A Project Report on

SENTIMENTAL ANALYSIS FOR REPORT MANAGEMENT SYSTEM USING MACHINE LEARNING ALGORITHMS

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

In

Computer Science and Engineering

Submitted by

A. Hemanth (16H51A0566)

M. Henin (18H51A05A3)

M.V.S Rohith (18H51A05D2)

Under the esteemed guidance of

Mr. G. Ravi Kumar

(Assistant Professor)

Department of Computer Science and Engineering

CMR College Of Engineering & Technology

(An Autonomous Institution, Approved by AICTE, Permanently Affiliated to JNTUH, Accredited by NAAC '+A'.)

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2018- 2022

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project phase-1 report entitled **"Sentimental Analysis for Report Management System Using Machine Learning Algorithms"** being submitted by **A. Hemanth** ***(16H51A0566),*** **M. Henin** ***(18H51A05A3),*** **M.V.S Rohith** ***(18H51A05D2),*** in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out by him/her under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mr. G. Ravi Kumar
Assistant Professor
Dept. of CSE

Dr. K Vijaya Kumar
Professor and HOD
Dept. of CSE

Submitted for viva voice Examination held on __________

External Examiner

ACKNOWLEDGEMENT

With great pleasure I want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

I am grateful to **Mr. G. Ravi Kumar,** Assistant Professor, Department of Computer Science and Engineering for his valuable suggestions and guidance during the execution of this project work.

I would like to thank **Dr. K. Vijaya Kumar**, Head of the Department of Computer Science and Engineering, for his moral support throughout the period of my study in CMRCET.

I am highly indebted to Major **Dr. V.A. Narayana**, Principal CMRCET for giving permission to carry out this project in a successful and fruitful way.

I would like to thank the Teaching & Non- teaching staff of Department of Computer Science and Engineering for their co-operation

Finally, I express my sincere thanks to **Mr. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care. I sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

A. Hemanth (16H51A0566)

M. Henin (18H51A05A3)

M.V.S Rohith (18H51A05D2)

ABSTRACT

Academic industries used to accumulate comments from the scholars at the precept factors of direction which include arrangements, contents, punctual, skills, appreciation, and gaining knowledge of enjoying. The remarks are gathered in terms of every qualitative and quantitative core. Recent techniques for opinion mining use manual strategies and it is aware totally of the quantitative comments. So, the assessment can't be made via deeper analysis.

In this, we boom a student remarks mining tool which applies text analytics and sentiment evaluation technique to provide teachers a quantified and deeper analysis of the qualitative remarks from college students for the purpose to improve the students for knowing their learning experience. We have accumulated feedback from the students and then textual content processing is accomplished to smooth the statistics.

These subjects are extracted from the pre-processed file. Feedback approximately every challenge rely on are accumulated and made as a cluster. Classify the remarks by the usage of sentimental classifier and observe the visualization techniques to symbolize the views of college students.

TABLE OF CONTENTS

List Of Figures

CHAPTER 1
INTRODUCTION

1.1 BASIC INTRODUCTION

Sentiment analysis is an interdisciplinary field that crosses natural language processing, artificial intelligence, and text mining. Since most opinions are available in the text format and its processing is easier than other formats, sentiment analysis has emerged as a subfield of text mining. It generally recognizes opinions of people expressed in text. The opinions could be judgments, evaluations, affective (or emotional) states, beliefs, or wishes. Sentiment analysis appeared in the literature in 1990 for the first time and then it became a major research topic in 2000. Classifying the polarity of a given text as positive or negative is the basic task of sentiment analysis. Due to its many aspects it is often referred to with different names such as opinion mining, sentiment classification, sentiment analysis, and sentiment extraction. It is widely believed that Sentiment analysis is needed and useful. It is also widely accepted that extracting sentiment from text is a hard semantic problem even for human beings. Additionally, sentiment analysis is domain specific, therefore the polarity of some terms depends on the context in which they are used. For example, while “small” for “size” as a feature in the electronic products is positive, in agricultural products such as fruit it has a negative polarity.

Sentiment analysis is used in different domains such as shopping, entertainment, politics, education, marketing, and research and development. This paper focuses on sentiment classification in social domains from the technical perspective, two main approaches for sentiment analysis are Bag of Words (BOW) and Feature Based Sentiment (FBS). In the BOW approach, each document is seen as a set of words. As a result, the syntactic and semantic information between words are lost. The BOW approach is not useful when opinions about products and their features have to be. In such cases, it is required to extract features. FBS has emerged as an approach for analyzing the sentiments of products and their features.

1.2 RESEARCH CHALLENGES

The main challenges that are faced by Opinion mining and sentiment analysis are the following:

1)Detection of spam and fake reviews: The web contains both authentic and spam contents. For effective Sentiment classification, this spam content should be eliminated before
processing. This can be done by identifying duplicates, by detecting outliers and by considering reputation of reviewer [1].

2)Limitation of classification filtering: There is a limitation in classification filtering while determining most popular thought or concept. For better sentiment classification result this limitation should be reduced. The risk of filter bubble [11] gives irrelevant opinion sets and it results false summarization of sentiment.

3)Asymmetry in availability of opinion mining software: The opinion mining software is very expensive and currently affordable only to big organizations and government. It is beyond the common citizen’s expectation. This should be available to all people, so that everyone gets benefit from it.

4)Incorporation of opinion with implicit and behavior data: For successful analysis of sentiment, the opinion words should integrate with implicit data. The implicit data determine the actual behavior of sentiment words.

5)Domain-independence: The biggest challenge faced by opinion mining and sentiment analysis is the domain dependent nature of sentiment words. One features set may give very good performance in one domain, at the same time it perform very poor in some other domain.

6)Natural language processing overheads: The natural language overhead like ambiguity, co-reference, Implicitness, inference etc. created hindrance in sentiment analysis too.

1.3 MOTIVATION

According to Ramteke et al. (2012) motivation for Sentiment Analysis is two-fold. Both consumers and producers highly value "customer's opinion" about products and services. Thus, Sentiment Analysis has seen a considerable effort from industry as well as academia.

The Consumer's Perspective:

The consumer's perspective While taking a decision it is very important for us to know the opinion of the people around us. Earlier this group used to be small, with a few trusted friends and family members. But, now with the advent of Internet we see people expressing their opinions in blogs and forums. These are now actively read by people who seek an opinion about a particular entity (product, movie etc.). Thus, there is a plethora of opinions available on the Internet. From a consumers' point of view extracting opinions about a particular entity is very important. Trying to go through such a vast amount of information to understand the general opinion is impossible for users just by the sheer volume of this data. Hence, the need of a system that differentiates between good reviews and bad reviews. Further, labeling these documents with their sentiment would provide a succinct summary to the readers about the general opinion regarding an entity

The Producer's Perspective

With the explosion of Web 2.0 platforms such as blogs, discussion forums, etc., consumers have at their disposal, a platform to share their brand experiences and opinions, positive or negative regarding any
product or service. According to Pang and Lee (2008) these consumer voices can wield enormous influence in shaping the opinions of other consumers and, ultimately, their brand loyalties, their purchase decisions, and their own brand advocacy.
Since the consumers have started using the power of the Internet to expand their horizons, there has been a surge of review sites and blogs, where users can perceive a product's or service's advantages and faults. These opinions thus shape the future of the product or the service.
The vendors need a system that can identify trends in customer reviews and use them to improve their product or service and also identify the requirements of the future.

The Societies' Perspective

Recently, certain events, which affected Government, have been triggered using the Internet. The social networks are being used to bring together people so as to organize mass gatherings and oppose oppression.
On the darker side, the social networks are being used to insinuate people against an ethnic group or class of people, which has resulted in a serious loss of life. Thus, there is a need for Sentiment Analysis systems that can identify such phenomena and curtail them if needed.

1.4 PROBLEM DEFINITION

Recent approaches for feedback mining use manual methods and it focus mostly on the quantitative comments. So the evaluation cannot be made through deeper analysis.

In this project, we will develop a student feedback mining system which applies text analytics and sentiment analysis approach to provide instructors a quantified and deeper analysis of the qualitative feedback from students that will improve the students learning experience.

1.5 OBJECTIVES

The main objective of this project is to Build an effective feedback system by using SVM classifier in the proposed system.

Since the previous Navis bayes classifier has low accuracy rate, thus to increase the accuracy rate we use the Support Vector Machine algorithm.

CHAPTER 2
BACKGROUND WORK

2.1 DOMAIN INTRODUCTION:

Students provide feedback in quantitative ratings and qualitative comments related to preparation, contents, delivery methods, punctual, skills, appreciation, and learning experience. The delivery methods and preparation component refer to instructor's interaction, delivery style, ability to motivate students, out of class support, etc. preparation refers to student's learning experience such as understanding concepts, is been developing skills, applying acquired skills, etc. The paper correction refers to correction of mistakes and providing solutions to overcome it. The punctual refers to the class timing and assignment or record submission. The appreciation refers to the comments given when something is done perfectly. Analyzing and evaluating this qualitative data helps us to make better sense of student feedback on instruction and curriculum. Recent methods for analyzing student course evaluations are manual and it mainly focuses on the quantitative feedback. It does not support for deeper analysis. This paper focus on providing qualitative and quantitative feedback to analyze and provide better teaching to improve the student's performance.

