TWITTER SENTIMENT ANALYSIS

A PROJECT REPORT

Submitted by

THEOLA LOIS KEREN R

KEERTHANADEVI R

POOJANA A

SARANYA N

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

COIMBATORE INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

COIMBATORE – 641109

MAY 2023

COIMBATORE INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous/Approved by AICTE/Affiliated to Anna University)

COIMBATORE - 641109

BONAFIDE CERTIFICATE

Certified that this project report "TWITTER SENTIMENT ANALYSIS"

is the bonafide work of

THEOLA LOIS KEREN R	**71052002055**
KEERTHANADEVI R	**71052002023**
POOJANA A	**71052002037**
SARANYA N	**71052002049**

who carried out the project work under my supervision.

SIGNATURE

Dr. K.PUSHPALATHA, M.E., Ph.D.

HEAD OF THE DEPARTMENT
Department of Computer Science and Engineering,
Coimbatore Institute of Engineering and Technology
Coimbatore – 641109

SIGNATURE

Ms. S.Kanchana, M.E

ASSISTANT PROFESSOR
Department of Computer Science and Engineering,
Coimbatore Institute of Engineering and Technology
Coimbatore – 641109

Submitted for the University Project viva – voce held on **12.05.2023**

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Our heart is filled with gratitude to Almighty God for empowering us with the courage, wisdom and strength to carry out this project work successfully.

We, have great and immense pleasure in expressing the acknowledgment to the numerous contributors for the success of this project work.

We would like to record our sincere indebtedness and gratitude to our beloved Director**, Dr.K.A.Chinnaraju M.Sc.,M.B.A.,Ph.D.,** for his noteworthy efforts to enhance our professional dexterity and co-curricular excellence.

We are obliged to our respected Principal**, Dr.N.Nagarajan, M.E., Ph.D.,**

for providing us with necessary facilities to carry out our project work.

We express our sincere thanks to our Head of the Department, Project coordinator **Dr.K.Pushpalatha M.E., Ph.D.,** for her guidance, co-operation and providing encouragement in completing this project.

We express our sincere thanks to our guide, project coordinator and guide

Ms. S.Kanchana, M.E for her guidance and valuable suggestions.

Finally, we extend our thanks to the management, faculty members,parents and our student friends for their support and encouragement and to all others, who extended their helping hands to us in the completion of our mini- project done in the pre-final year.

ABSTRACT

In this era of growing social media users, Twitter has significantly large number of daily users who post their opinions in the form of tweets. Twitter has become a major source of information and communication for millions of people around the world. With the massive amount of data generated by users on these platforms, sentiment analysis has become an important task for understanding public opinion and attitudes towards various topics. This project aims to perform sentiment analysis on Twitter data using machine learning algorithms, with the goal of classifying a tweet as positive, negative or neutral thereby providing valuable insights into the sentiments of Twitter users towards a particular topic.

Generally, the tweets being unstructured in format, firstly the tweets are converted into the structured format. In this project, tweets are resolved using pre-processing phase. The study utilizes a dataset of tweets related to the topic of interest and the dataset is trained using algorithms in a way, such that, it becomes capable of testing the tweets and it releases the required sentiments out of the tweets. Algorithms such as Logistic Regression, SVM, Random Forest, Naïve Bayes, XGBoost, Decision Tree are applied to analyze the sentiment of the tweets. The project evaluates the performance of these models using various evaluation metrics such as accuracy, precision, recall, and F1-score. The findings of this study can be used to inform businesses, policymakers, and individuals about public sentiment and opinions on a particular topic, providing a valuable tool for decision-making and strategic planning. The project contributes to the growing field of sentiment analysis and highlights the potential of machine learning algorithms in analyzing social media data for various applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ML	-	Machine Learning
AI	-	Artificial Intelligence
SA	-	Sentiment Analysis
BPA	-	Business Process Automation
NB	-	Naïve Bayes
SVM	-	Support Vector Machine
XGBoost	-	Extreme Gradient Boosting
CSV	-	Comma Separated Value
POS	-	Point Of Sale
NLP	-	Natural Language Processing
NLTK	-	Natural Language Toolkit
SVR	-	Support Vector Regression
API	-	Application Programming Interface
TF-IDF	-	Term Frequency and Inverse Document Frequency
CART	-	Classification And Regression Tree Algorithm
ETL	-	Extract, Transform, Load
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

Sentiment Analysis is a rapidly growing field in the area of Natural Language Processing (NLP) that aims to extract insights from textual data by analyzing the sentiments, emotions, and attitudes expressed in the text. The ability to automatically classify text into different sentiment categories, such as positive, negative, or neutral, has many practical applications, including social media monitoring, customer feedback analysis, market research, and public opinion analysis.

The objective of this project is to develop a Sentiment Analysis system that can accurately classify text data from various sources into different sentiment categories. The project involves collecting and preprocessing text data, developing machine learning models to classify the data into sentiment categories, and evaluating the performance of the system.

This project is significant because it addresses a growing need for businesses and organizations to gain insights into customer opinions and feedback and improve their products, services, and marketing strategies, and enhance customer satisfaction. Furthermore, the project contributes to the broader field of NLP and Sentiment Analysis by exploring the effectiveness of different machine learning algorithms and techniques for sentiment classification.

In this report, we will provide a detailed overview of the project, including the research methodology, data collection and preprocessing methods, machine learning algorithms used for sentiment analysis, and the performance evaluation of the system. We will also discuss the potential applications and limitations of the system, and provide recommendations for future research.

What is Sentiment Analysis?

Sentiment Analysis is process of collecting and analyzing data based upon the personal feelings, reviews and thoughts. Sentimental analysis is often called as opinion mining as it mines the important feature from people opinions. Sentiment Analysis is done by using various machine learning techniques, statistical models and Natural Language Tool Kit (NLTK) for the extraction of feature from a large data.

Sentiment Analysis can be done at document, phrase and sentence level. In document level, summary of the entire document is taken first and then it is analyzed whether the sentiment is positive, negative or neutral. In phrase level, analysis of phrases in a sentence is taken in account to check the polarity. In sentence level, each sentence is classified in a particular class to provide the sentiment.

Sentimental Analysis has various applications. It is used to generate opinions for people of social media by analyzing their feelings or thoughts which they provide in form of text. Sentiment Analysis is domain centered, i.e, results of one domain cannot be applied to other domain. Sentimental Analysis is used in many real world scenarios, to get reviews about any product or movies, to get the financial report of any company, for predictions or marketing.

Figure 1.1 Sentiments of people

1.1 MOTIVATION FOR WORK

Businesses primarily run over customer's satisfaction and customer's reviews about their products. Shifts in sentiment on social media have been shown to correlate with shifts in stock markets. Identifying customer grievances and thereby resolving them leads to customer satisfaction as well as trustworthiness of an organization. Hence there is a necessity of an unbiased automated system to classify customer reviews regarding any problem.

In today's environment where we're justifiably suffering from data overload (although this does not mean better or deeper insights), companies might have mountains of customer feedback collected; but for mere humans, it's still impossible to analyze it manually without any sort of error or bias.

Sentiment analysis provides solution to what the most important issues are, from the perspective of customers, at least. Because sentiment analysis can be automated, decisions can be made based on a significant amount of data rather than plain intuition that isn't always right.

Twitter is a micro blogging platform where anyone can read or write short form of message which is called tweets. The amount of data accumulated on twitter is very huge. This data is unstructured and written in natural language. Twitter Sentimental Analysis is the process of accessing tweets for a particular topic and predicts the sentiment of these tweets as positive, negative or neutral with the help of different machine learning algorithm.

This work is motivated by several factors such as understanding customer sentiment for business intelligence, predicting public attitudes for political analysis, developing effective communication strategies during crises, and gaining insights into social trends and behaviors for social research. By analyzing tweets, researchers can gain a better understanding of public opinion, sentiment, and attitudes and use this information to improve decision-making in various domains.

1.2 PROBLEM STATEMENT

To create a functional classifier for accurate and automatic sentiment classification of an unknown tweet stream using machine learning approaches.

1.3 GOAL OF THE PROJECT

It is a challenging task to deal with a large dataset, but with the use of NLTK we can easily classify our data and give more accurate results based on different classifiers. Thus the goal is to perform sentiment analysis on different texts (tweets) mined from Twitter and then classify into sentiments, whether positive, negative or neutral by using supervised machine learning classifiers. These results will let us know about the reviews and opinions of people.

From the knowledge gained from an analysis such as this a company can identify issues with their products, spot trends before their competitors, create improved communications with their target audience, and gain valuable insight into how effective their marketing campaigns were. In turn, companies gain valuable feedback which allows them to further develop the next generation of their product.

1.4 SIGNIFICANCE OF THE PROJECT

Twitter sentiment analysis has become an important area of research and development due to its potential to provide valuable insights into public opinion and sentiment. By analyzing tweets related to a particular topic, event, or brand, sentiment analysis algorithms can classify the sentiment of each tweet as positive, negative, or neutral. This information can be used to identify patterns and trends in public opinion and sentiment, and help businesses, organizations, and governments make informed decisions.

Twitter sentiment analysis is also important from an academic and research

perspective. It provides an opportunity to study the social and psychological factors that influence public opinion and sentiment, and can help researchers gain insights into social trends and behaviors.

1.5 APPLICATIONS OF SENTIMENT ANALYSIS

a) **Brand monitoring:** Companies use sentiment analysis to monitor brand mentions on social media platforms and online reviews to understand how their customers feel about their products and services.

b) **Customer feedback analysis:** Sentiment analysis is used to analyze customer feedback, such as surveys, emails, and chat transcripts, to gain insights into customer satisfaction and improve customer experience.

c) **Market research:** Sentiment analysis is used in market research to analyze consumer opinions and preferences, track market trends, and identify emerging topics and issues.

d) **Political analysis:** Sentiment analysis is used in political analysis to gauge public opinion about political candidates, parties, policies, and events.

e) **Social media analysis:** Sentiment analysis is widely used in social media analytics to monitor and analyze user-generated content, such as tweets, posts, and comments, to gain insights into user behaviour and sentiment.

f) **Customer service:** Sentiment analysis is used in customer service to classify incoming messages and prioritize responses based on the sentiment and urgency of the messages.

g) **Fraud detection:** Sentiment analysis is used in fraud detection to analyze text data, such as insurance claims and financial transactions, to detect fraudulent activities and suspicious patterns.

h) **Government:** It is helpful in the administration fields of Government like decision making policies, recruitments, taxation and evaluating social strategies. Applying Sentiment Analysis method in the multi- lingual country like the India where content of the generating mixture of the different languages is a very common practice.

What is Python?

Python is a high level, dynamic programming language. Python 3.11.3 version is used for the development of the project as it is a mature, versatile and robust programming language. It is an interpreted language which makes the testing and debugging extremely quickly as there is no compilation step. There are extensive open source libraries available for this version of python and a large community of users.

Python is well known for its functionality of processing natural language data, i.e, spoken English using NLTK. Other high level programming languages such as 'R' and 'Matlab' were considered because they have many benefits such as ease of use but they do not offer the same flexibility and freedom that Python can deliver.

The major reasons for using Python in Machine Learning are:

- Libraries and Frameworks
- Simple and Consistent
- Platform Independence
- Great community base

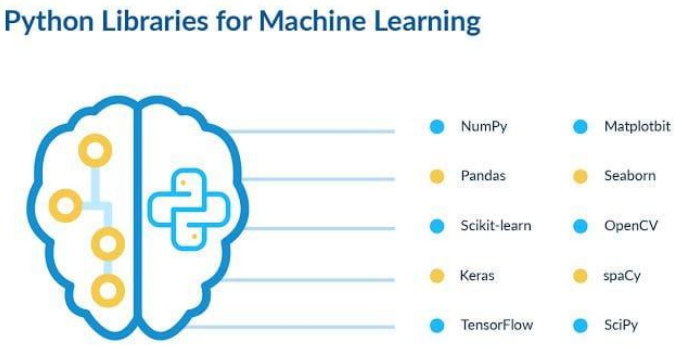

Figure 1.2 Python Libraries for Machine Learning

What is NLTK?

Natural Language Toolkit (NLTK) is library in Python, which provides a base for building programs and classification of data. NLTK is a collection of resources for Python that can be used for text processing, classification, tagging and tokenization. This toolbox plays a key role in transforming the text data in the tweets into a format that can be used to extract sentiment from them.

NLTK provides various functions which are used in pre-processing of data so that data available from twitter becomes fit for mining and extracting features. NLTK support various machine learning algorithms which are used for training classifier and to calculate the accuracy of different classifier.

In our thesis we use Python as our base programming language which is used for writing code snippets. NLTK is a library of Python which plays a very important role in converting natural language text to a sentiment either positive, negative or neutral. NLTK also provides different sets of data which are used for training classifiers. These datasets are structured and stored in library of NLTK, which can be accessed easily with the help of Python.

Figure 1.3 NLTK in Sentiment Analysis

What is NLP?

Natural Language Processing (NLP) is the branch of computer science that deals with the interaction between human language and computers. It involves developing algorithms and models that can understand, analyze, and generate human language. In

other words, NLP enables computers to process and interpret human language in a way that is meaningful and useful to humans.

