Model Performance

Accuracy	= (TN+TP)/(TN+FP+FN+TP)
Precision	= TP/(FP+TP)
Sensitivity	= TP/(TP+FN)
Specificity	= TN/(TN+FP)

Figure 5.3: Confusion Matrix

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}$$

$$Precision = \frac{T_p}{T_p + F_p}$$

$$Recall = \frac{T_p}{T_p + T_n}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Figure 5.4: Formulae

5.5 Summary

This chapter deals with the various techniques used in the development of the project,starting with the language and platform selection to finally explain the entire process of implementation steps.Using the above methodologies we have developed predictive models based on the features described in feature engineering. The results are assessed and the performance of the model is evaluated using the aforementioned metrics .

Chapter 6

Testing

6.1 Software Testing

To deliver evidence about the excellence of the product or system under test software testing used.From an unbiased view, it tries to establish the quality of the product.

The resolution of software testing is to find the bugs or errors or in the program.Because of being an iterative process,a bug identification can lead to illumination of another bug.Because of the countless number of possible examinations for any simple software component, all software components use same strategy to select test that are feasible for the available time and resources.

6.1.1 White Box Testing

Basically white box testing is a method of software testing that tests the working of the program or an application rather than functionality.Path testing is an obvious example.

It is basically used in spaces where a black box testing cannot reach.For redundancy that occurs in regression of our project, white box testing is used. Since errors can occur anywhere in classification code ,redundancy tests are done.

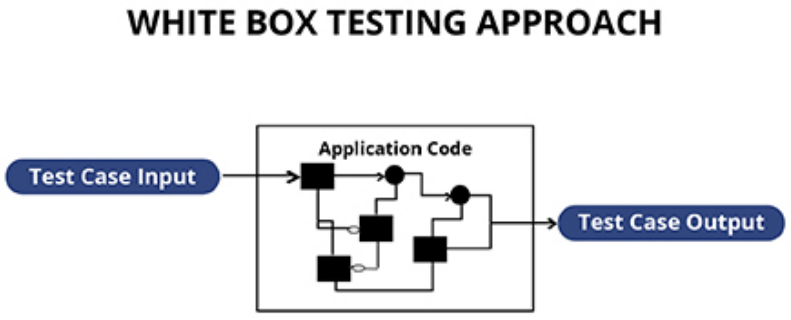

Figure 6.1: White box testing

6.1.2 Black Box Testing

In black box testing the function of the black box is understood completely in terms of inputs and outputs in-spite of not knowing the content or implementation of black box . Specification testing is also called as functional testing because the program is considered as a function that maps values from its input domain to values in its output range. Test cases of Specification based testing has two specific advantages:

- They are not dependent of how the software is implemented.In case if the implementation changes , the test cases are still useful.
- Reduction in the overall project development interval because both the test case development and implementation can occur in parallel.

6.2 Levels of Testing

There are normally 3 levels of testing:unit testing,integration testing and system testing. From the point of view of customers there are 2 different levels:Low-level testing,High-level testing.

- Low level testing is a set of tests for different level components of software application.
- High level testing is a set of tests for the application as a whole.

6.2.1 Unit Testing

In the method individual units of source code,program module associated control data,other procedures are tested to find their fit for use.Specific functionality of the developed subunits are tested.The more appropriate testing at unit testing is structural testing.This testing is followed by integration testing.Integration testing basically clubs all the modules tested for unit testing and perform the tests defined in integration test plan to them.This is then followed by system testing.

6.2.2 Integration Testing

Integration testing is the second level of testing.As mentioned before this combines all the modules of unit testing and tests them.Its main goal is to test whether there are any problems in the integration

of different modules.

6.2.3 System Testing

System testing basically evaluates whether a system meets the requirements specified before.For example , a system testing may include inputing values in the input fields,printing results ,format of the results etc.

It also tests the behaviour of system as specified by the customer.It not only tests the requirements in software/hardware specification but also tests beyond that.This is then followed by acceptance testing.

6.2.4 Acceptance Testing

As the name itself suggests acceptance testing is used to test the system compliance for the requirements.Hence basically the system is tested for acceptability.It is the last test that is performed before making the system.

6.3 Unit Testing of Module

Execution of the prediction system is tested for various conditions and the test cases are tabulated as follows:

Test Case ID	Unit Test Case P 1
Description	To test the predictor with input values
Input	86.1,0,90,0.024822,810,0
Expected Output	0
Actual Output	0
Remarks	The system performed as expected.