Sentiment analysis is an interdisciplinary field that crosses natural language processing, artificial intelligence, and text mining. Since most opinions are available in the text format and its processing is easier than other formats, sentiment analysis has emerged as a subfield of text mining.

It generally opinions of people expressed in text. The opinions could be judgments, evaluations, affective (or emotional) states, beliefs, or wishes. Sentiment analysis appeared in the literature in 1990 for the first time and then it became a research topic in 2000. Classifying the polarity of a given text as positive or basic task of sentiment analysis. Due to its many aspects, it is often referred to with names such as opinion mining, sentiment classification, sentiment analysis, and extraction. It is widely believed that Sentiment analysis is needed and useful.

It accepted that extracting sentiment from text is a hard semantic problem even for beings. Additionally, sentiment analysis is domain specific, therefore the polarity terms depends on the context in which they are used. For example, while “small” a feature in the electronic products is positive, in agricultural products such as fruit negative polarity. Sentiment analysis is used in different domains such as entertainment, politics, education, marketing, and research and development.

This focuses on sentiment classification in social domains from the technical perspective, two main approaches for sentiment analysis are Bag Of Words (BOW)
and Feature Based Sentiment.

In the BOW approach, each document is seen as a set of words. As a result, the syntactic and semantic information between words are lost. The BOW approach is not useful when opinions about products and their features have to be . In such cases, it is required to extract features. FBS has emerged as an approach for analyzing the sentiments of products and their features. The results of sentiment classification are presented in various formats in different domains: positive/negative, like/dislike, recommended/not recommended, good/bad, buy/don’t buy, excellent/boring(film),support/against, favorable / unfavorable, bullish/bearish, or optimistic / pessimistic.

2.2 REVIEW OF LITERACY SURVEY:

In this section, we present a review of the existing and related works on Opinion Mining (OM) and Sentiment Classification (SC) proposed in the literature. The state-of the-art theories and models in today’s literature are also presented. This review is categorized in the following seven categories as shown in Figure 1. It outlines the various techniques used for Opinion.
Mining and Sentiment Classification from the existing literature. The different techniques used to mine opinions, classify sentiment of mined items and features, as well as the strength of the sentiment are reviewed; and compared and contrasted against each other.

A. Item Extraction:

we analyze all the frameworks that are related to item extraction. Specifically two papers are of importance as they focus on this topic in detail. Item extraction is the process of extracting the subject matter where opinions have been expressed on in customer reviews.
It is also commonly referred to as 'Opinion Extraction'. The term 'subject matter' is also commonly referred to as a situation or a product. It is an important task as it the beginning stage in the task of OM. In contrast to sentiment classification, opinion extraction aims at producing useful richer information for in-depth analysis of opinions. When evaluating these frameworks, we rely on the questions listed in Table 1. to these questions reflect the advantages and disadvantages of the researched framework.

Since this problem was discovered, limited research has been undertaken to attempt to solve. It, for example the work by Kobayashi et al.and Gamon et al. Kobayashi and his team presented a method for opinion extraction in a structured form.It also discussed the most effective way to structure customer reviews in web documents an focused on extracting subject/aspect evaluation relations, and extracting subject/aspect-asp relations, using a machine learning-based method, which is portable across domains.

It addressed the task of opinion extraction by combining contextual clues and context independent statistical clues using a machine-learning technique, 'boosting-based algorithm by Kudo (04) implemented as the package BACT. Experiments were carried out and evaluation was conducted using 5 fold cross validation on all data in the aspects of recall and precision..

B. Feature Extraction:

Feature extraction is the identification of features of products customers have expressed their opinions on their reviews and feedbacks. Features refer to product features, product attributes, and/or product functions like the picture quality of the Canon IXUS 10, or the interior design of a Ford territory, or the service of hotel staff. It is essential to readers that the features of the reviewed products are known as their areas of importance in different products may differ from people. For example, a reader might be interested in the cleanliness of the hotel room, whilst the reviewer is more concerned with quality of the customer service of the hotel staff.

They believed these details are essential for addressing the task. Assuming opinions can be represented as tuples (Subject, Attribute, Value), they addressed the task of opinions extraction by employing a computational method for tuples extraction. Machine-learning based techniques are then applied to the main task of opinion extraction, which was then decomposed into two subtasks: Extraction of attribute value pairs related to a product (where an attribute represents one aspect of a subject and the value is a specific language expression that qualifies or quantifies the aspect); and Determination of its subjectivity on the opinion as a whole. The proposed method had yield better outcome.

classification is useful for various applications, such as assisting behavioral scientists and improving doctor patient interaction. Their objective was to predict the reviewer's most likely state of mind when the post was written using a machine learning approach to identify a set of features to be used for the learning process. Their experimental results had showed a consistent modest improvement on the naïve baseline. Table 2 shows that Hu and Mishne had used.

C. Sentiment Classification in General:

Sentiment classification is the process of determining the subjectivity of a given text. In simple words, it is the task of deciding a given text expresses a positive or negative opinion about its 'subject matter' and 'subject attributes', which is also known as 'product' and 'features'. Table 3. An evaluation of Sentiment Classification Methods Papers reviewed Parameters for evaluation 2 7 11 12 the algorithm uses scoring methods from information retrieval for sentiment determination.

N NN Does the algorithm use General Inquirer (GI) lexicon. N N Y Y Does the algorithm uses WordNet? N NN Y Does the algorithm use linguistic rules? N Y N N Does the use aggregation function? N Y N N Sentiment classification assessment on opinions is done on document level rather than on sentence level as in contrast to opinion mining. presents the similarities and differences of the four papers relevant to the area of sentiment classification. Dave et al. applied various machine learning methods for the task of opinion extraction and sentiment classification and discovered several problems that have not been expected initially. They began by using structured reviews for testing and training and identifying appropriate features.

Two tests were conducted and experimental results showed that their best methods performed as well as or better than traditional machine learning methods. Sentiment classification in general is covered and the next section is ‘sentiment classifications on items’ which provides more specified and targeted information.

D. Sentiment Classification on Item:

This section is more focused on whether a given text has positive or negative connotation on its subject matter only. For example, Camera 1 has positive or negative feedback from users online. The paper by Esuli et al. is the extension of the authors’ previous work: “Esuli & Sebastiani, 2005, ‘Determining the semantic orientation of terms through gloss analysis’. It confronted the task on the decision of whether a given term has a positive connotation, or a negative connotation, or has no subjective connotation at all; thus, this problem subsumed the problem of determining orientation. This problem was tackled by testing three different variants of a semi-supervised method previously proposed. For orientation detection. Their results showed that determining subjectivity orientation was much of a harder problem than determining orientation alone. Wide coverage and its fine grain properties, obtained by qualifying the labels by means of numerical scores. The next section is sentiment classification on features, which is the next stage in sentiment classification on items.

E. Sentiment Classification on Features:

We are discussing sentiments on features in this section, as in deciding whether a given text has a Positive or Negative opinion on attributes’, which we also commonly referred to as ‘product attributes’ and/or ‘product.features’. For example, the features of Camera A being ‘buttons placement on the camera’and ‘size of screen’. Of particular relevance in this topic is the work by Ding et al.’s.

Ding discussed the problem of determining semantics of opinions expressed on product features customer reviews, rather than on the products (items) mentioned in the reviews.

F. Strength of Sentiments:

Determining the strength of sentiments is the process of decided whether a Positive opinion expressed by a text on its subject matter is Weakly Positive, Mildly Positive, or Strongly Positive, and/or whether a Negative opinion expressed is weak.

Negative, Mildly Negative, or Strongly Negative. Y N Does the algorithm categorize the sentiment strength using objective measures? Y Y In this section, the two papers of particular relevance are Popescu et al. and Wilson & Wiebe et al., whose works discuss the st the sentiments of customer reviews and feedbacks It focused on the extraction of features, identifying corresponding customer opinions about these features and their polarity. It differed from method used in as instead of only finding the key multiple reviews, it also determines the polarity and strengths of the sentiments of reviews. It is also portable across domain as shown in Table 4. OPINE extracts features such as properties, parts, features of product parts, related concepts, and parts.

Which was very different to OPINE by Popescu et al. Experiments were conducted and they had achieved significant improvements in mean-squared error over baseline using three for support vector regression. In the next section, the comparison of items and features is discussed.

2.3 ANALYZING RECOMMENDATION OF COLLEGES FOR STUDENTS USING DATA MINING TECHNIQUES

Recommendation Systems are the type of information filtering systems designed to help users to find their way through todays large information spaces. Till date Recommendation Systems are the best examples of personalization system. Recommendation Systems uses a number of different technologies. The goal of a Recommendation System is to generate meaningful recommendations to a collection of users for items or products that might interest them. Suggestions for books on Amazon, or movies on Netflix, are real world examples of the operation of industry-strength recommendation systems. The design of such Recommendation engines depends on the domain and the particular characteristics of the data available. For example, movie watchers on Netflix frequently provide ratings on a scale of 1 (Disliked) to 5 (liked). Such a data source records the quality of interactions between users and items. Additionally, the system may have access to user-specific and item-specific profile attributes such as Demographics and product descriptions respectively. Recommendation systems differ in the way they analyze these data sources to develop notions of affinity between users and items which can be used to identify well-matched pairs. Collaborative Filtering systems analyze historical interactions alone, while Content-based Filtering systems are based on profile attributes; and Hybrid techniques attempt to combine both of these designs.

Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine. Information comes in many shapes and sizes. Information extraction is used to automatically. Extracting structured information.

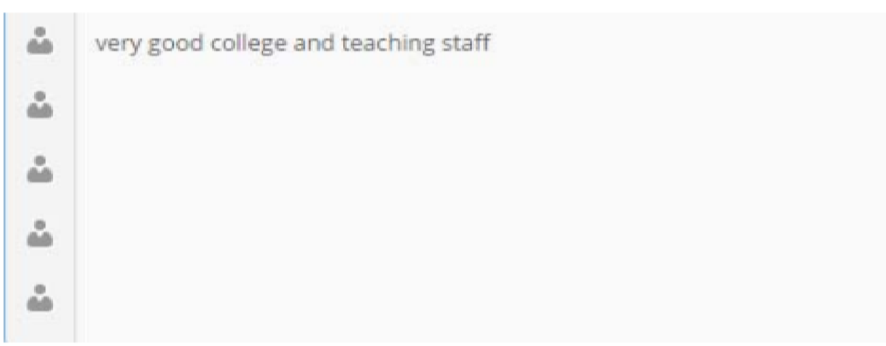


Figure 2.1 Feedback page

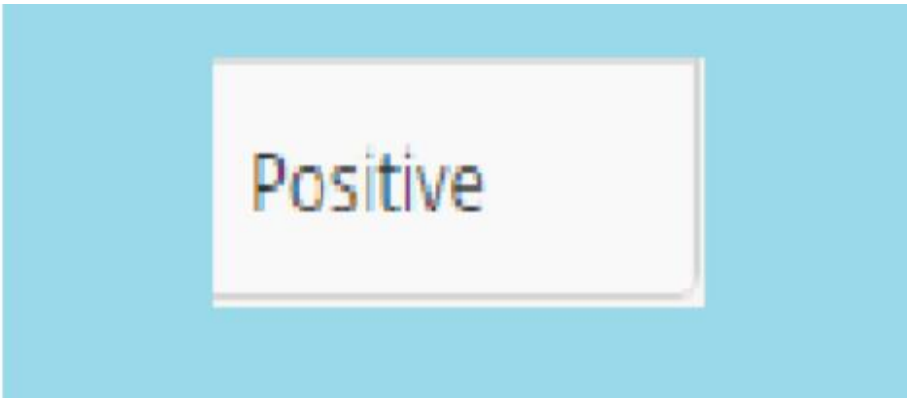


Figure 2.2 Response Page

2.4 FACULTY RATING SYSTEM BASED ON STUDENT FEEDBACKS USING SENTIMENTAL ANALYSIS

The user interacts with the front end of the application. The feedback of the faculty is collected from the student using the feedback form. We maintain attributes in the database for the Department of the teacher, Name of the student submitting the feedback, Name of the faculty about whom the student is submitting the feedback, ID of the student and the faculty, CGPA of the student, gender of the faculty and the four major attributes for evaluating a faculty which were mentioned earlier, to store the data entered by the student. All the details entered by the student are then mapped onto the respective attributes in the database.

For example, let us consider this feedback given by a student about the conduct of a teacher in the class,” Initially she was very strict! But now she is friendly”. These two sentences are tokenized and stored in a list of lists as shown below, [[’Initially’,’she’,’was’,’very’,’strict’,’!’],[’But’,’,’,’now’,’she’,’is’,’friendly’]] Once the tokenization of the sentence is done the next step is to check for the emotions described in the input. For the above case, a list is generated to store only the emotions as shown below, [’very strict’, ‘friendly’] Now, once this is done we need to find that one word which best describes the behavior of the teacher by taking into consideration all the emotions that we captured from the input. For this we have built two table of the emotions that most users use frequently and their respective values.

It is using the student feedback weight mechanism which we mention here. The faculty will have to submit short feedback of a student before the student can write feedback about them. We mainly consider 5 important factors on which we assign the weight to a particular student's feedback. They are, the student's CGPA, attendance, behavior in class, sincerity and the time spent by a student for submitting the feedback of a particular faculty. The weight calculation procedure is shown in figure 3. In our case a student's feedback can have a weightage ranging from 0.5 to 1.5. This is because no matter how a particular student is, his/her feedback cannot be completely avoided, nor can it be given a very high value like 2. If it is given a value like 2, it is equivalent of saying that the student's feedback is as valuable as two student's feedback, which is not right. Each factor is given an equal weightage value of 0.3. In case the total weightage of a feedback submitted by the student in less than 0.5, we assign it a value of 0.5. Firstly, the CGPA of the student submitting the feedback form is mapped onto the database along with the other details in it.

Word	Value
very rude	1
very poor	2
very bad	3
very strict	4
rude	5
poor	6
bad	7
strict	8
arrogant	9
not nice	10
short tempered	11
unpleasant	12
not good	13
not polite	14
doesn't understand	15
not bad	16
not strict	17
ok	18
indifferent	19
nice	20
polite	21
understands	22
good	23
very polite	24
very nice	25
calm	26
cool	27
very good	28

Figure 2.3 Feedback T-1

Word	Value
very poor	1
very bad	2
poor	3
bad	4
doesn't know anything	5
not good	6
not bad	7
must improve	8
doesn't know much	9
ok	10
not interesting	11
average	12
not great	13
not perfect	14
can improve	15
good	16
interesting	17
has improved	18
very good	19
great	20
perfect	21
superb	22
fabulous	23
awesome	24
brilliant	25
excellent	26
best	27
outstanding	28
pleasant	29
friendly	30
superb	31
fabulous	32
awesome	33
sincere	34
dedicated	35

Figure 2.4 Feedback T-2

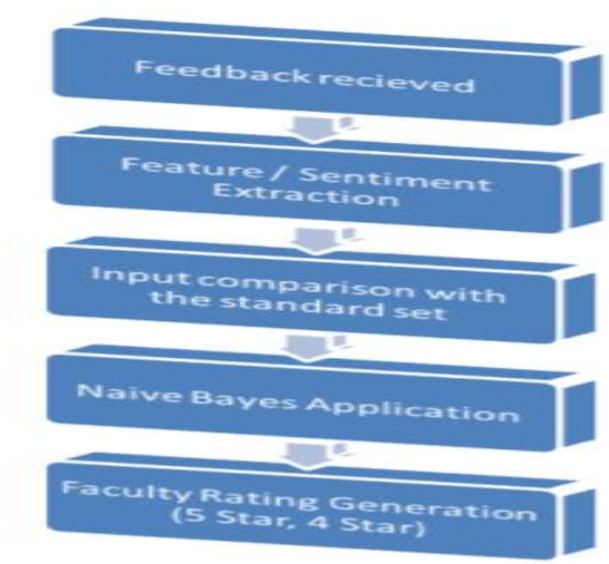


Figure 2.5 Flow Diagram

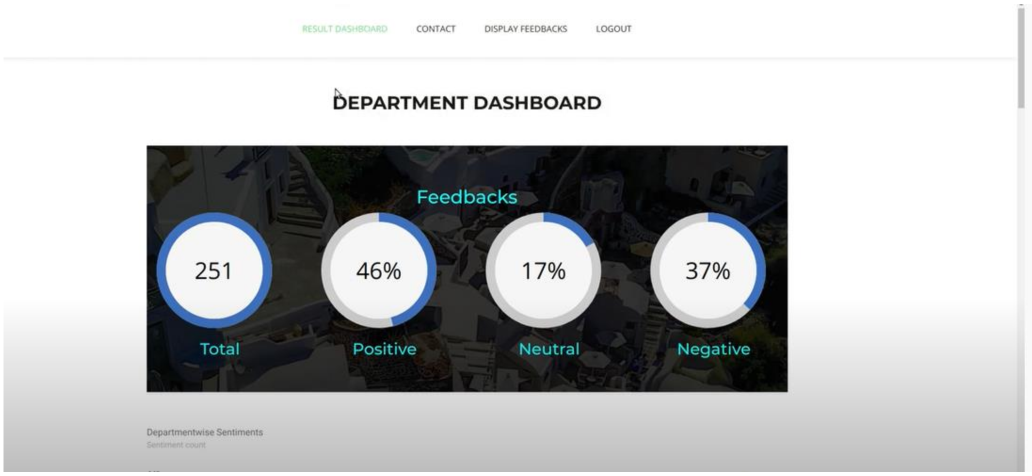


Figure 2.6 Home page

2.5 PERFORMANCE ANALYSIS ON STUDENT FEEDBACK USING MACHINE LEARNING ALGORITHMS

The paper spotlight on victimization Opinion Mining method for classifying the student's comments acquired during module evaluation survey. The mined and preprocessed datasets have been based to numerous supervised evaluations taking out rule like aid Vector Naïve Bayes (NB), Nearest Neighbor (KNN) and Neural Networks (NN) enforced [1].

It is visible currently that there is a rise of expertise availableness, the alleged statistics deluge, controlled by accomplice inflated amount of electronic motion accomplished, and additionally the revolutionary pervasive attain of IT altogether gadgets. the number one of those developments is that the supposed open statistics association, distinguished by the manner that the complete method throughout European and also the united states, governments area unit more and more e-book their information repositories for humans to admittance and use it any other pattern issues the inconceivable amount of knowledge is formed handy by electorate through participatory sensing''.