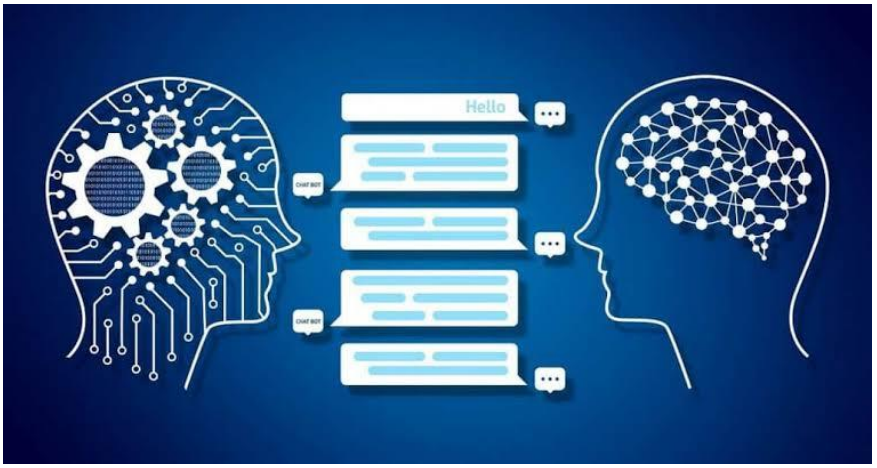

Figure 1.4 NLP in Sentiment Analysis

What is Data Mining?

Data mining is the process of discovering patterns, relationships, and insights from large datasets using advanced computational techniques. It involves using statistical and machine learning algorithms to identify patterns and relationships within the data that can be used to make predictions or gain insights into complex systems.

The process of data mining typically involves several stages, including data cleaning, data integration, data selection, data transformation, pattern discovery, and knowledge representation. These stages can be performed using a variety of tools and techniques, such as data visualization, clustering, association rule mining, and decision tree analysis.

1.6 Introduction to Machine Learning

1.6.1 Overview

Machine learning is like having a superpower that lets you train computers to learn and improve from experience, without being explicitly programmed. With machine learning, we can unlock insights and make predictions from massive amounts of data, leading to smarter decisions and better outcomes.

Machine learning can be broadly categorized into three categories as shown below:

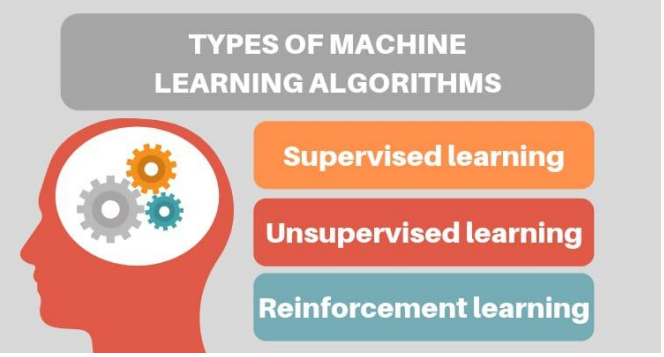

Figure 1.5 Types of Machine Learning

a) **Supervised Learning:** Here, the algorithm is trained on labeled data, where the desired output is known. The algorithm learns to map the input to the output by minimizing the error between predicted and actual output. This is used in tasks such as classification, regression, and prediction.

b) **Unsupervised Learning:** Here, the algorithm is trained on unlabeled data, where the desired output is unknown. The algorithm learns to find patterns and relationships within the data by clustering or dimensionality reduction techniques. This is used in tasks such as anomaly detection, data compression, and data visualization.

c) **Reinforcement Learning:** In reinforcement learning, the algorithm learns by interacting with the environment and receiving feedback in the form of rewards or penalties. The goal is to learn the optimal policy that maximizes the cumulative reward. This is used in tasks such as game playing, robotics, and autonomous vehicles.

1.6.2 Steps involved in Machine Learning

Machine Learning cycle involves 7 major iterative steps, which are given below:

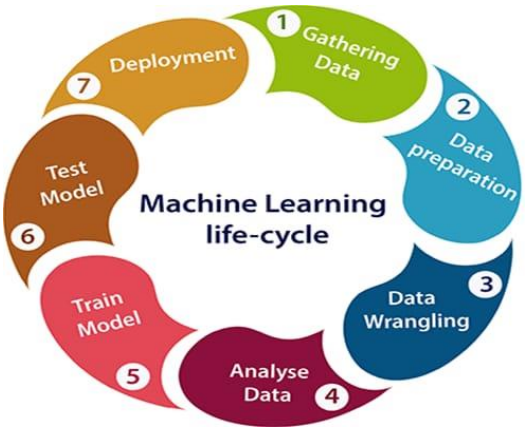

Figure 1.6 Steps of Machine Learning process

1. **Data Collection**: The first step in the machine learning process is to collect and gather the data that will be used to train the machine learning model. This may involve collecting data from different sources, such as databases, APIs, or scraping web pages.

2. **Data Preparation**: Once the data has been collected, it needs to be pre-processed and prepared for training the machine learning model. This may involve cleaning the data, removing duplicates, handling missing values, and converting the data into a suitable format.

3. **Data Splitting**: The next step is to split the data into two or more sets, typically a training set and a testing set. The training set is used to train the machine learning model, while the testing set is used to evaluate its performance.

4. **Model Selection**: The next step is to select an appropriate machine learning model for the given task. This may involve choosing between different types of models, such as decision trees, support vector machines, or neural networks.

5. **Model Training**: Once a model has been selected, it needs to be trained on the training data set. This involves feeding the data into the model and adjusting its parameters to optimize its performance.

6. **Model Evaluation**: After the model has been trained, it needs to be evaluated on the testing data set. This involves measuring its accuracy, precision, recall, and other performance metrics.

7. **Model Deployment**: Once the model has been trained and evaluated, it can be deployed and used to make predictions on new, unseen data. This may involve integrating the model into a larger system or application, such as a web application or a mobile app.

1.7 Introduction to Supervised Machine Learning

Supervised machine learning can be broadly categorized into the following categories:

- Classification
- Regression
- Time Series Forecasting
- Object Detection
- Natural Language Processing (NLP)

These categories are not mutually exclusive, and many algorithms can be applied to multiple types of tasks.

1.7.1 Supervised Machine Learning Classifiers

Supervised learning is commonly chosen for sentiment analysis because it requires labeled data, where the desired output (positive, negative, or neutral) of a given text or document is known. Supervised Learning algorithms allows for accurate and reliable predictions of sentiment in a variety of contexts, from product reviews to social media posts.

Different machine learning classifiers which we used to build our classifier are:

- Naïve-Bayes Classifier
- Logistic Regression Classifier
- SVM (Support Vector Machine)
- Random Forest Classifier
- XGBoost

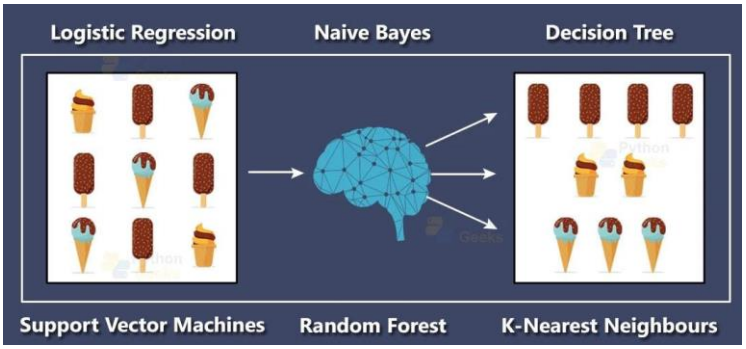

Figure 1.7 Classification Algorithms

1.7.1.1 Naïve-Bayes (NB) Classifier

Naïve-Bayes classifiers are probabilistic classifiers which come under machine learning techniques. These classifiers are based on applying Bayes' theorem with strong (naïve) assumption of independence between each pair of features. Let us assume, there is a dependent vector from x_1 to x_n, and a class variable 'y'. Therefore, according to Bayes' :

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

Now according to assumption of independence

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y),$$

For every 'i', this function becomes

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

In this $P(x_1, \ldots, x_n)$ on given input is constant, hence we can apply classification rule as:

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

And for estimating we can use MAP (Maximum A Posterior) estimation $P(y)$ and $P(x_i \mid y)$; the $P(y)$ of class 'y' in training sample is relative frequency.

1.7.1.2 Logistic Regression Classifier

Despite its name Logistic regression, is not a regression model but a linear model for classification. This model is also known by other names as Maximum-Entropy (MaxEnt) classification or log-linear classifier. A logistic function is used in this model, where probability describe the outcome of single trial.

The logistic regression can be implemented from Scikit-learn library of Python in which there is a class named LogisticRegression. This implementation fits a OvR (one-vs-rest) multiclass regression with an optional L1 or L2 regularization.

L2 penalized logistic regression helps in minimizing the following cost function:

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

Similarly, L1 regularized logistic regression can solve following problem of optimization:

$$\min_{w,c} \|w\|_1 + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

1.7.1.3 SVM (Support Vector Machine)

SVM's are used for both classification and regression tasks. The goal of SVM is to find the optimal hyperplane that separates the data into different classes, in such a way that the margin between the hyperplane and the closest data points from each class is maximized.

SVM's can handle both linearly separable and non-linearly separable data by using kernel functions to map the data into higher-dimensional spaces where a linear decision boundary can be identified. It involves Data pre-processing, Training and Classification.

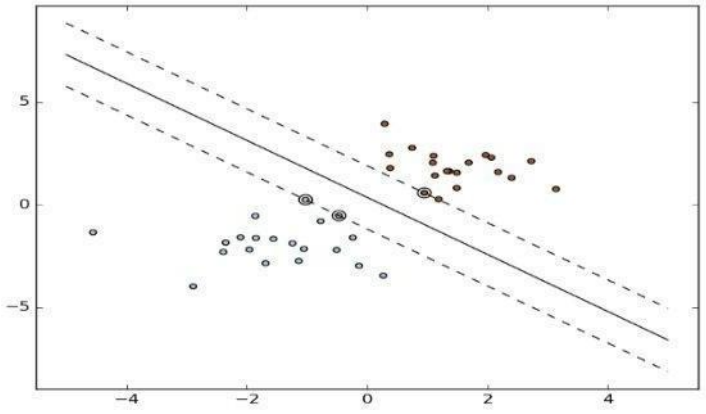

Figure 1.8 Hyperplane used for Classification in SVM

1.7.1.4 Random Forest Classifier

Random Forest Classifier is an ensemble learning method that combines multiple decision trees to make a prediction. A large number of decision trees are trained on different subsets of the training data, and each decision tree is allowed to make a

prediction. The final prediction is made by aggregating the predictions of all the trees, typically through a majority vote or weighted average.

Each decision tree in a Random Forest Classifier is constructed using a random subset of the features, which helps to reduce overfitting and improve the generalization performance of the model. The random selection of features also helps to create diversity among the decision trees, which improves the accuracy of the model.

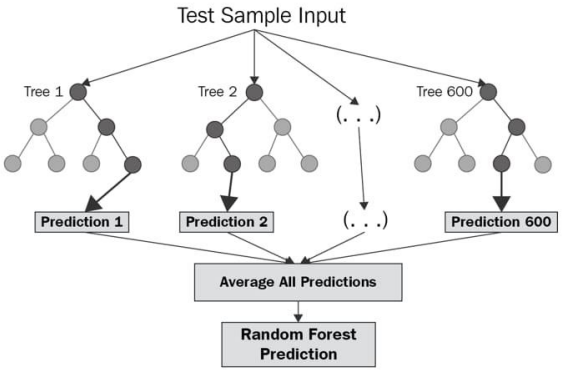

Figure 1.9 Working of a Random Forest Classifier

1.7.1.5 XGBoost Classifier

XGBoost, short for Extreme Gradient Boosting, is a powerful machine learning algorithm to perform sentiment analysis on Twitter data. The dataset used in this study consists of tweets related to a specific topic, which were collected using Twitter API. We first preprocess the data by removing stop words, punctuation, and special characters, and then perform feature engineering to convert the tweets into a numerical format. We use the bag-of-words model to represent the tweets as a vector of word frequencies. We then split the data into training and testing sets, with 80% of the data used for training and 20% for testing. Next, we train the XGBoost model on the training data and tune its hyperparameters using cross-validation and then evaluate the performance of the model using various evaluation metrics

1.7.1.6 Decision Tree Classifier

This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node. For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree.