Unit Test Case 1

Test Case ID	Unit Test Case P 2
Description	To test the predictor with input values
Input	76.22,1,684,0.013187,82573,0
Expected Output	0
Actual Output	0
Remarks	The system performed as expected.

Unit Test Case 2

Test Case ID	Unit Test Case P 3
Description	To test the predictor with input values
Input	47.12,2,79964,0,300,0
Expected Output	1
Actual Output	1
Remarks	The system performed as expected.

Unit Test Case 3

Test Case ID	Unit Test Case P 4
Description	To test the predictor with input values
Input	106.67,0,21,0.06431,670,0
Expected Output	0
Actual Output	0
Remarks	The system performed as expected.

Unit Test Case 4

6.4 Results

Q_ID	TAG	A_ID	U_REPUTATION	Q_Title	Q_BODY	A_BODY	TIME_DIFF	Accepted_Answer_ID	A_SCORE	COMMENT
23143135	python	23143351	90	How to ignore \n in regular expressions in python?	<p>So i have a regex telling if a number …	<p>The regex in the quest…	810	23143210	0	nan
23143135	python	23161101	684	How to ignore \n in regular expressions in python?	<p>So i have a regex telling if a number …	<p>Yeah, that $ matching o…	82573	23143210	1	nan
23143135	python	23143210	79964	How to ignore \n in regular expressions in python?	<p>So i have a regex telling if a number …	<p>From the <a href="htt…	300	23143210	2	Actually worked, thank…
23143135	python	23143310	21	How to ignore \n in regular expressions in python?	<p>So i have a regex telling if a number …	<p>regex ahould be <c…	670	23143210	0	nan

Figure 6.2: Test case- Sample question with 4 answers

```
The probabilities of each answer on non-accepted and accepted classes are :
        0.0       1.0
0  0.078278  0.921722
1  0.078317  0.921683
2  0.072837  0.927163
3  0.078294  0.921706
KweriME predicts the best answer to be answer number = 3 out of given 4 answers
```

Figure 6.3: Probabilities of prediction for each answer in Test case

```
Prediction accuracy of Decision Tree is 82.60
F1 score of Decision Tree Testing is 0.89
The confusion matrix for Decision Tree testing is
 [[142  65]
 [109 684]]
Decision Tree AUC = 0.77
             precision    recall  f1-score   support

        0.0       0.57      0.69      0.62       207
        1.0       0.91      0.86      0.89       793

avg / total       0.84      0.83      0.83      1000
```