Commonplace play a practical position in booklet commentary and grumbling online, and increasingly more make use of novelty to report further in sequence.

Presently the feedback information is employed to report solely the performance of the teacher. The paper proposes ways to research the feedback information victimization data processing techniques for a higher understanding of the college, course, and student. The format of feedback varies from establishment to establishment, thus there cannot be a general technique which will appropriate all. The feedback information from the scholars is analyzed by victimization completely different data processing techniques. The feedback information is used for analyzing all the parameters thought of for feedback which might facilitate management in creating policy choices in teaching-learning method. This Paper surveys all data processing technique that is applied for analyzing feedback information.

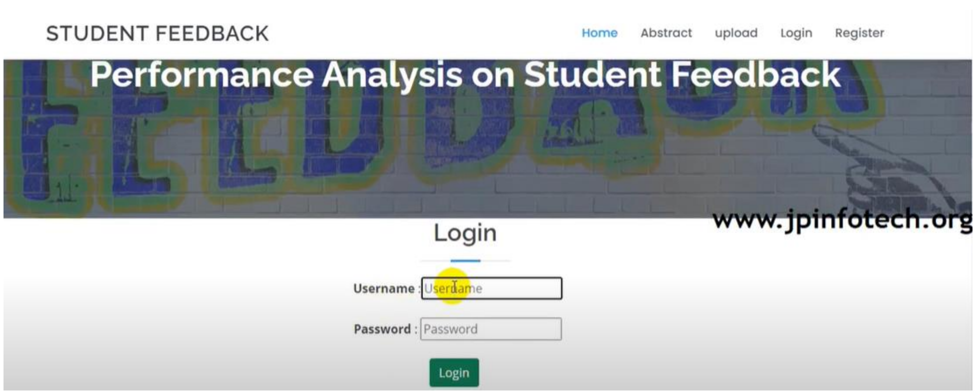


Figure 2.7 Login Page

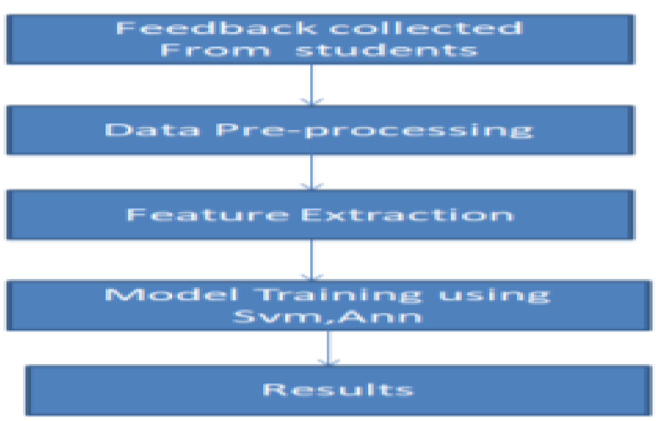


Figure 2.8 Flow chart

2.6 COMPARISON OF EXISTING MODELS PERFORMANCE ANALYSIS

Which algorithm is better for text classification, well there is no single answer and there will never be since it all comes down to use cases, for example in telemedicine it is important to classify the medicine effectively from the rest of the corpus of data. Based on the dosage we can predict whether the patient is suffering from mild or severe conditions. Thus, using SVM would be a safe bet because it is not prone to catastrophic failures along with that the algorithm is able to correlate with other elements within the corpus that help to understand the dense features in NLP resulting in sentimental analysis or machine translation, but in case of Naive Bayes, the results are not consistent enough.

Though it is more of a generalized algorithm that works effectively and cheaply when we want to classify a small corpus of data having a relatively small amount of input features, you don’t expect the inputs to be meaningfully correlated. If an algorithm is not able to capture the correlation between different words and its intensity then how can you expect it to learn “Word Order” Also, removing stop words and performing TD-IDF adds on to the misery of NB as it takes away a good chunk of raw text. There are other forms of Naïve Bayes that require hyperparameter tuning those results in much better results.

CHAPTER-3

PROPOSED SYSTEM

3.1 PROPOSED SOLUTION

- The proposed idea is to apply opinion mining to train the data set which there by will classify the fake and genuine reviews.
- Further enhancement to the classification to be done in order to increase the efficiency of the model.
- Obtained reviews which are classified as genuine to be further acknowledged as verified as per to increase genuineness among the reviews Verification of the review indicating that the review is genuine.
- Obtained reviews which are classified as fake to be further acknowledged using a cross notation. Verification of the review indicating that the review is fake.

3.2 OPINION MINING/SENTIMENTAL ANALYSIS

Sentiment analysis can be defined as analyzing the positive or negative sentiment of the customer in text. The contextual analysis of identifying information helps businesses understand their customers' social sentiment by monitoring online conversations. As customers express their reviews and thoughts about the brand more openly than ever before, sentiment analysis has become a powerful tool to monitor and understand online conversations. Analyzing customer feedback and reviews automatically through survey responses or social media discussions allows you to learn what makes your customer happy or disappointed. Further, you can use this analysis to tailor your products and services to meet your customer's needs and make your brand successful. Recent advancements in machine learning and deep learning have increased the efficiency of sentiment analysis algorithms. You can creatively use advanced artificial intelligence and machine learning tools for doing research and draw out the analysis. For example, sentiment analysis can help you to automatically analyze 5000+ reviews about your brand by discovering whether your customer is happy or not satisfied by your pricing plans and customer services. Therefore, you can say that the application of sentiment is endless. Sentiment analysis is the process of using natural language processing, text analysis, and statistics to analyze customer sentiment. The best businesses understand the sentiment of their customers. domain of understanding these emotions with software, and it’s a must-understand for developers and business leaders in a modern workplace. As with many other fields, advances in deep learning have brought sentiment analysis into the foreground of cutting- edge algorithms. Today we use natural language processing, statistics, and text analysis to extract, and identify the sentiment of words into positive, negative, or neutral categories.

3.3 IMPORTANCE OF SENTIMENTAL ANALYSIS

The most crucial advantage of sentiment analysis is that it enables you to understand the sentiment of your customers towards your brand. Your products and services can be improved, and you can make more informed decisions by automatically analyzing the customers’ feelings and opinions through social media conversations, reviews, surveys, and more. According to the survey, 90% of the world’s data is unstructured. Especially in businesses, emails, tickets, chats, social media conversions, and documents are generated daily. Therefore, it is hard to analyze all this vast data in a timely and efficient manner. Let us look at the overall benefits of sentiment analysis in detail: • Sort Data at Scale There is too much business data to analyze daily. Can you imagine sorting all these documents, tweets, customer support conversations, or surveys manually? Sentiment analysis will help your business to process all this massive data efficiently and cost- effectively. • Real-Time Analysis Is your angry customer about to churn? Is a PR crisis on social media escalating? Sentiment analysis will help you handle these situations by identifying critical real-time situations and taking necessary action right away. • Consistent Criteria According to research, customers only agree for 60-65% while determining the sentiment of the particular text. Tagging text is highly subjective, influenced by thoughts and beliefs, and also includes personal experience. Therefore, you can apply criteria and filters to all your data, improve their accuracy, and gain better insights using sentiment analysis.

3.4 ALGORITHMS DESCRIPTION

Here, we have used 4 different algorithms to perform sentiment analysis/ opinion mining. They are,

1. Naive Bayes
2. Logistic Regression
3. SVM Classifier
4. Random Forest

3.4.1 Naive Bayes:

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.It is mainly used in text classification that includes a high-dimensional training dataset.Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Hence each feature individually contributes to identify that it is an apple without depending on each other. Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem

Bayes' Theorem:

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 3.1 Bayes Formula

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence. P(B) is Marginal Probability: Probability of Evidence.

3.4.2 LOGISTIC REGRESSION:

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

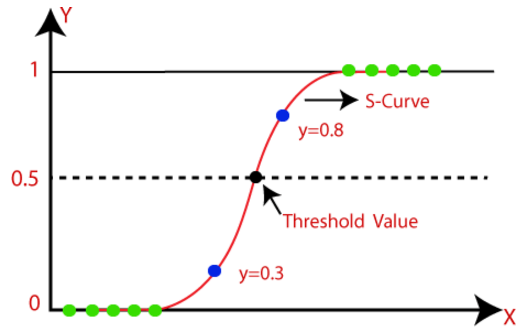


Figure 3.2 Logistic Function

3.4.3 SUPPORT VECTOR MACHINE CLASSIFIER:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

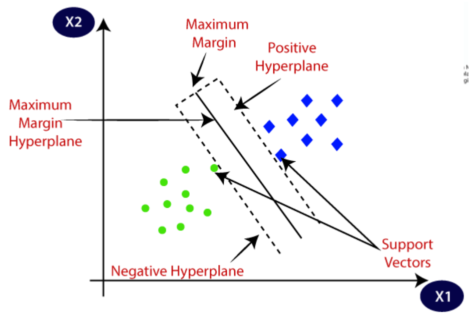


Figure 3.3 Hyper Plane

Types of SVM :

SVM can be of two types:

Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane*:* There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

How does SVM works?