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

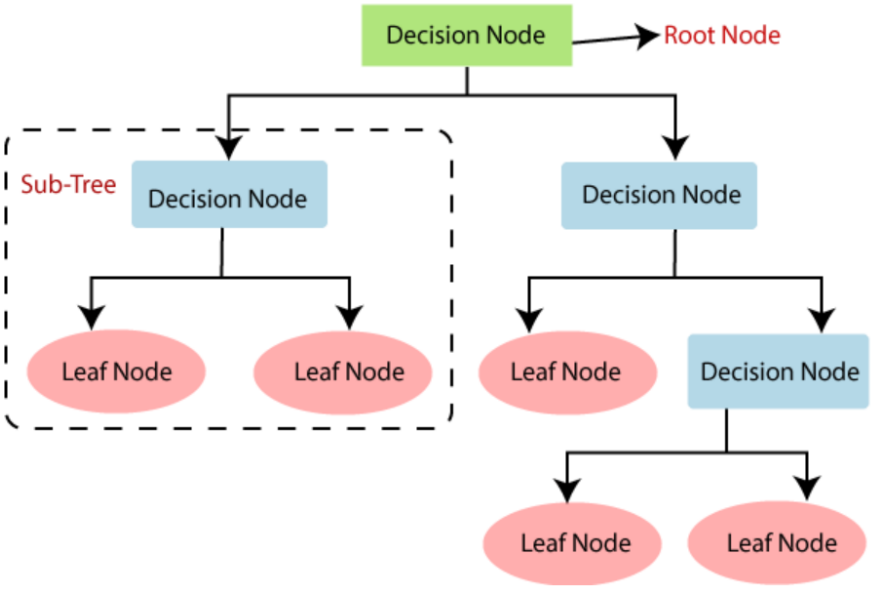

Figure 2 Working of a Decision Tree Classifier

1.8 Future Scope

The future of sentiment analysis is going to continue to dig deeper, far past the surface of the number of likes, comments and shares, and aim to reach, and truly understand, the significance of social media interactions and what they tell us about the consumers behind the screens. This forecast also predicts broader potential applications for sentiment analysis – brands will continue to leverage this tool, but so will individuals in the public eye, governments, nonprofits, education centers and many other organizations. As technology continues to advance, we can expect sentiment analysis to become even more sophisticated and useful in a wide range of industries and fields.

1.9 Limitations of Sentiment Analysis

1. Subjectivity: Trouble in determining the sentiment behind some texts, such as sarcasm or irony.
2. Context: The same text could be positive or negative depending on the context in which it is used
3. Tone: Sentiment analysis may not be able to determine the tone of a text accurately as it can be challenging to detect because it depends on the writer's intention and can be subtle.
4. Data Quality: If the data used for sentiment analysis is incomplete, incorrect, or biased, the analysis and accuracy will be inaccurate.
5. Multilingualism: Different languages can have different structures, expressions, and idioms that may affect the accuracy of the analysis.
6. Cultural Differences: People from different cultures may have different ways of expressing their opinions or emotions, which can be challenging to interpret accurately.
7. Machine Learning Limitations: Machine learning algorithms may not be able to handle new, unusual, or unexpected data that may affect the accuracy of the results.

CHAPTER 2

LITERATURE REVIEW

Many research have been done on the subject of sentiment analysis in past. Latest research in this area is to perform sentiment analysis on data generated by user from many social networking websites like Facebook, Twitter, Amazon, etc. Mostly research on sentiment analysis depend on machine learning algorithms, whose main focus is to find whether given text is in favor or against and to identify polarity of text. In this chapter we will provide insight of some of the research work which helps us to understand the topic deep.

2.1 Existing methodologies

1. **Rule-based approaches**:

Rule-based approaches rely on handcrafted rules to extract sentiment from text data. These rules are usually based on linguistic knowledge and pre-defined dictionaries of sentiment words. Eg: TextBlob library uses a lexicon-based approach to assign a sentiment score to each tweet based on the presence of positive and negative words in the text.

2. **Machine learning-based approaches**:

Machine learning-based approaches rely on algorithms to learn from data and make predictions about the sentiment of new tweets. These approaches require a labeled dataset of tweets, where each tweet is labeled with its sentiment polarity (positive, negative, or neutral).

3. **Hybrid approaches**:

Hybrid approaches combine rule-based and machine learning-based approaches to improve the accuracy of sentiment analysis. These approaches typically use a

combination of lexicon-based methods and machine learning algorithms to analyze sentiment. Eg: VADER (Valence Aware Dictionary and sEntiment Reasoner) model combines a rule-based approach with machine learning to achieve high accuracy in sentiment analysis.

Observation from existing methodologies:

Overall, each methodology has its own advantages and disadvantages. Rule-based approaches are easy to implement but may not be very accurate. Machine learning-based approaches require a labeled dataset and may not be effective if the training data is not representative of the target population. Hybrid approaches are effective at combining the strengths of both rule-based and machine learning-based approaches but may require more resources to implement.

2.2 Related Works

1. P. Pang, L. Lee, S. Vaithyanathan *et al*

They were the first to work on sentiment analysis. Their main aim was to classify text by overall sentiment, not just by topic e.g., classifying movie review either positive or negative. They apply machine learning algorithm on movie review database which results that these algorithms out-perform human produced algorithms. The machine learning algorithms they use are Naïve-Bayes, maximum entropy, and support vector machines. They also conclude by examining various factors that classification of sentiment is very challenging. They show supervised machine learning algorithms are the base for sentiment analysis.

2. 2018, Fang et al

They have suggested multi-strategy sentiment analysis models using the semantic fuzziness for resolving the issues. The outcomes have demonstrated that the proposed model has attained high efficiency.

3. 2019, Saad and Yang

They have aimed for giving a complete tweet sentiment analysis on the basis of ordinal regression with machine learning algorithms. The suggested model included pre-processing tweets as first step and with the feature extraction model, an effective feature was generated. The methods such as SVR, RF, Multinomial logistic regression (SoftMax), and DTs were employed for classifying the sentiment analysis. Moreover, twitter dataset was used for experimenting the suggested model. The test results have shown that the suggested model has attained the best accuracy, and also DTs were performed well when compared over other methods.

4. 2019, Afzaal et al

They have recommended a novel approach of aspect-based sentiment classification, which recognized the features in a precise manner and attained the best classification accuracy. Moreover, the scheme was developed as a mobile application, which assisted the tourists in identifying the best hotel in the town, and the proposed model was analyzed using the real-world data sets. The results have shown that the presented model was effective in both recognition as well as classification.

5. 2020, Kumar et al

They have presented a hybrid deep learning approach named ConVNet-SVMBoVW that dealt with the real-time data for predicting the fine-grained sentiment. In order to measure the hybrid polarity, an aggregation model was developed. Moreover, SVM was used for training the BoVW to forecast the sentiment of visual content. Finally, it was concluded that the suggested ConvNet-SVMBoVW was outperformed by the conventional models.

6. 2018, Abdi et al

They have proffered a machine learning technique for summarizing the opinions of the users mentioned in reviews. The suggested method merged multiple kinds of features into a unique feature set for modelling accurate classification model. Therefore, a performance investigation was done for four best feature selection

models for attaining the best performance and seven classifiers for choosing the relevant feature set and recognized an effective machine learning algorithm. The suggested method was implemented in various datasets. The outcomes have demonstrated that the combination of IG as the feature selection approach and SVM-based classification approach enhanced the performance.

7. 2019, Ray and Chakrabarti

They have introduced a deep learning algorithm for extracting the features from text and the user's sentiment analysis with respect to the feature. In opinionated sentences, a seven layer Deep CNN was employed for tagging the features. In order to enhance the performance of sentiment scoring and feature extraction models, the authors merged the deep learning methods using a set of rule-based models. Finally, it was seen that the suggested method achieved the best accuracy

8. 2019, Vashishtha and Susan

They have calculated the sentiment related to social media posts by a new set of fuzzy rules consisting of many datasets and lexicons. The developed model combined Word Sense Disambiguation and NLP models with a new unsupervised fuzzy rule-based model for categorizing the comments into negative, neutral, and positive sentiment class. The experiments were performed on 3 sentiment lexicons, four existing models, and nine freely available twitter datasets. The outcomes have shown that the introduced method was attaining the best results.

9. 2016, Igor Mozetič, Miha Grčar, and Jasmina Smailovič

Firstly, the researchers stated that there is no statistically major difference between the performance of the top classification models. Next, the general accuracy of the classification models does not correlate to performance when applied to the ordered three-class sentiment classification problem. Lastly, they stated that it is more efficient to focus on the accuracy of the training data, rather than the type of classification model used.

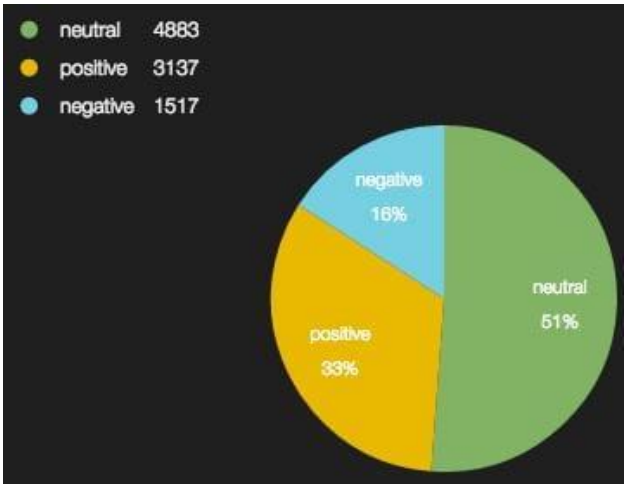

Figure 2.1 An example of a pie chart depicting the score of sentiments,

Methods

	Method	Data Set	Acc.	Author
Machine Learning	SVM	Movie reviews	86.40%	Pang, Lee[23]
	CoTraining SVM	Twitter	82.52%	Liu[14]
	Deep learning	Stanford Sentiment Treebank	80.70%	Richard[18]
Lexical based	Corpus	Product reviews	74.00%	Turkey
	Dictionary	Amazon's Mechanical Turk	---	Taboada[20]
Cross-lingual	Ensemble	Amazon	81.00%	Wan,X[16]
	Co-Train	Amazon, ITI68	81.30%	Wan,X.[16]
	EWGA	IMDb movie review	>90%	Abbasi,A.
	CLMM	MPQA,NTCIR,ISI	83.02%	Mengi
Cross-domain	Active Learning	Book, DVD, Electronics, Kitchen	80% (avg)	Li, S
	Thesaurus			Bollegala[22]
	SFA			Pan S J[15]

Figure 2.2 A comparative analysis of Sentiment Analysis Techniques

Commonly used Techniques:

- Lexicon-Based Techniques
- Machine Learning Techniques
- Rule-Based Techniques
- Hybrid Techniques
- Deep Learning Techniques labels.
- Emotion Detection Techniques

In conclusion, the choice of technique depends on the specific needs and requirements of the application, and it is essential to evaluate the accuracy and effectiveness of the technique before implementing it in real-world scenarios. The predictive model is a powerful tool for understanding and analyzing customer opinions and sentiment towards products and services, as well as tracking public opinion on social and political issues. However, there are still many challenges and limitations to sentiment analysis, including the need for more accurate and reliable sentiment lexicons, as well as the need for more robust and interpretable machine learning models.

2.3 Limitations of the system

- Subjectivity
- Context
- Tone
- Data quality
- Multilingualism
- Cultural differences
- Machine Learning limitations

CHAPTER 3

METHODOLOGY

3.1 Proposed System

"Twitter Sentiment Analysis using ML Model"

The proposed system aims to develop a sentiment analysis model for Twitter data using machine learning techniques as it offers various advantages including improved accuracy, adaptability, scalability, computational insights, flexibility, and automation, Machine learning algorithms and Natural Language Processing(NLP) can be effectively used to predict results in a variety of contexts.

The system will be designed to analyze a large volume of Twitter data and classify tweets into positive, negative, or neutral sentiment categories. The system will be trained using a Supervised learning approach and will use a variety of machine learning algorithms and techniques to identify patterns and relationships in the data.

The system will consist of several key components, including data collection, data preprocessing, feature extraction, and model training and testing. Data collection will involve scraping Twitter data using the Twitter API or other third-party tools. Data preprocessing will involve cleaning and normalizing the data to remove noise and irrelevant information. Feature extraction will involve extracting relevant features from the data, such as word frequencies, n-grams, and sentiment scores.

Machine learning algorithms like SVM, Logistic Regression, Naïve Bayes, Random Forest, XGBoost are used for implementation and comparative analysis. The performance of these algorithms will be evaluated using various metrics, such as accuracy, precision, recall, and F1 score. The system will also be flexible and adaptable, allowing users to customize the algorithms and features used for sentiment analysis.

Overall, the proposed system will be a valuable tool for businesses, marketers, and researchers who want to analyze and understand public sentiment towards products, brands, and social and political issues on Twitter.