Figure 6.4: Decision Tree Performance

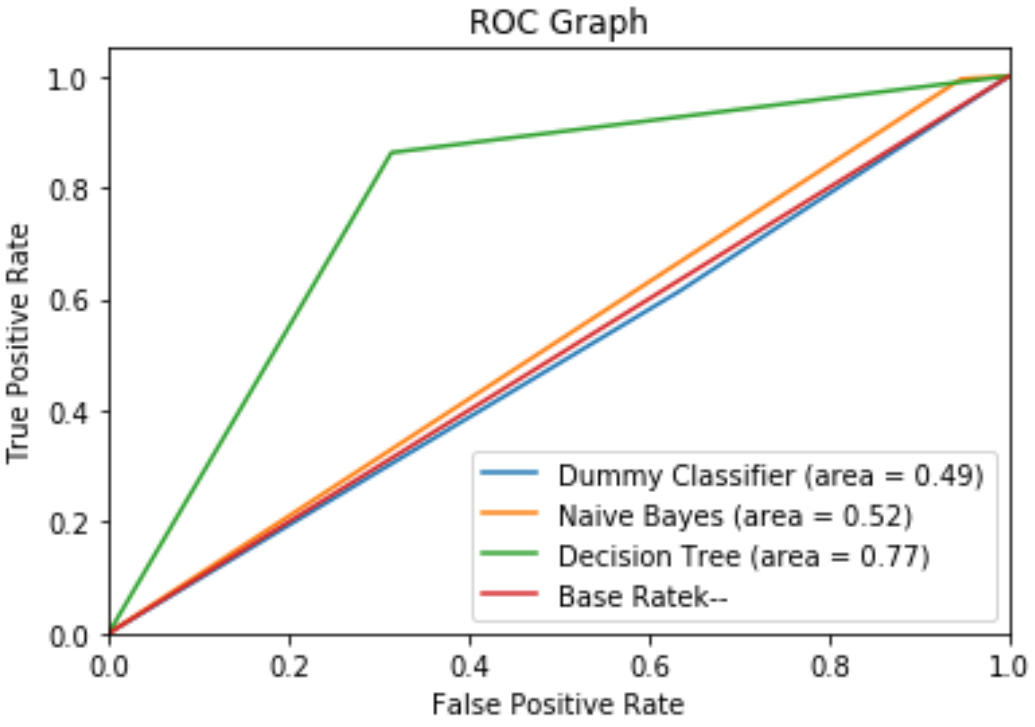

Figure 6.5: Receiver Operating Curve-Area Under Curve

	Accuracy	F1 score	AUC
Dummy Classifier	56.20	0.69	0.49
Naïve Bayes	80.00	0.89	0.52
Decision Trees	82.60	0.89	0.77

Figure 6.6: Summary of performances of classifiers

6.5 Summary

Testing plays an important part in software development. Software should be tested for checking errors, faults, bugs etc. and thus, it should be fixed accordingly in order to conform to the client's requirements. Testing is conducted in phases which are- unit testing, integration testing and system testing. Each phase plays an important part and helps in producing the correct and accurate output.

Chapter 7

Analysis & Conclusion

7.1 Analysis

As a part of the prediction stage in our system, we initially split the dataset obtained to training and test data following the 0.8:0.2 ratio i.e.,80% of the information was utilized as preparing information and the last 20% was used as test data. In the binary classifier we fed the data into the Decision tree algorithm and modeled the classifier. To this modeled classifier we fed in the test information and the class marks for the test information are anticipated .We obtained an accuracy of 83%.The Summary of our analysis is as shown below:

- Initially we ran all our models with all features
- Furthermore, we also tried removing features of StackOverflow metrics - Answer Score and Answerer Score in-order to understand the effect of contextual features
- Summarizing the result we see that apart from answer specific features such as Answer Score and Answerer Score,contextual features such as Readability and Similarity play an important role in classifying an answer as an Accepted Answer.
- We considered contextual features which are the features that address the relation between the question and answers, time lag between the posting time of question and the time of the response content and features which are the analytical features of the answer itself as well as sentimental analysis of the comments. We found that contextual features play a major role in prediction decisions.

7.2 Conclusion

We study and analyze the answers with their questions to predict whether the answer will get accepted or not. We perform an extensive empirical analysis on the retrieved dataset to identify and extract features and perform correlations between various variables to know the causality.
Our findings will suggest that:

- Prior involvement of the answerer on question tags and topics increases the chance to give the answer for that question.
- Expertise will increase the chance in acceptance of the answer.

- Topical compatibility between the question and answer increases the satisfaction of asker or community with that answer.

Armed with this observation, we have used classification algorithms to predict acceptability of the answer by the asker or community. So, the outcome of this study lies around predicting the acceptance of the answer as the best answer and various other performance metrics and accuracy ratios that deal with it.

7.3 Limitations

- The results of the study were limited to the area in which research data was collected.
- This research takes into consideration only those predictors of accepted answers that are in the range of our study.

7.4 Future Work

- First, classifier model must be designed in such a way that the class imbalance problem doesn't influence the accuracy of both the classes and so overall accuracy keeping in view PRC area.
- Future researchers can examine the same correlations by carrying out longitudinal research study i.e, doing research on data which is gathered over a long span of time.
- Future work can also consider including more factors from the database that could have more effect on deciding answer acceptability rate, for example, those based on temporal factors w.r.t. questions, answers and users etc.
- Ensemble learning can be devised to model the suggested problem in order to improve the accuracy.

7.5 Summary

In this Chapter we discussed the analysis of the project which we developed. The limitations of the project are looked at and future scope shows what developments and improvisations can be made to get better results.

Bibliography

[1] Yuhua Lin, Haiying Shen *, SmartQ: A Question and Answer System for Supplying High-Quality and Trustworthy Answers Journal of Latex Class* . Files Vol. 14, NO. 8, August 2015.

[2] Tirath Prasad Sahu, Naresh Kumar Nagwani, Shrish Verma *, Selecting Best Answer: An Empirical Analysis on Community Question Answering*. Sites Date of publication : August 26 , 2016 Digital Object Identifier 10.1109/ACCESS.2016.2600622