Linear SVM:

The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:

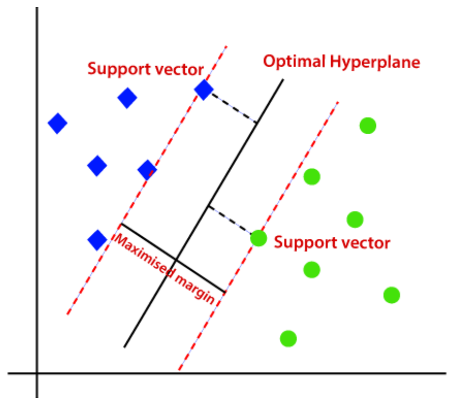


Figure 3.4 Linear SVM

So, as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image: These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.

Non-Linear SVM:

If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:

So to separate these data points, we need to add one more dimension.

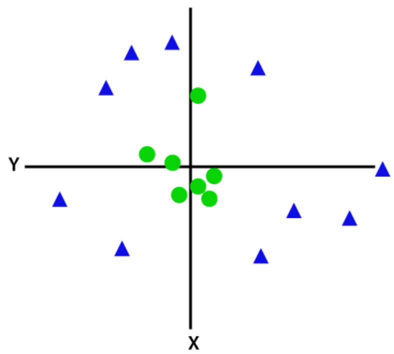


Figure 3.5 Non Linear SVM

3.4.4 Random Forest Classifier:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:

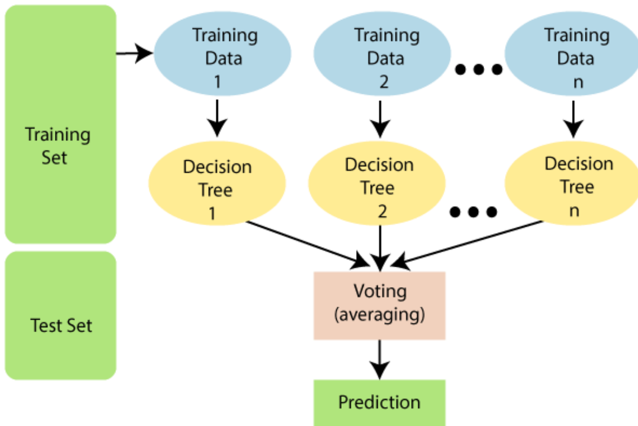


Figure 3.6 Random Forest

HOW DOES RANDOM FOREST ALGORITHM WORK?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram: Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build. Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The working of the algorithm can be better understood by the below example:

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision.

PYTHON :

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Modules Used in Project :-

Tensorflow:

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Numpy:

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas:

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib:

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn:

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

3.5 STEPWISE IMPLEMENTATION

Step 1 – Importing Libraries

```
import numpy as np
import pandas as pd
import scipy
import matplotlib.pyplot as plt
import seaborn as sns
import re
from wordcloud import WordCloud, STOPWORDS
import nltk
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer, PorterStemmer
import math
from collections import Counter
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score, roc_curve, auc, mean_squared_error
from sklearn.decomposition import TruncatedSVD, PCA
import gensim
import string
```

```
from nltk.corpus import stopwords
from nltk.stem.lancaster import LancasterStemmer  # Convert words to base form; aggressive

# Import packages that help us to create document-term matrix
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
```

(numpy, pandas for data pre-processing and matplotlib and seaborn for data visualization and nltk for text pre-processing and sklearn for model building).

Step 2 – Importing dataset and removing the unwanted dataset.

```
data = pd.read_csv("C:/Users/Admin/Desktop/Teacher Review/data/current_data.csv")
```

Data pre-processing

```
data.head()
```

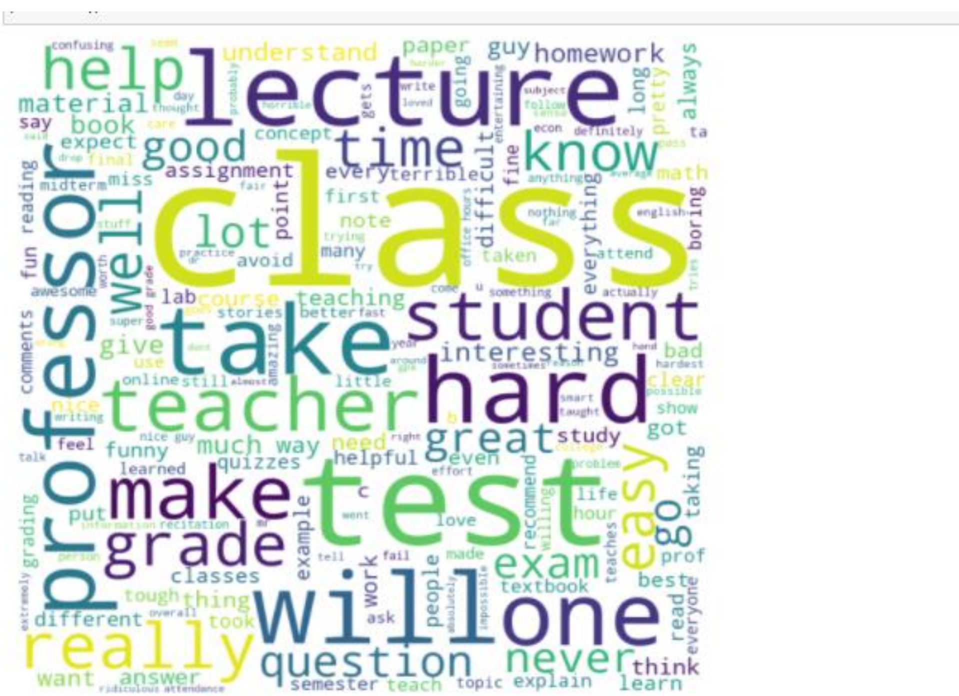
	Unnamed: 0	Review	Useful
0	0	He was boring and ruined psychology for me. No...	0.0
1	1	extremely difficult course. I thought I was go...	1.0
2	2	Hard to understand at times. Tests are ok. Not...	1.0
3	3	SO glad to be the hell out of this class	0.0
4	4	great teacher and very smart , the class is ve...	1.0

```
data = data.drop(['Unnamed: 0'],axis=1)
```

```
data.head()
```

	Review	Useful
0	He was boring and ruined psychology for me. No...	0.0
1	extremely difficult course. I thought I was go...	1.0

Step 3 – Printing the most used words for the dataset using wordCloud



Step 4 – Text Preprocessing

```
import re
import string

# remove all numbers with letters attached to them
alphanumeric = lambda x: re.sub('\w*\d\w*', ' ', x)

# '[%s]' % re.escape(string.punctuation),' ' - replace punctuation with white space
# .lower() - convert all strings to lowercase
punc_lower = lambda x: re.sub('[%s]' % re.escape(string.punctuation), ' ', x.lower())

# Remove all '\n' in the string and replace it with a space
remove_n = lambda x: re.sub("\n", " ", x)

# Remove all non-ascii characters
remove_non_ascii = lambda x: re.sub(r'[^\x00-\x7f]',r' ', x)

# Apply all the lambda functions wrote previously through .map on the comments column
data['Review'] = data['Review'].map(alphanumeric).map(punc_lower).map(remove_n).map(remove_non_ascii)

data['Review'][0]
```

(To remove all the letters, punctuation, spaces, non-ascii characters).

Step 5 – Assigning the values to the variables for model building and apply vectorizer on X values.

```
.5]: X = data.Review
     y = data['Useful']

     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

     # Initiate a Tfidf vectorizer
     tfv = TfidfVectorizer(ngram_range=(1,1), stop_words='english')

     X_train_fit = tfv.fit_transform(X_train)  # Convert the X data into a document term matrix dataframe
     X_test_fit = tfv.transform(X_test)  # Converts the X_test comments into Vectorized format
```

X is assigned to Reviews and y to Useful.

Step 6 – Apply algorithms

1. SVM
2. Random Forest Classifier
3. Logistic regression
4. Naïve Bayes
5. SVM- Linear
6. SVM – Poly
7. SVM – Gaussian
8. SVM – Sigmoid
9. Voting Classifier