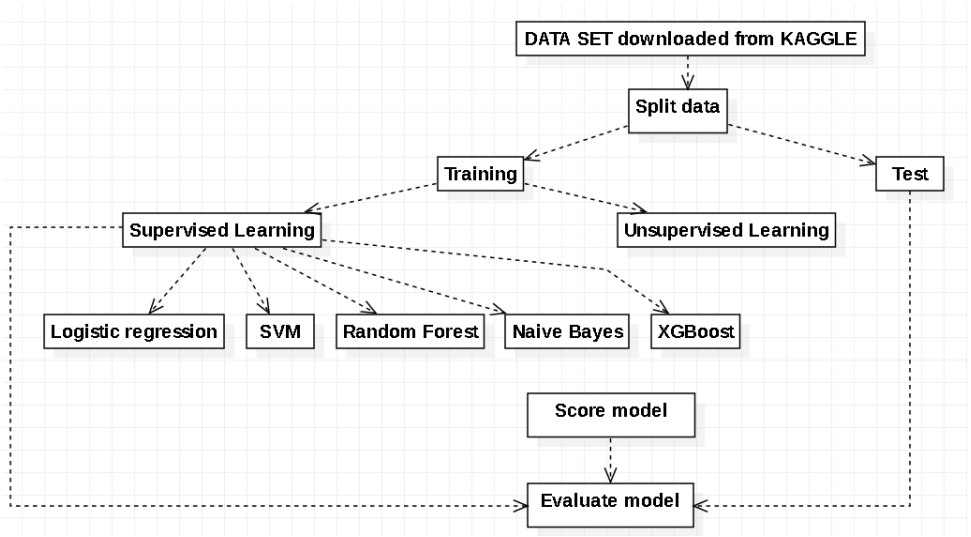

Figure 3.1 Use case diagram of the working model of the proposed system

3.2 System Architecture

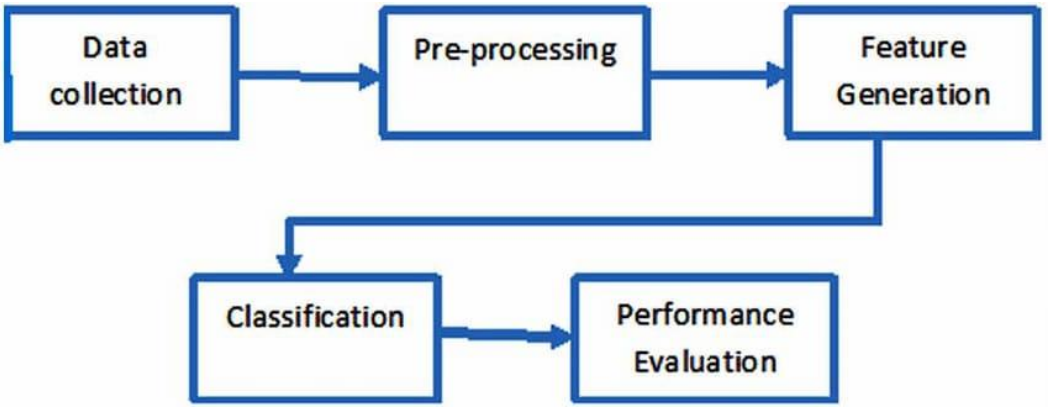

Figure 3.2 Architecture of the system

The Architecture includes the following steps:

1. **Data Collection**: Twitter data will be collected using the Twitter API or other third-party tools. The data will be stored in a database or file system for further processing.

2. **Data Preprocessing**: The collected data will be preprocessed to remove noise and irrelevant information. This step will involve text cleaning, such as removing URLs, mentions, and hashtags, as well as tokenization, stemming, and stop word removal.

3. **Feature Generation**: Relevant features will be extracted from the preprocessed data, such as word frequencies, n-grams, and sentiment scores. These features will be used to train and test the machine learning model.

4. **Model Training and Testing**: The machine learning model will be trained using a supervised learning approach, with labeled data for sentiment classification. The model will use various machine learning algorithms, such as support vector machines, decision trees, and Naive Bayes, to classify tweets into positive, negative, or neutral sentiment categories. The performance of the model will be evaluated using various metrics, such as accuracy, precision, recall, and F1 score.

5. **Classification**: The features extracted from the text, such as words, n-grams, or other linguistic features, to predict the sentiment label as positive, negative and neutral categories

6. **Performance evaluation**: The accuracy, precision, recall, F1-score, and the confusion matrix, which show the distribution of true positive, true negative, false positive, and false negative labels are measured and displayed.

3.3 Features of the proposed system

- Speed and scale
- Accuracy, Customization
- Aspect based analysis
- Data collection from various sources- social media, multimedia
- Flexible deployment and Visualization

CHAPTER 4

SYSTEM DESIGN AND ARCHITECTURE

The system design and architecture of a Twitter sentiment analysis project typically involves several key components and requires a solid understanding of NLP, machine learning, and software engineering principles.

4.1 UML diagrams:

Unified Modeling Language (UML) diagrams are widely used in software engineering and project management to represent and visualize the system design and architecture, for communication, documentation and analysis. UML diagrams provide a standardized way to communicate and document the different components, interactions, and behaviors of the system, and facilitate the collaboration and understanding among team members and stakeholders.

There are several types of UML diagrams that can be used for system design:

1. Use Case Diagrams: Use case diagrams depict the interactions between the system and its users or external entities, and show the different use cases, actors, and scenarios that the system supports.

2. Class Diagrams: Class diagrams describe the structure and relationships of the classes and objects in the system, and show the attributes, methods, and associations between them.

3. Sequence Diagrams: Sequence diagrams illustrate the interactions between the different components and objects in the system over time, and show the order of messages and actions exchanged between them.

4. Activity Diagrams: Activity diagrams represent the flow of activities and processes

in the system, and show the steps, decisions, and conditions that are involved in executing a particular task or use case.

5. Component Diagrams: Component diagrams show the physical or logical components of the system, and depict the interfaces, dependencies, and relationships between them.

6. Deployment Diagrams: Deployment diagrams illustrate the physical architecture of the system, and show how the different components and nodes are distributed across different machines and environments.

UML specification defines two major kinds of UML diagram:

- Structure diagrams
- Behavior diagrams

Structure diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts.

Behavior diagrams show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time.

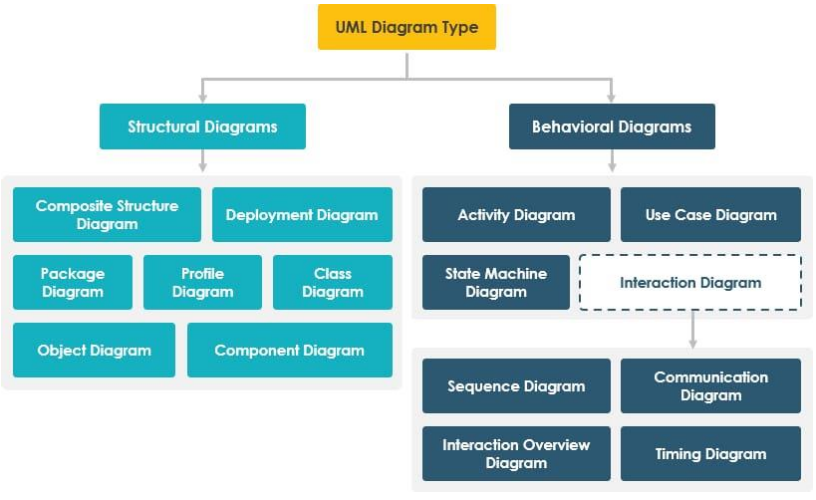

Figure 4.1 UML Diagrams

4.1.1 Use Case diagram

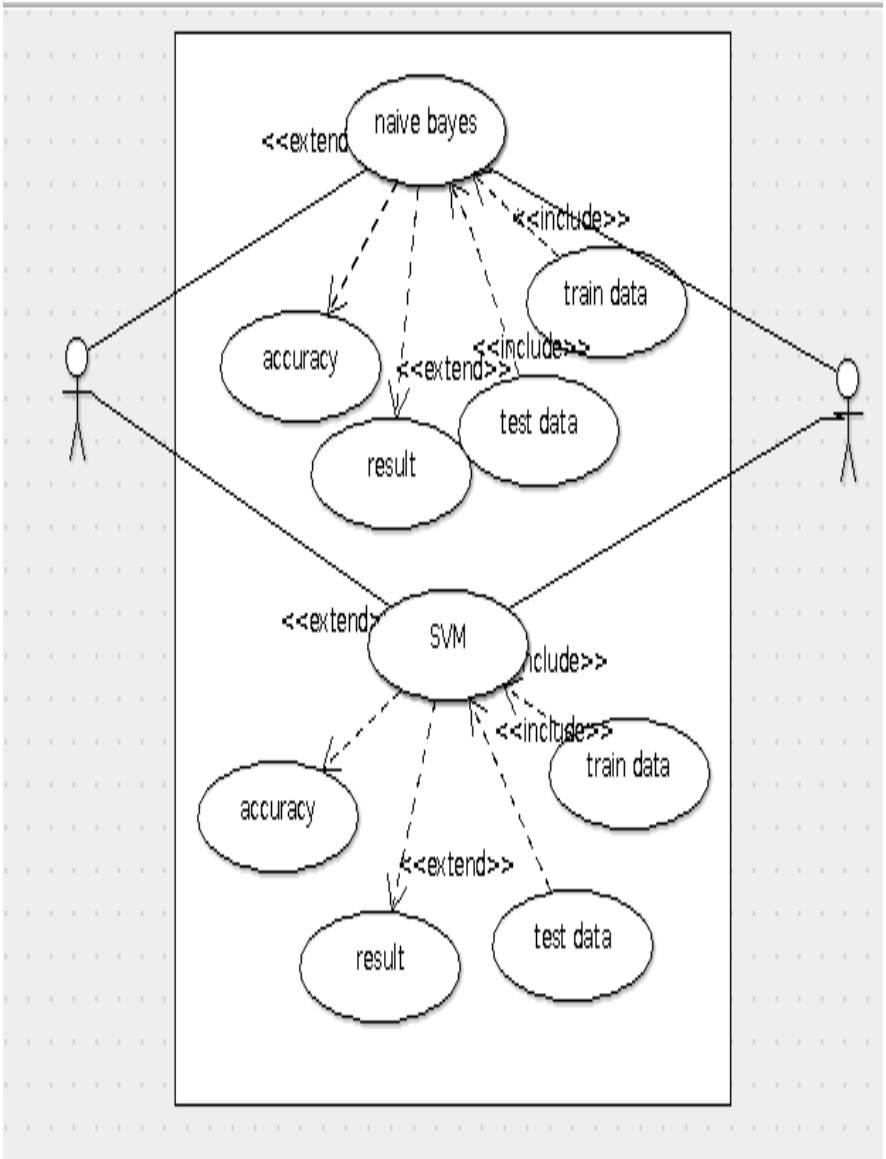

Figure 4.1.1 Use Case Diagram

4.1.2 Class diagram

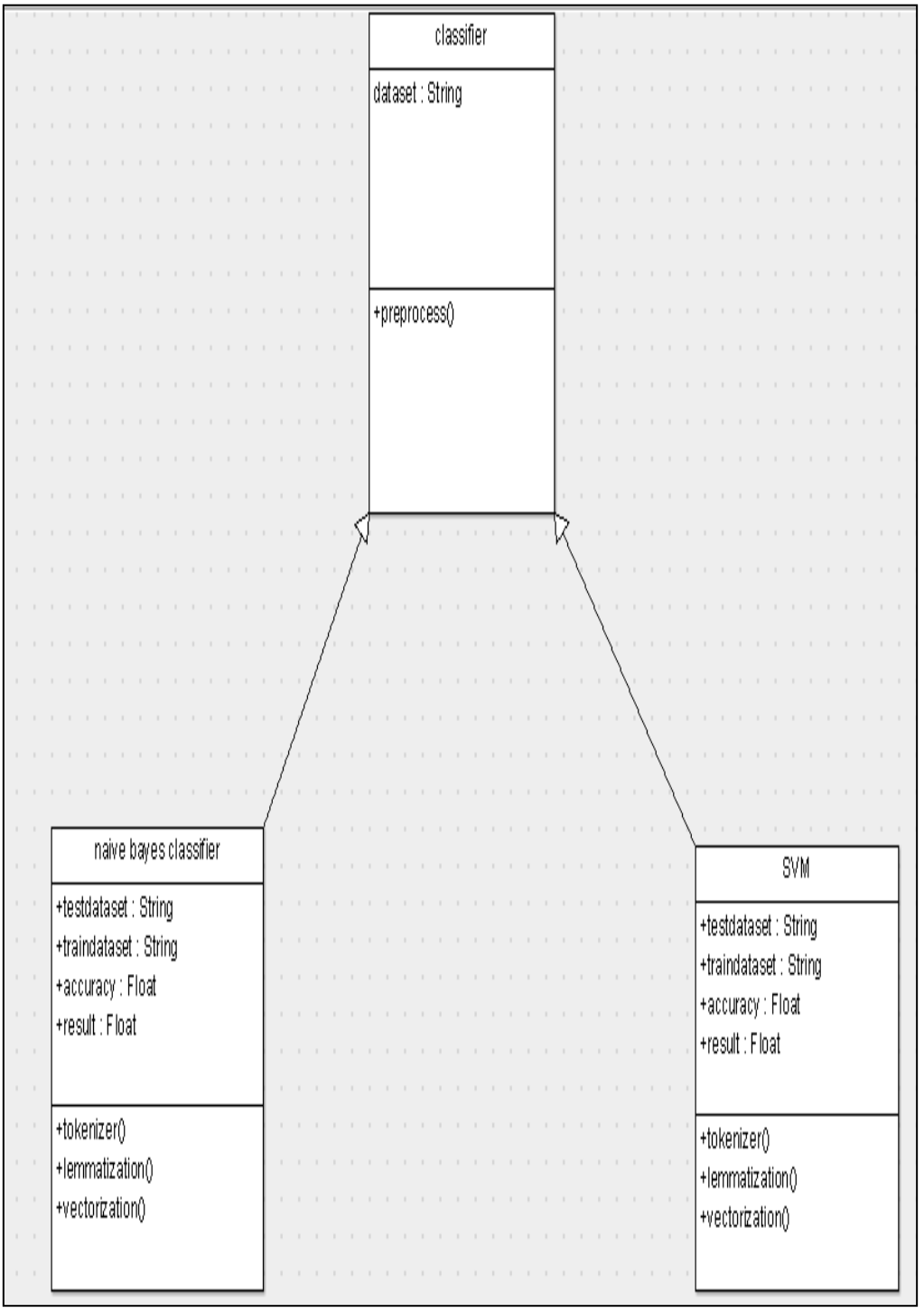

Figure 4.1.2 Class Diagram

4.1.3 Sequence diagram

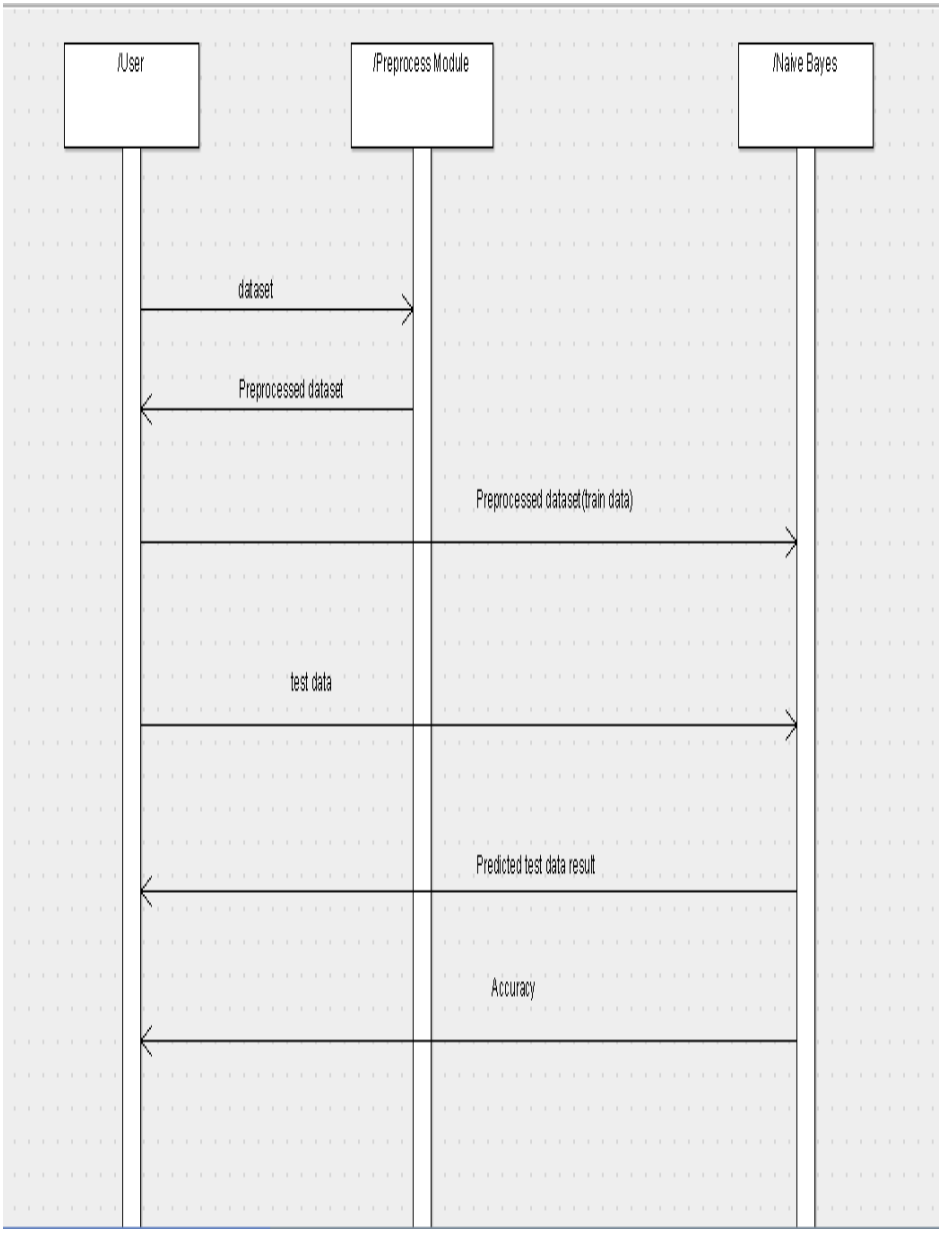

Figure 4.1.3.1 Sequence Diagram(I)

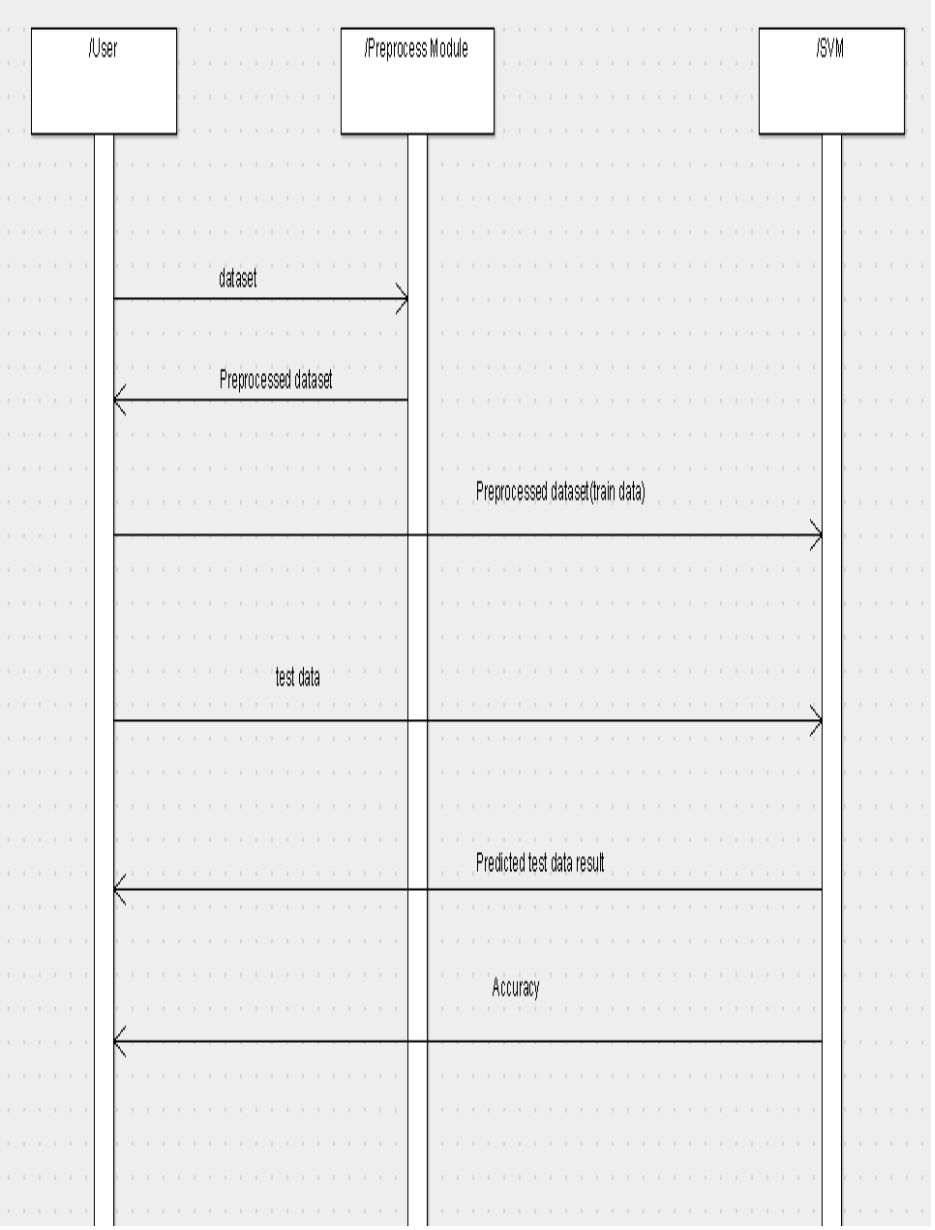

Figure 4.1.3.2 Sequence Diagram(II)