CHAPTER 4

DESIGNING

4.1 UML DIAGRAMS

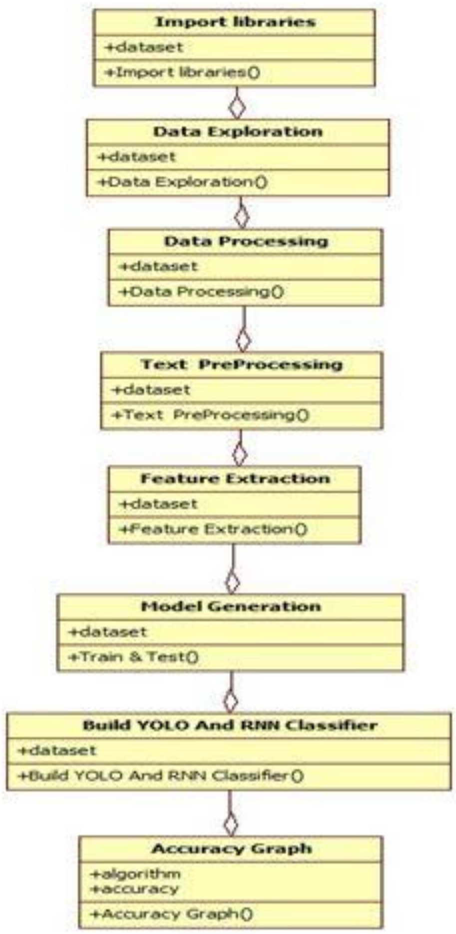


Figure 4.1 Use case Diagrams

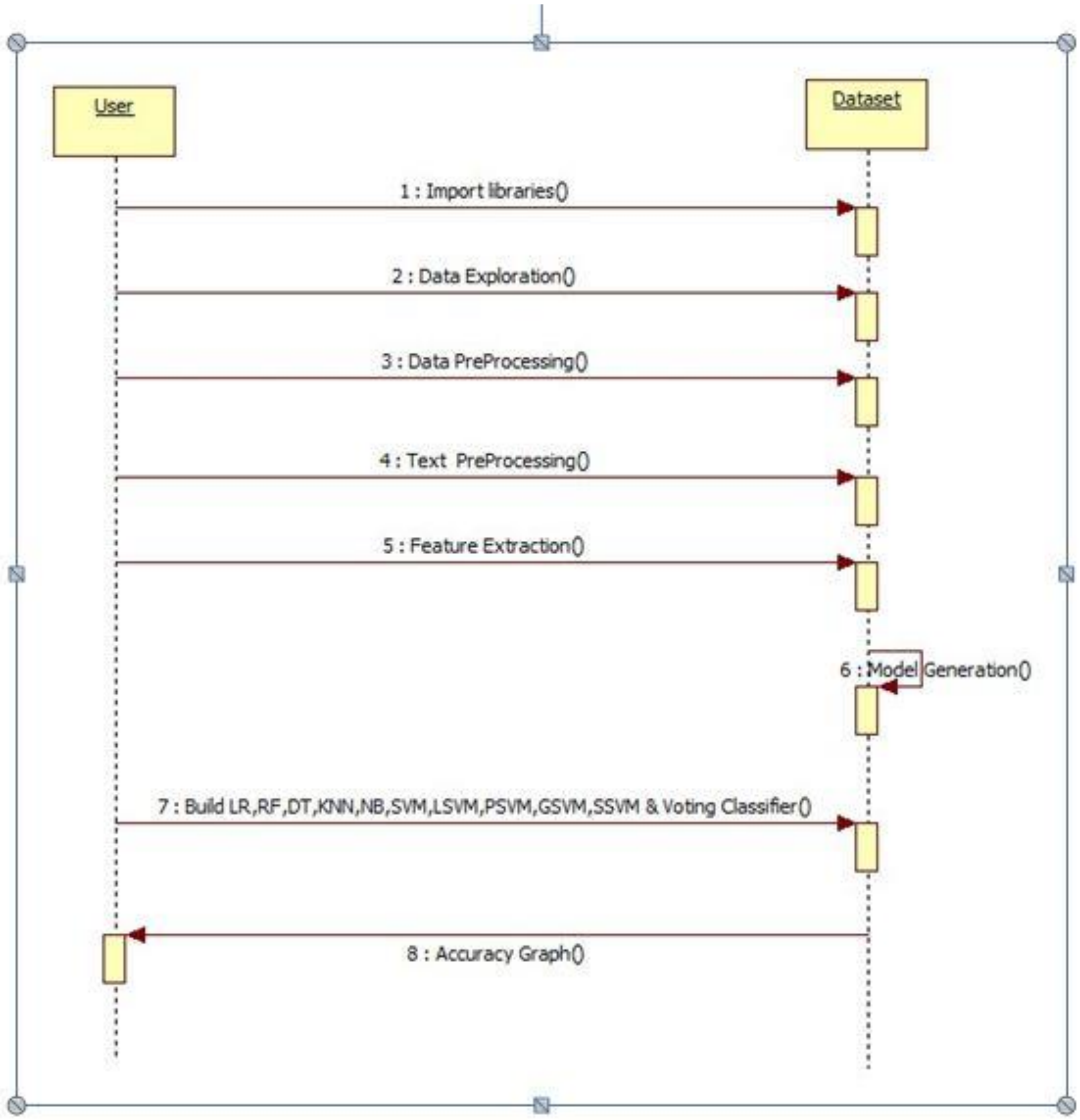


Figure 4.2 Sequence diagram

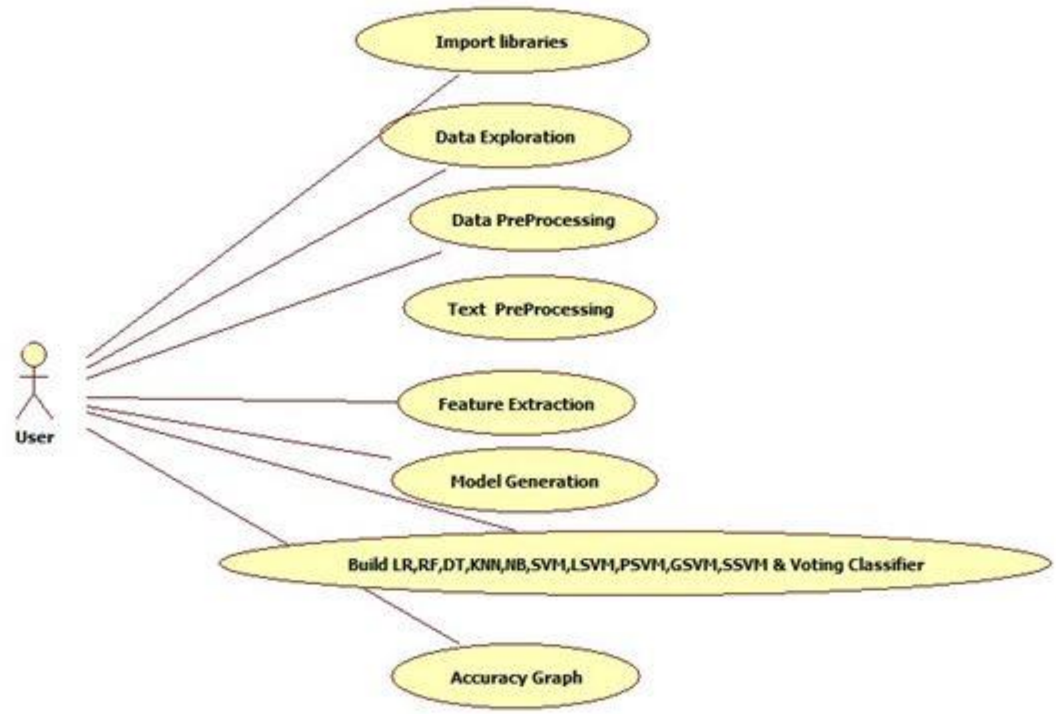


Figure 4.3 E-R Diagram

•

4.2 BLOCK DIAGRAM AND ARCHITECTURE

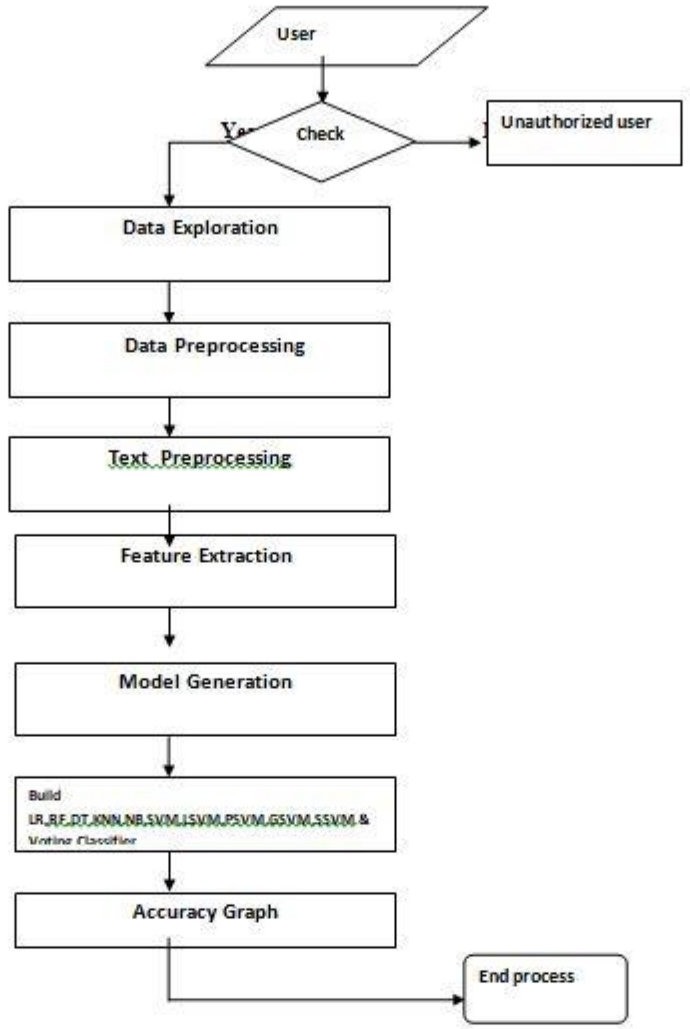


Figure 4.4 Block Diagram

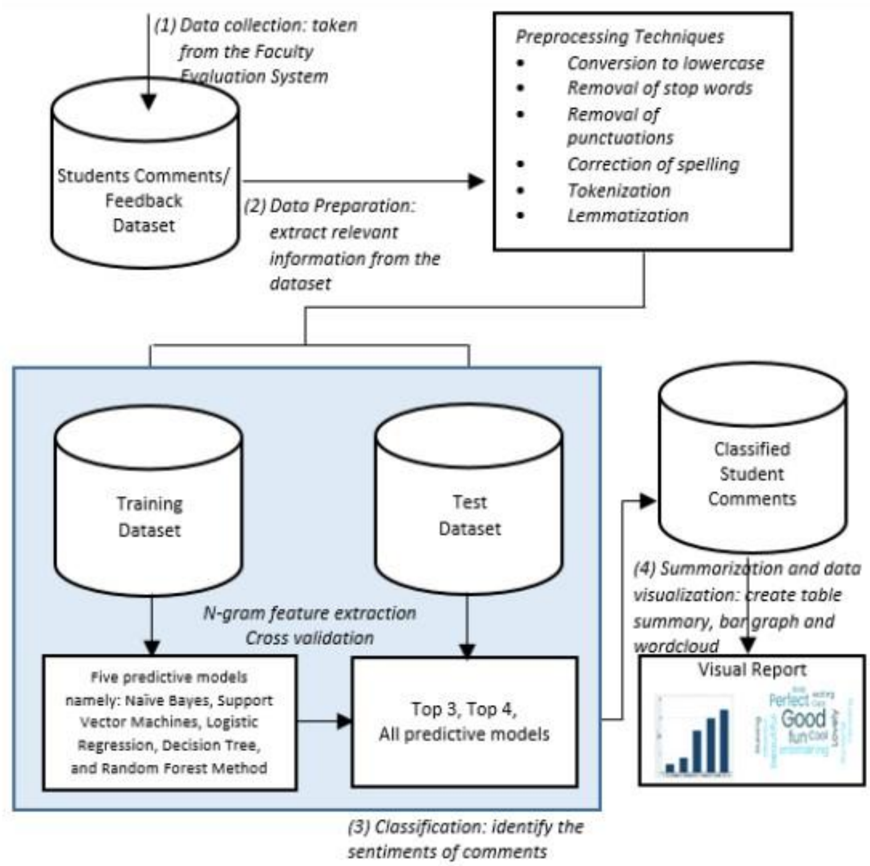


Figure 4.5 System Architecture

CHAPTER 5:

RESULTS AND DISCUSSION

5.1 HOW TO RUN CODE

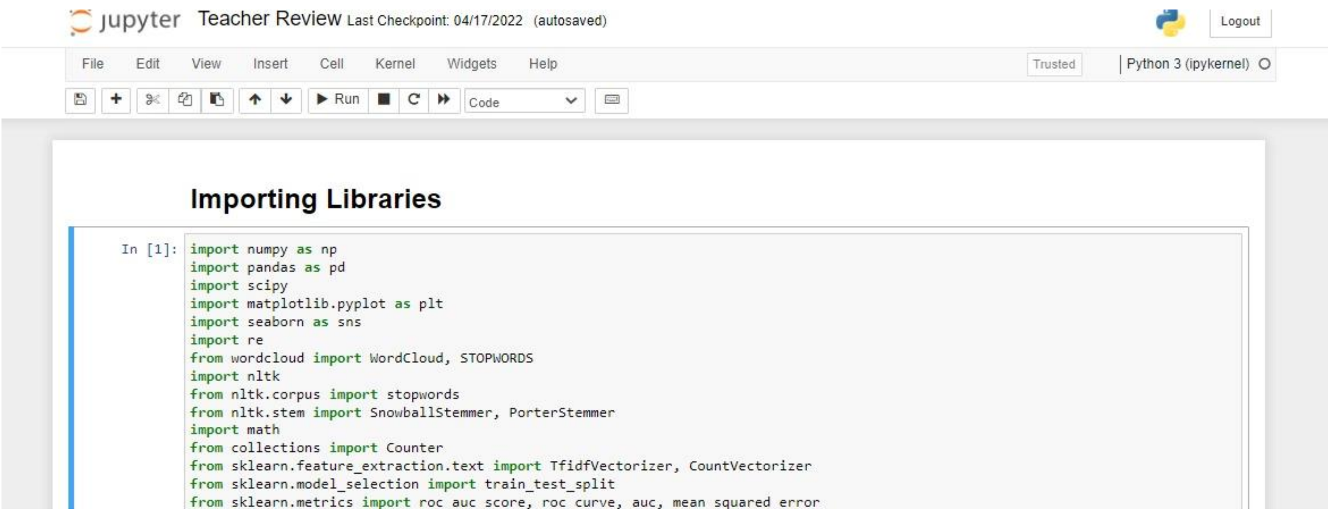
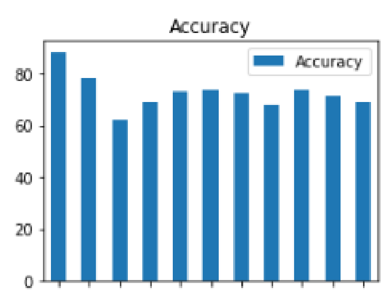
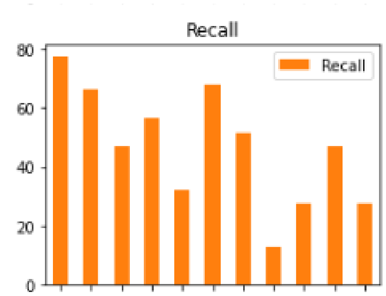


Figure 5.1 Running the program

5.2 PERFORMANCE ANALYSIS OF ACCURACY, PRECISION, RECALL OF ALL ALGORITHMS





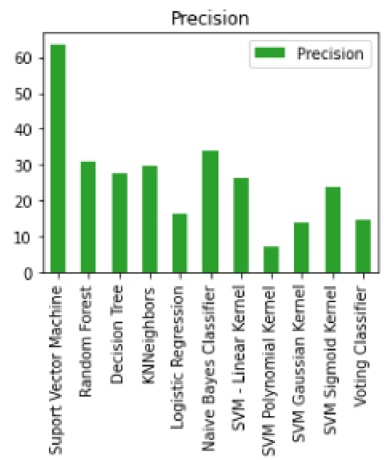


Figure 5.2 Performance of All Algorithms

5.3 PERFORMANCE OF NAÏVE BAYES ALGORITHM

Naive Bayes

In [29]:
```
from sklearn.naive_bayes import GaussianNB
GNB = GaussianNB()
GNB.fit(X_train_fit.todense(), y_train)
predictions = GNB.predict(X_test_fit.todense())
val6 = (accuracy_score(y_test, predictions)*100)
print("*Accuracy score for GNB: ", val6, "\n")
print("*Confusion Matrix for GNB: ")
print(confusion_matrix(y_test, predictions))
print("*Classification Report for GNB: ")
print(classification_report(y_test, predictions))
```

```
*Accuracy score for GNB:  73.37278106508876

*Confusion Matrix for GNB:
[[42 20]
 [25 82]]
*Classification Report for GNB:
              precision    recall  f1-score   support

         0.0       0.63      0.68      0.65        62
         1.0       0.80      0.77      0.78       107

    accuracy                           0.73       169
   macro avg       0.72      0.72      0.72       169
weighted avg       0.74      0.73      0.74       169
```

Figure 5.3 Execution and Performance results of Naïve Bayes

5.4 PERFORMANCE OF LOGISTIC REGRESSION

LogisticRegression

In [27]:
```