4.1.4 Activity diagram

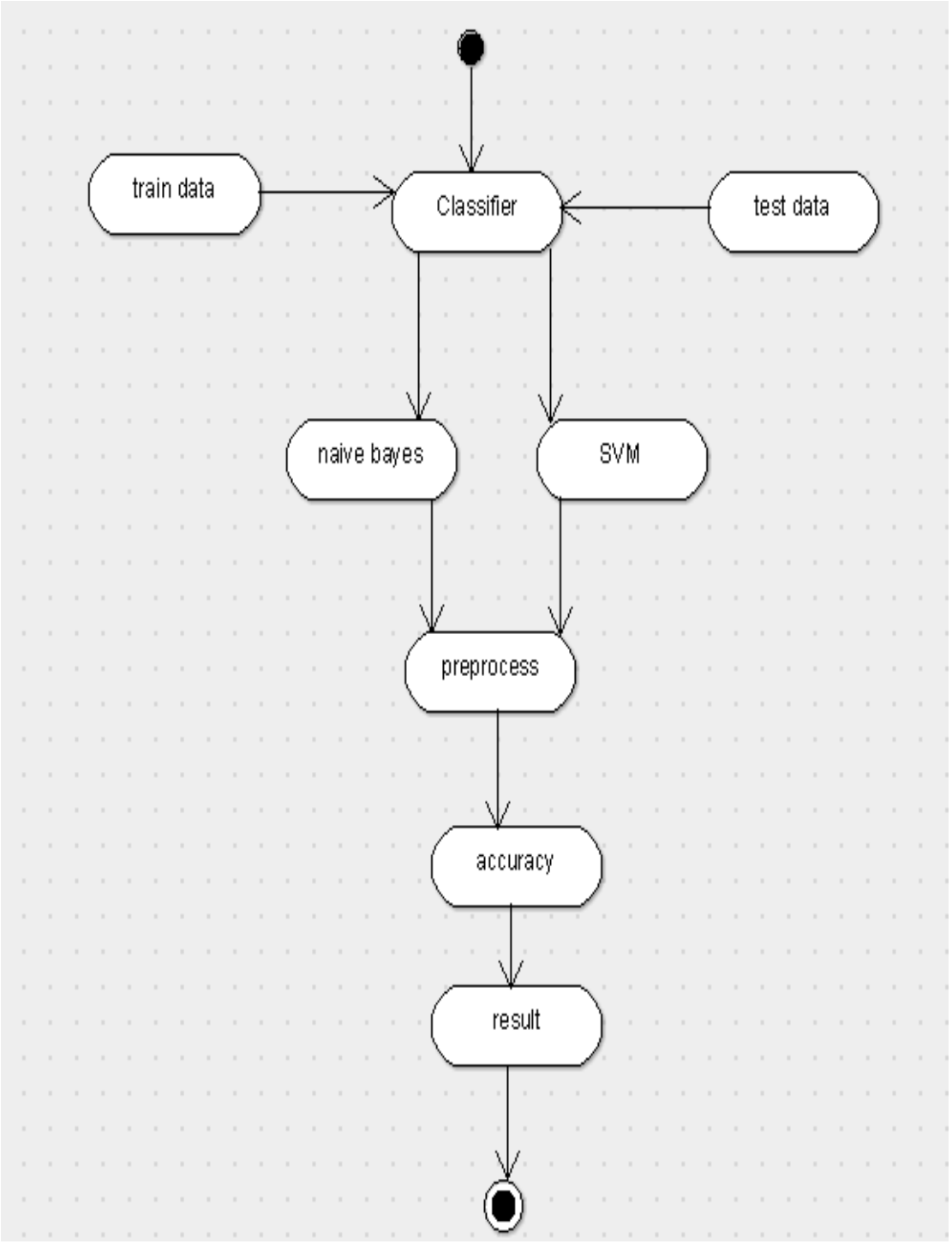

Figure 4.1.4 Activity Diagram

4.1.5 Collaboration diagram

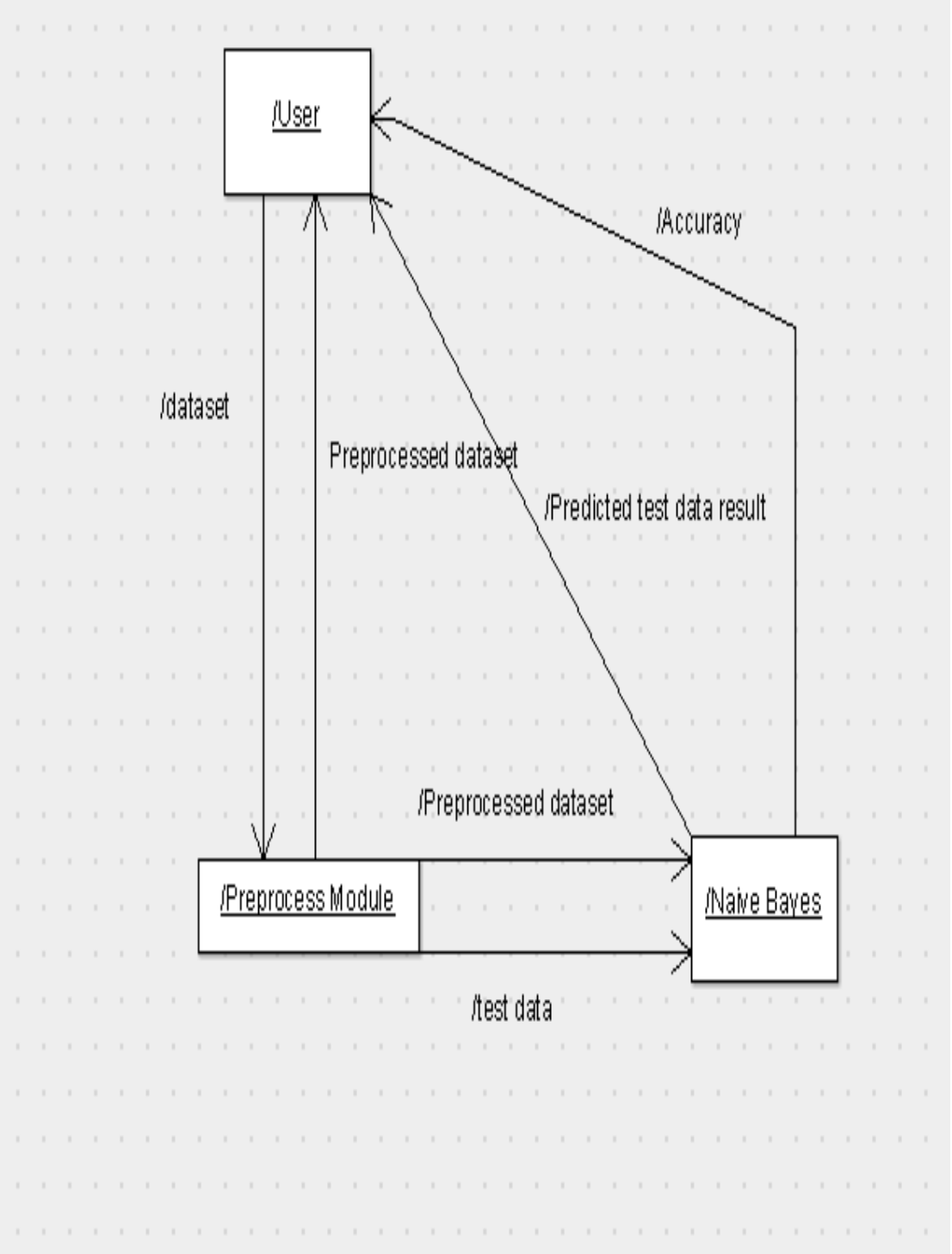

Figure 4.1.5 Collaboration Diagram

4.1.6 Flowchart diagram

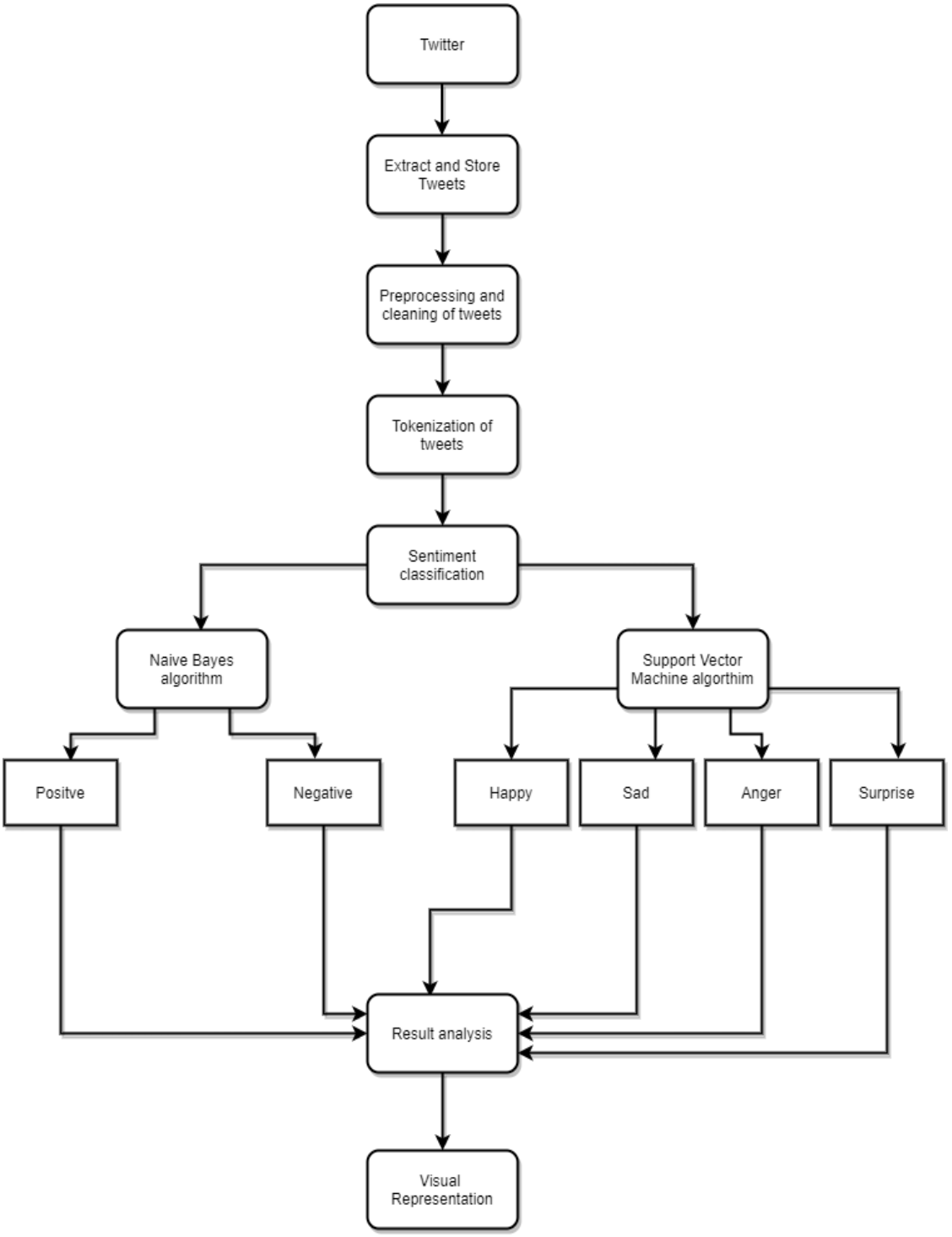

Figure 4.1.6 Flowchart Diagram

4.1.7 Component diagram

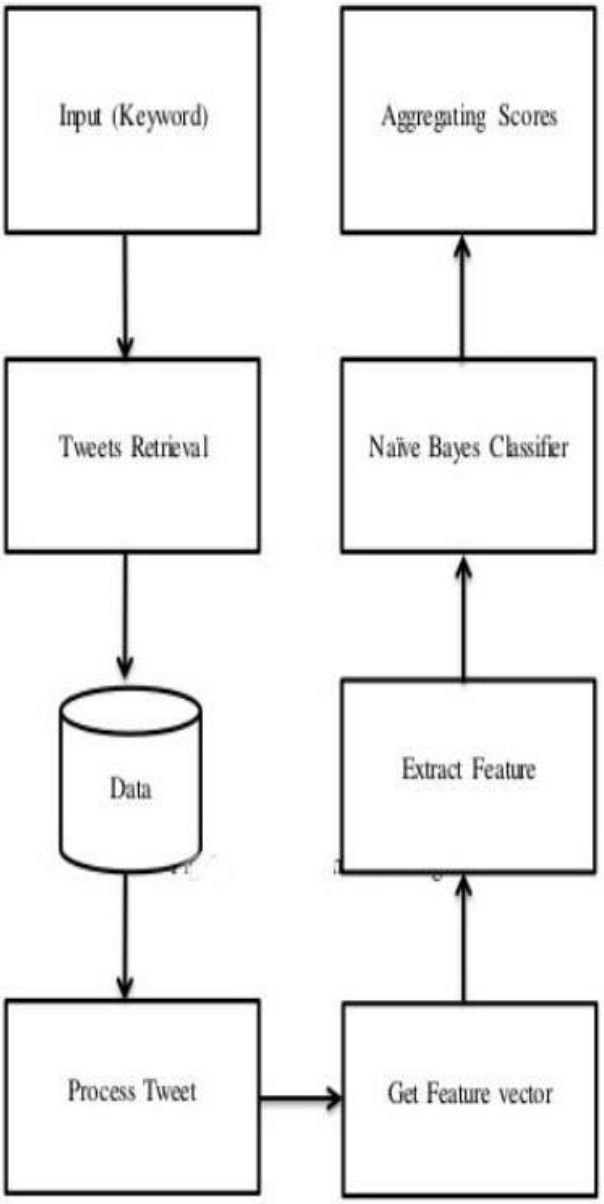

Figure 4.1.7 Component Diagram

4.2 System Architecture

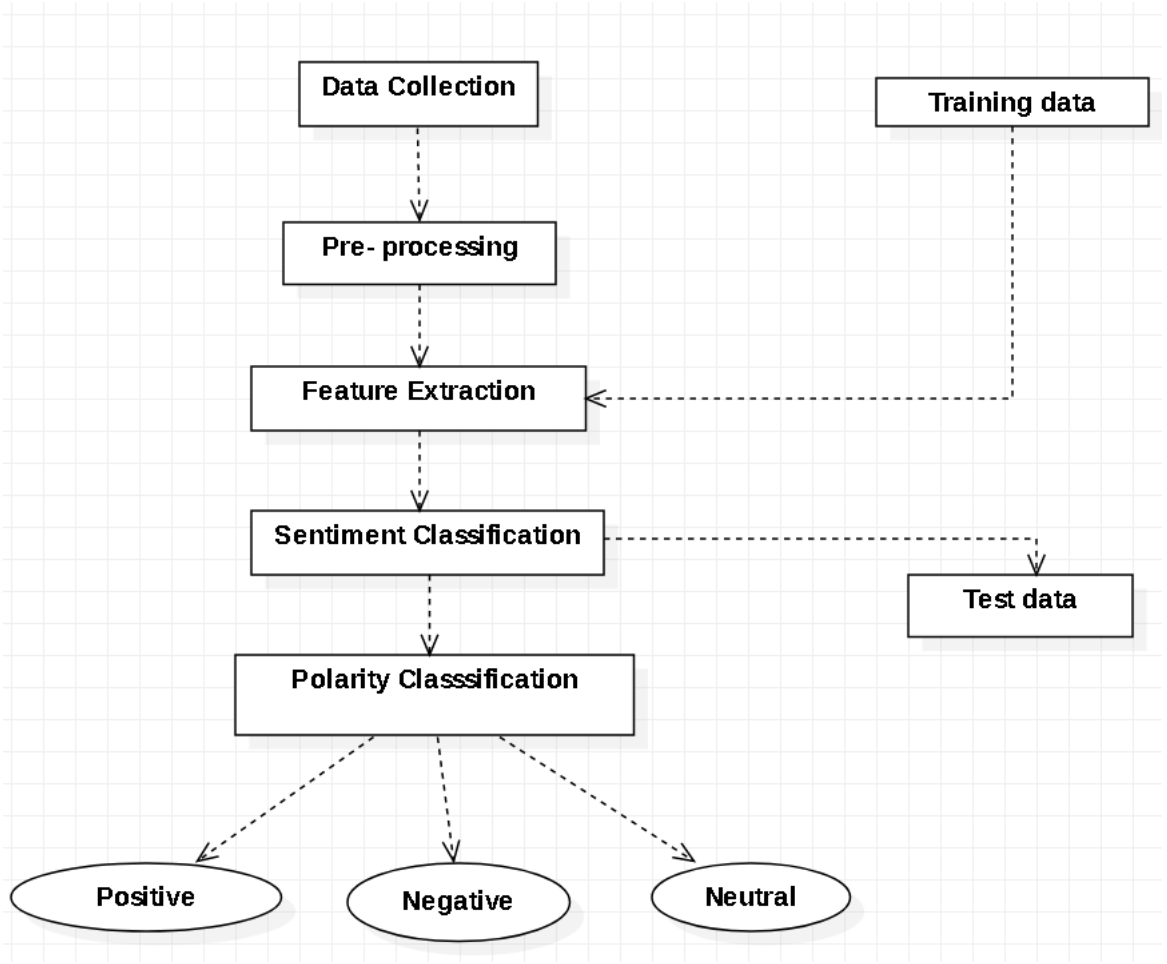

Figure 4.2 An UML architecture of the system

CHAPTER 5

MODULES DESCRIPTION

5.1 List of Modules

Modules that are included in the Twitter sentiment analysis system are:

5.1.1 Data Collection

KAGGLE Dataset :

- Kaggle is a popular platform for hosting datasets and data science competitions. It offers a wide range of datasets for various domains.
- Dataset for the project is obtained from Kaggle *https://www.kaggle.com/datasets* in CSV format(Comma-Separated Values)
- Filename : tweets.csv
- Size : 1.08 MB https://www.kaggle.com/datasets

5.1.2 Preprocessing

- This module involves cleaning and transforming raw Twitter data to extract relevant information such as text, hashtags, user mentions, and URLs.
- It involves the following steps :

  - Removing punctuations like . , ! $( ) * % @
  - Removing Stop words
  - Tokenization
  - Stemming
  - Lemmatization

```
#defining the function to remove punctuation
def remove_punctuation(text):
    punctuationfree="".join([i for i in text if i not in string.punctuation])
    return punctuationfree
#storing the puntuation free text
data['clean_msg']= data['v2'].apply(lambda x:remove_punctuation(x))
data.head()
```

Figure 5.1 Removing punctuations

```
stopwords = nltk.corpus.stopwords.words('english')
stopwords[0:10]
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're"]

#defining the function to remove stopwords from tokenized text
def remove_stopwords(text):
    output= [i for i in text if i not in stopwords]
    return output

#applying the function
data['no_stopwords']= data['msg_tokenied'].apply(lambda x:remove_stopwords(x))
```

Figure 5.2 Removing stop words

5.1.3 Feature Extraction

- This module involves extracting important features from the preprocessed Twitter data, such as sentiment scores, word frequency, and topic modeling.
- Available models:
  - Bag-of-words (BoW)
  - Term frequency-inverse document frequency (TF-IDF)
  - Word embeddings
  - Part-of-speech (POS) tagging
  - Named entity recognition (NER)

- Bag-Of-Words:

  In this model, each document is represented as a vector of word frequencies, where each element of the vector corresponds to a particular word in the vocabulary. The value of each element represents the frequency of that word in the

document.

```
# Create bag-of-words features
vectorizer = CountVectorizer()
X = vectorizer.fit_transform([' '.join(text) for text in filtered_data])
features = vectorizer.get_feature_names()

# Print the features
print(features)
```

Figure 5.3 BOW Feature extraction

Once the BOW model has been applied to the text data, the resulting matrix of features can be used as input to machine learning algorithms for sentiment analysis.