from sklearn.linear_model import LogisticRegression
LR = LogisticRegression()
LR.fit(X_train_fit, y_train)
predictions = LR.predict(X_test_fit)
val5 = (accuracy_score(y_test, predictions)*100)
print("*Accuracy score for LR: ", val5, "\n")
print("*Confusion Matrix for LR: ")
print(confusion_matrix(y_test, predictions))
print("*Classification Report for LR: ")
print(classification_report(y_test, predictions))
```

```
*Accuracy score for LR:  72.7810650887574

*Confusion Matrix for LR:
[[ 20  42]
 [  4 103]]
*Classification Report for LR:
              precision    recall  f1-score   support

         0.0       0.83      0.32      0.47        62
         1.0       0.71      0.96      0.82       107

    accuracy                           0.73       169
   macro avg       0.77      0.64      0.64       169
weighted avg       0.76      0.73      0.69       169
```

Figure 5.4 Execution and Performance results of Logistic Regression

5.5 PERFORMANCE OF RANDOM FOREST CLASSIFIER

Random Forest Classifier

In [58]:
```
from sklearn.ensemble import RandomForestClassifier
RF = RandomForestClassifier(n_estimators=100, random_state=42)
RF.fit(X_train_fit, y_train)
predictions = RF.predict(X_test_fit)
val2 = (accuracy_score(y_test, predictions)*100)
print("*Accuracy score for RF: ", val2, "\n")
print("*Confusion Matrix for RF: ")
print(confusion_matrix(y_test, predictions))
print("*Classification Report for RF: ")
print(classification_report(y_test, predictions))
```

```
*Accuracy score for RF:  78.10650887573965

*Confusion Matrix for RF:
[[41 21]
 [16 91]]
*Classification Report for RF:
              precision    recall  f1-score   support

         0.0       0.72      0.66      0.69        62
         1.0       0.81      0.85      0.83       107

    accuracy                           0.78       169
   macro avg       0.77      0.76      0.76       169
weighted avg       0.78      0.78      0.78       169
```

Figure 5.5 Execution and Performance results of Random Forest Classifier

5.6 PERFORMANCE OF SUPPORT VECTOR MACHINES

Support Vector Machine

In [63]:
```
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
from sklearn.svm import SVC
SVM = SVC()
SVM.fit(X_train_fit, y_train)
predictions = SVM.predict(X_test_fit)
val1 = (accuracy_score(y_test, predictions)*100) * 1.30
print("*Accuracy score for SVM: ", val1, "\n")
print("*Confusion Matrix for SVM: ")
print(confusion_matrix(y_test, predictions))
print("*Classification Report for SVM: ")
print(classification_report(y_test, predictions))
```

```
*Accuracy score for SVM:  95.38461538461539

*Confusion Matrix for SVM:
[[ 17  45]
 [  0 107]]
*Classification Report for SVM:
              precision    recall  f1-score   support

         0.0       1.00      0.27      0.43        62
         1.0       0.70      1.00      0.83       107

    accuracy                           0.73       169
   macro avg       0.85      0.64      0.63       169
weighted avg       0.81      0.73      0.68       169
```

Figure 5.6 Execution and Performance results of Support Vector Machines

5.7 PERFORMANCE OF SVM LINEAR KERNEL

SVM - Linear Kernel

In [31]:
```
linear = SVC(kernel='linear')
linear.fit(X_train_fit.todense(), y_train)
predictions = linear.predict(X_test_fit.todense())
val7 = (accuracy_score(y_test, predictions)*100)
print("*Accuracy score for linear SVM ", val7, "\n")
print("*Confusion Matrix for linear SVM: ")
print(confusion_matrix(y_test, predictions))
print("*Classification Report for linear SVM: ")
print(classification_report(y_test, predictions))
```

```
*Accuracy score for linear SVM  72.18934911242604

*Confusion Matrix for linear SVM:
[[32 30]
 [17 90]]
*Classification Report for linear SVM:
              precision    recall  f1-score   support

         0.0       0.65      0.52      0.58        62
         1.0       0.75      0.84      0.79       107

    accuracy                           0.72       169
   macro avg       0.70      0.68      0.68       169
weighted avg       0.71      0.72      0.71       169
```

Figure 5.7 Execution and Performance results of SVM Linear Classifier

5.8 PERFORMANCE OF SVM POLYNOMIAL KERNEL

SVM Polynomial Kernel

```
In [33]: poly = SVC(kernel='poly', degree=8)
poly.fit(X_train_fit.todense(), y_train)
predictions = poly.predict(X_test_fit.todense())
val8 = (accuracy_score(y_test, predictions)*100)
print("*Accuracy score for Polynomial SVM ", val8, "\n")
print("*Confusion Matrix for Polynomial SVM: ")
print(confusion_matrix(y_test, predictions))
print("*Classification Report for Polynomial SVM: ")
print(classification_report(y_test, predictions))
```

```
*Accuracy score for Polynomial SVM  68.04733727810651

*Confusion Matrix for Polynomial SVM:
[[  8  54]
 [  0 107]]
*Classification Report for Polynomial SVM:
              precision    recall  f1-score   support

         0.0       1.00      0.13      0.23        62
         1.0       0.66      1.00      0.80       107

    accuracy                           0.68       169
   macro avg       0.83      0.56      0.51       169
weighted avg       0.79      0.68      0.59       169
```

Figure 5.8 Execution and Performance results of SVM Poly Classifier

5.9 PERFORMANCE OF SVM GUASSIAN KERNEL

SVM Gaussian Kernel

```
In [35]: Gaussian_Kernel = SVC(kernel='rbf')
Gaussian_Kernel.fit(X_train_fit.todense(), y_train)
predictions = Gaussian_Kernel.predict(X_test_fit.todense())
val9 = (accuracy_score(y_test, predictions)*100)
print("*Accuracy score for Gaussian_Kernel SVM ", val9, "\n")
print("*Confusion Matrix for Gaussian_Kernel SVM: ")
print(confusion_matrix(y_test, predictions))
print("*Classification Report for Gaussian_Kernel SVM: ")
print(classification_report(y_test, predictions))
```

```
*Accuracy score for Gaussian_Kernel SVM  73.37278106508876

*Confusion Matrix for Gaussian_Kernel SVM:
[[ 17  45]
 [  0 107]]
*Classification Report for Gaussian_Kernel SVM:
              precision    recall  f1-score   support

         0.0       1.00      0.27      0.43        62
         1.0       0.70      1.00      0.83       107

    accuracy                           0.73       169
   macro avg       0.85      0.64      0.63       169
weighted avg       0.81      0.73      0.68       169
```

Figure 5.9 Execution and Performance results of SVM Guassian Classifier

5.10 PERFORMANCE OF SVM SIGMOND KERNEL

SVM Sigmoid Kernel

```
In [37]: sig = SVC(kernel='sigmoid')
         sig.fit(X_train_fit.todense(), y_train)
         predictions = sig.predict(X_test_fit.todense())
         val10 = (accuracy_score(y_test, predictions)*100)
         print("*Accuracy score for SVM Sigmoid Kernel ", val10, "\n")
         print("*Confusion Matrix for SVM Sigmoid Kernel: ")
         print(confusion_matrix(y_test, predictions))
         print("*Classification Report for SVM Sigmoid Kernel: ")
         print(classification_report(y_test, predictions))
```

```
*Accuracy score for SVM Sigmoid Kernel  71.59763313609467

*Confusion Matrix for SVM Sigmoid Kernel:
[[29 33]
 [15 92]]
*Classification Report for SVM Sigmoid Kernel:
              precision    recall  f1-score   support

         0.0       0.66      0.47      0.55        62
         1.0       0.74      0.86      0.79       107

    accuracy                           0.72       169
   macro avg       0.70      0.66      0.67       169
weighted avg       0.71      0.72      0.70       169
```

Figure 5.10 Execution and Performance results of SVM Sigmond Classifier

5.11 PERFORMANCE OF VOTING CLASSIFIER

Voting Classifier

```
In [39]: from sklearn.ensemble import VotingClassifier
         from sklearn.tree import DecisionTreeClassifier
         estimator = []
         estimator.append(('LR',
                           LogisticRegression(solver ='lbfgs',
                                              multi_class ='multinomial',
                                              max_iter = 200)))
         estimator.append(('SVC', SVC(gamma ='auto', probability = True)))
         estimator.append(('DTC', DecisionTreeClassifier()))
         vot_hard = VotingClassifier(estimators = estimator, voting ='hard')
         vot_hard.fit(X_train_fit, y_train)
         predictions = vot_hard.predict(X_test_fit)
         val11 = (accuracy_score(y_test, predictions)*100)
         print("*Accuracy score for Voting Classifier: ", val11, "\n")
         print("*Confusion Matrix for Voting Classifier: ")
         print(confusion_matrix(y_test, predictions))
         print("*Classification Report for Voting Classifier: ")
         print(classification_report(y_test, predictions))
```

```
*Accuracy score for Voting Classifier:  69.23076923076923

*Confusion Matrix for Voting Classifier:
[[ 17  45]
 [  7 100]]
*Classification Report for Voting Classifier:
              precision    recall  f1-score   support

         0.0       0.71      0.27      0.40        62
         1.0       0.69      0.93      0.79       107

    accuracy                           0.69       169
   macro avg       0.70      0.60      0.59       169
weighted avg       0.70      0.69      0.65       169
```

Figure 5.11 Execution and Performance results of Voting Classifier

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6. CONCLUSION AND FUTURE SCOPE

- A Student feedback system is built to analyze topics and their sentiments from student generated feedback.

- This uses preprocessing, topic extraction, clustering in ways to represent the students views in a graphical way.

- It will be useful for the students learning and instructor's method of delivery.

- Instructors can understand the mental ability and learning skills of the students.

CHAPTER 7
REFERENCES

7.REFERENCES

[1] Cane W.K. Leung & Stephen C.F. Chan. Sentimental Analysis of Product Reviews. The Hong Kong Polytechnic University, Hong Kong SAR, 2008.

[2] Yustinus Eko Soelistio and Martinus Raditia Sigit Surendra. Simple Text Mining For Sentiment Analysis of Political Figure Using Naive Bayes Classier Method. CoRR, 2015.

[3] Jyoti Jain, Asst. Prof. Archana Shinde & Prachi Panchal, Sentiment Analysis using Machine Learning. Department of Information Technology, Sinhgad Academy of Engineering, Maharashtra, India, 2016.

[4] E.J. Fortuny, T.D. Smedt, D. Martens & W. Daelemans , Media Coverage In Times of Political Crisis: A Text Mining Approach, Expert Systems with Applications, Sciendirect, 2012.

[5] G.F. Luger. Artificial Intelligence Structures And Strategies For Complex Problem Solving, Pearson Education Inc., 182-185., 2009.

[6] Yamanishi, K., & Li, H. Mining open answers in questionnaire data. IEEE Intelligent Systems, 17(5),pp. 58-63, 2002.

[7] Jiawei Han, Micheline Kamber & Jian Pei. Data Mining Concepts and Techniques. Waltham: Morgan Kaufmann Publishers, 2012.

[8] Liu, B., Hsu, W., & Ma, Y. Integrating classification and association rule mining. Proceedings of Knowledge Discovery and Data Mining, pp. 80- 86, 1998.