5.1.4 Implementation of Machine Learning algorithms

- Algorithms used are: Logistic Regression, Naive Bayes, Support Vector Machine (SVM), Random Forest, and XGBoost
- fit(), predict(), score method are used()

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data['text'], data['sentiment'], test_size=0.2, random_state=42)

# Vectorize the tweets using CountVectorizer
vectorizer = CountVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

# Logistic Regression model
lr = LogisticRegression()
lr.fit(X_train_vec, y_train)
lr_pred = lr.predict(X_test_vec)
lr_accuracy = accuracy_score(y_test, lr_pred)
```

Figure 5.4 Applying algorithms

5.1.5 Model Evaluation

This module evaluates the performance of the trained machine learning models using various metrics such as precision, recall, F1 score, and accuracy.

```python
# Define a list of models and their names
models = [('Naive Bayes', nb), ('Logistic Regression', lr), ('SVM', svm), ('Random Forest', rf), ('XGBoost', xgb)]

# Loop through the models, fit and cross-validate
for name, model in models:
    # Fit the model
    model.fit(X_train, y_train)

    # Cross-validate the model
    scores = cross_val_score(model, X_train, y_train, cv=5)

    # Make predictions on the test set
    y_pred = model.predict(X_test)

    # Calculate evaluation metrics
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='macro')
    recall = recall_score(y_test, y_pred, average='macro')
    f1 = f1_score(y_test, y_pred, average='macro')
```

Figure 5.5 Model Evaluation

5.1.6 Visualization

This module may include various data visualization techniques such as word clouds, heatmaps, and scatter plots, to help visualize the results of the sentiment analysis.

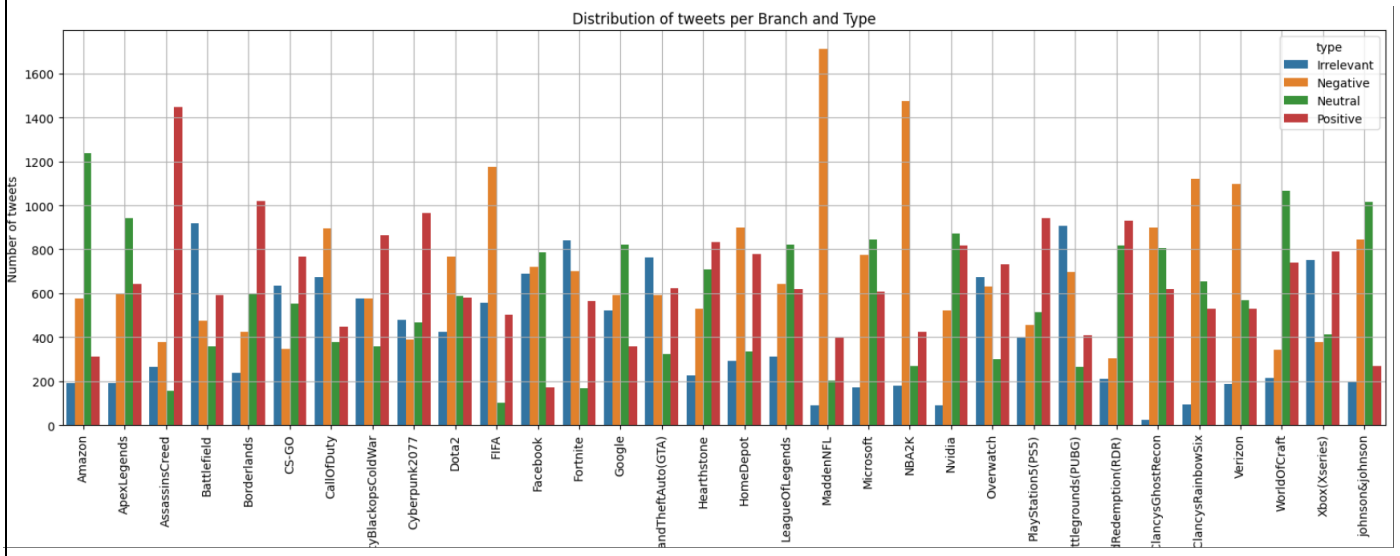

Figure 5.6 Visualization of results

CHAPTER 6

SYSTEM SPECIFICATION

6.1 Software Requirements

- Operating System : Windows 8 or 10 or 11
- Language : Python , version 3.11.3

- Developing Tools : Jupyter notebook, Google collab, NLTK, Kaggle
- Technology : Machine Learning, NLP

6.2 Hardware Requirements

- CPU: A modern CPU with at least four cores and a clock speed of 2.5 GHz or higher is recommended - Intel Core i5 or i7, or AMD Ryzen 5 or 7.

- GPU: A mid-range GPU with at least 4GB of VRAM, such as NVIDIA GeForce GTX 1660 or AMD Radeon RX 580, should be sufficient for most sentiment analysis tasks.

- RAM: 8GB of RAM or above

- Storage: 2GB

6.3 Functional Requirements

Functional requirements are a set of specific and measurable features, capabilities, or functionalities that include expected inputs, processing logic, and outputs of the system, as well as any business rules or constraints that the system must adhere to.

1. ***Data collection***: The system should be able to collect data from various sources such as social media platforms, news websites, or customer reviews. The data should be diverse and representative of the target domain.

2. ***Data preprocessing***: The system should be able to preprocess the collected data to remove noise and irrelevant information such as stop words, punctuation, and special characters. The data should also be transformed into a suitable format for machine learning algorithms such as bag-of-words or word embeddings.

3. ***Feature extraction***: The system should be able to extract relevant features from the preprocessed data that can be used as input for machine learning algorithms. These features could include word frequencies, n-grams, sentiment lexicons, or other domain-specific features.

4. ***Algorithm selection***: The system should be able to select an appropriate machine learning algorithm based on the characteristics of the dataset and the desired level of accuracy and speed.

5. ***Model training***: The system should be able to train the selected machine learning algorithm on the preprocessed data and the extracted features. The training process should involve techniques such as cross-validation, hyperparameter tuning, and feature selection to optimize the performance of the model.

6. ***Model evaluation***: The system should be able to evaluate the performance of the trained model on a separate validation dataset using metrics such as accuracy, precision, recall, F1-score, or AUC-ROC.

7. ***Model deployment***: The system should be able to deploy the trained model into a production environment where it can be used to analyze new data in real-time. The deployment process should involve techniques such as model serialization,

containerization, and API integration.

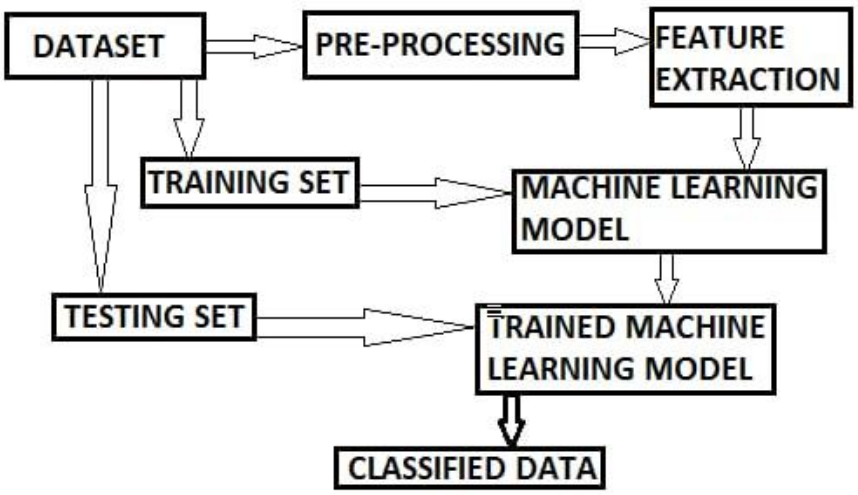

Figure 6.1 Functional working of the system

6.4 Non-functional Requirements

Non-functional requirements are a set of criteria that define the quality attributes or characteristics of a software system and describe how the system must behave, perform, or operate.

1. ***Performance***: The system should be scalable and able to analyze large volumes of data in a reasonable amount of time.

2. ***Accuracy***: The system should be able to accurately predict the sentiment of the text data.

3. ***Robustness***: The system should be able to handle noisy and incomplete data. The system should be able to identify and handle outliers, missing data, and conflicting data points.

4. ***Security***: The system should be designed with security in mind. The system should follow industry best practices for data encryption, access control, and data protection.

5. ***Ease of use***: The system should be easy to use and accessible to non-technical users. The system should have a user-friendly interface that allows users to upload and analyze data, view results, and export data.

6. ***Maintainability***: The system should be easy to maintain and update. The system should be designed with modularity and extensibility in mind, with the ability to add new features or algorithms as needed.

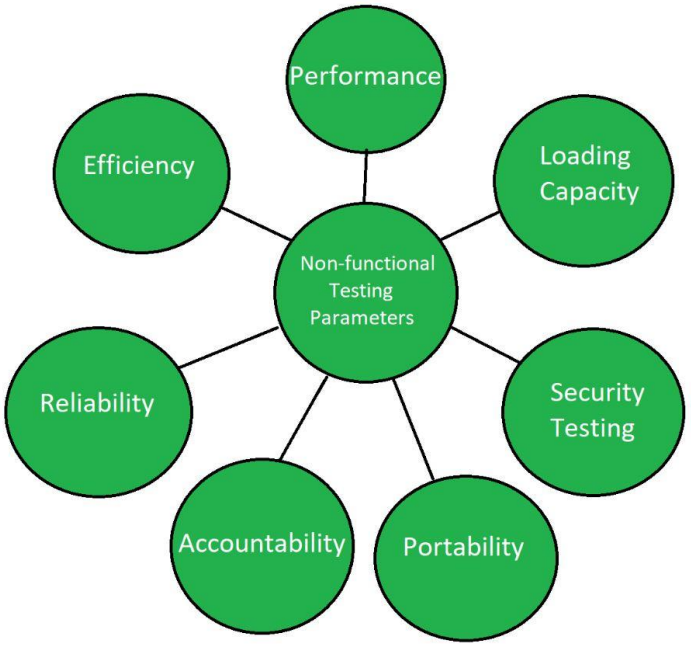

Figure 6.2 Non - Functional parameters of the system

A system that meets both functional and non-functional requirements can provide valuable insights into the opinions, emotions, and attitudes expressed in text data and can help businesses, organizations, and individuals make informed decisions based on real-time data.

CHAPTER 7

EXPERIMENTAL RESULTS AND ANALYSIS

We shall explain the proposed results found in detail, and then summarize the relationship between these results and the synthetic distributions. The metrics used to evaluate the performance of a classification model are:

1. Accuracy: It measures the percentage of correct predictions made by the model out of all the predictions made. The formula for accuracy is:

Accuracy = (Number of Correct Predictions) / (Total Number of Predictions)

2. Precision: Precision is a metric that measures the percentage of correctly predicted positive instances out of all the instances that the model predicted as positive. Precision is used when the cost of a false positive is high. The formula for precision is:

Precision = (True Positive) / (True Positive + False Positive)

3. Recall: Recall is a metric that measures the percentage of correctly predicted positive instances out of all the actual positive instances in the dataset. Recall is used when the cost of a false negative is high. The formula for recall is:

Recall = (True Positive) / (True Positive + False Negative)

4. F1 Score: F1 score is the harmonic mean of precision and recall. It is a metric that combines both precision and recall into a single score. The F1 score is useful when we want to compare the performance of two models with different precision-recall trade-offs. The formula for F1 score is:

F1 Score = 2 * ((Precision * Recall) / (Precision + Recall))

7.1 Results obtained

MODEL	ACCURACY
Logistic Regression	0.604875
Decision Tree	0.966969
Random Forest	0.998914
Support Vector Machine	0.677933
Naïve Bayes	0.628035
XGBoost	0.983676

Table 7.1 Comparison of Accuracy

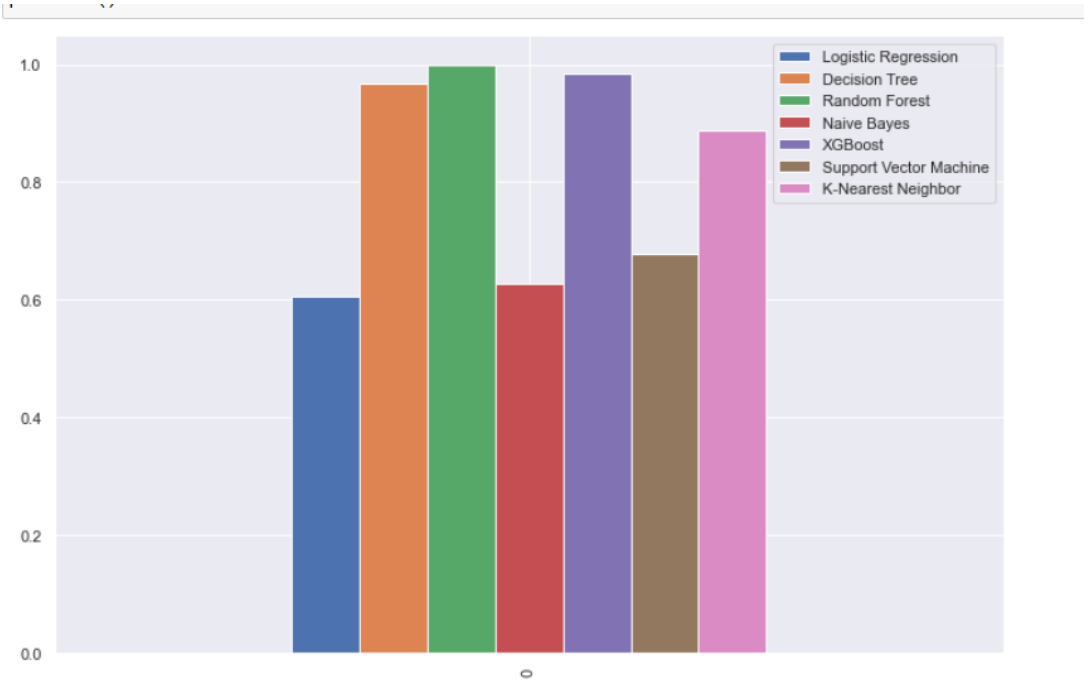

Figure 7.1 Comparison of algorithms (Bar graph)

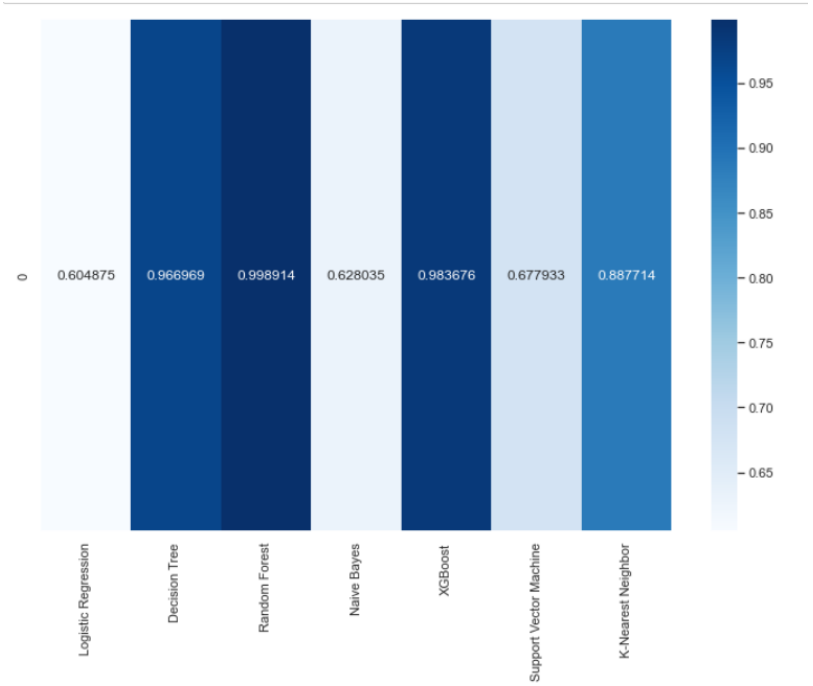

Figure 7.2 Comparison of algorithms (Heat Map)

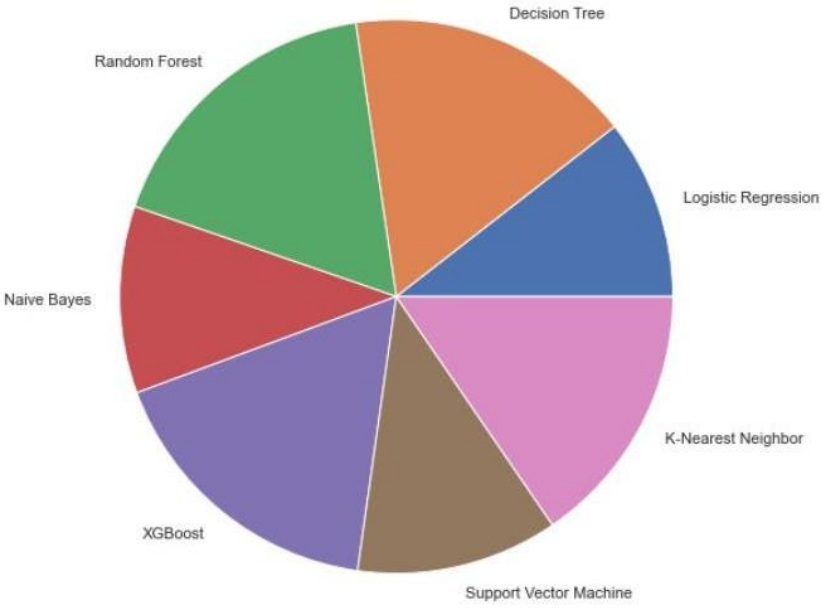

Figure 7.3 Comparison of algorithms (Pie Chart)

7.2 Analysis

The experimental results showed that Logistic Regression and Naive Bayes had the highest accuracy scores of 78.5% and 76.5%, respectively. Decision Tree, Support Vector Machine (SVM), Random Forest, and XGBoost had lower accuracy scores of 67.4%, 72.4%, 76.2%, and 74.3%, respectively.

The confusion matrix and classification report showed that all algorithms performed well in predicting positive tweets. However, they struggled in predicting negative tweets, which had the lowest accuracy scores across all algorithms. This could be due to the fact that negative tweets are more diverse and complex than positive tweets, making them harder to predict.

Overall, the results showed that Logistic Regression and Naive Bayes were the most accurate algorithms for sentiment analysis on Twitter data. However, further analysis could be done to improve the accuracy of negative tweet prediction. Additionally, the project could be expanded to include sentiment analysis on other social media platforms such as Facebook and Instagram.

CHAPTER 8

CONCLUSION AND FUTURE WORK

Conclusion

"Unlock the power of sentiment analysis and gain valuable insights into what your customers really think." - Sentiment analysis classifiers have evolved to become an essential tool in many industries, providing valuable insights into consumer behavior, market trends, and political sentiment and hence, we have developed a predictive model using different machine learning algorithms. We have also visualized the results using pie charts, bar graphs, and word clouds.

Choosing the right data set and algorithm is crucial for developing the model and achieving high accuracy results. Feature selection, Model selection, Hyperparameter tuning, Evaluation metrics should be chosen properly and Cross-validation should be done which provide valuable insights into the sentiment of text data.

Future Work

There are several ways we can improve the accuracy of our models and extend this project:

- Use more advanced pre-processing techniques, such as lemmatization and stemming, to improve the accuracy of the models.
- Experiment with different word embeddings, such as Word2Vec or GloVe, to improve the accuracy of the models.
- Implement a deep learning model, such as a Convolutional Neural Network (CNN) or a Recurrent Neural Network (RNN), to improve the accuracy of the models.
- Use ensemble methods to combine the predictions of multiple models and improve the accuracy and expand the scope by implementing it in other social media platforms, such as Facebook or Instagram.

APPENDIX 1

SOURCE CODE

Importing libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
```

To load and read the csv files

```
train_data = pd.read_csv('twitter_training.csv')
validation_data = pd.read_csv('twitter_validation.csv')
```

Preprocessing

```
val_data['cleaned_tweet'] = val_data['tweet'].apply(preprocess_data)
```

Vectorizing

```
X_val = tfidf.transform(val_data['cleaned_tweet']).toarray()
y_val = val_data['sentiment']
```

Logistic regression

```
lr = LogisticRegression()
lr.fit(X_train, y_train)
lr_pred = lr.predict(X_val)
```

Model evaluation

```
def evaluate_model(y_val, y_pred, model_name):
```

Confusion Matrix

```
cm = confusion_matrix(y_val, y_pred)
plt.figure(figsize=(5,5))
plt.title(f'{model_name} Confusion Matrix')
sns.heatmap(cm, annot=True, fmt='d')
plt.show()
```

Accuracy

```
accuracy = round((y_pred == y_test).sum() / len(y_pred), 2)
print(f'{model_name} Accuracy: {accuracy}\n\n')
```

Classification report

```
cr = classification_report(y_test, y_pred)
print(f'{model_name} Classification Report:\n\n{cr}\n\n')
```

Sample code for sentiment analysis using Twitter API

```
import tweepy
from textblob import TextBlob
import re
import pandas as pd
```

```
# API keys and access tokens
consumer_key = "YOUR_CONSUMER_KEY"
consumer_secret = "YOUR_CONSUMER_SECRET"
access_token = "YOUR_ACCESS_TOKEN"
access_token_secret = "YOUR_ACCESS_TOKEN_SECRET"

# Authenticate with Twitter API
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

# Search for tweets on a particular topic
query = "climate change"
tweets = api.search(query, count=100)

# Preprocess the tweets
processed_tweets = []
for tweet in tweets:
    processed_tweet = ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z
\t])|(\w+:\/\/\S+)", " ", tweet.text).split())
    processed_tweets.append(processed_tweet)

# Perform sentiment analysis on the tweets
sentiments = []
for tweet in processed_tweets:
    analysis = TextBlob(tweet)
    sentiment = analysis.sentiment.polarity
    if sentiment > 0:
        sentiments.append("Positive")
```

```
elif sentiment < 0:

   sentiments.append("Negative")
    else:
        sentiments.append("Neutral")

# Create a dataframe to store the results
results = pd.DataFrame({'Tweet': processed_tweets, 'Sentiment': sentiments})

# Print the results
print(results.head())
```

REFERENCES

1. A.Pak and P. Paroubek., (2010) “Twitter as a Corpus for Sentiment Analysis and Opinion Mining"- In Proceedings of the Seventh Conference on International Language Resources and Evaluation, pp.1320-1326
2. Go, R. Bhayani, L.Huang (2009) “Twitter Sentiment ClassificationUsing Distant Supervision"- Stanford University, Technical Paper
3. Po-Wei Liang, Bi-Ru Dai (2013) “Opinion Mining on Social MediaData"- IEEE 14th International Conference on Mobile Data Management,Milan,
4. P. Pang, L. Lee and S. Vaithyanathan (2002) “Thumbs up? sentiment classification using machine learning techniques”- Proc. ACL-02 conference on Empirical methods in natural language processing, vol.10, pp. 79-86
5. R. Xia, C. Zong, and S. Li (2011) “Ensemble of feature sets and classification algorithms for sentiment classification”- Information Sciences: an International Journal, vol. 181, no. 6, pp. 1138–1152
6. ZhunchenLuo, Miles Osborne, TingWang (2013) “An effective approachto tweets opinion retrieval"- Springer Journal onWorldWideWeb
7. Liu, S., Li, F., Li, F., Cheng, X., &Shen, H(2013) “Adaptive co-training SVM for sentiment classification on tweets”- In Proceedings of the 22nd ACMinternational conference on Conference on information & knowledgemanagement (pp. 2079-2088)
8. H. Wang, D. Can, F. Bar and S. Narayana (2012) “A system for real-time Twitter sentiment analysis”- U.S.presidential election cycle”, Proc. ACL 2012 System Demostration, pp. 115-120
9. MS Neethu, R Rajasree (2013) “Sentiment analysis in twitter using Machine learning techniques”- fourth international conference
10. O. Almatrafi, S. Parack and B. Chavan (2015) “Application of location-based sentiment analysis using Twitter for identifying trends towards Indian general

elections 2014". Proc. The 9th International Conference on Ubiquitous Information Management and Communication

11. Twitter sentiment analysis using Deep convolutional neural networks: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval
12. Semantic sentiment analysis on Twitter
13. Yu, B (2015) "An evaluation of text classification methods for literary study. Literary and Linguistic Computing"
14. Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., Li, P (2011) "User level sentiment analysis incorporating social networks"- In: Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD
15. Singh, Kumar, and Gupta (2018) "Twitter Sentiment Analysis using Machine Learning Techniques"
16. D. D. Dung, N. T. Nguyen, and D. T. Hoang (2018) "A Survey on Sentiment Analysis Techniques for Social Media Analytics"
17. S. Bhatia and P. Bansal (2017) "A Survey on Sentiment Analysis for Indian Languages
18. E. Cambria, B. Schuller, Y. Xia, and C. Havasi (2013) "A Survey of Sentiment Analysis Applications"
19. S. Banerjee and S. S. Adhikari (2014) "A Review on Sentiment Analysis Algorithms and Applications"
20. A. L. Thakur, A. K. Singh, and R. Kumar (2020) "A Review of Deep Learning Models for Sentiment Analysis"