REAL TIME TWITTER SENTIMENT ANALYSIS

A Mini Project Report

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD,
KUKATPALLY, HYDERABAD – 500 085

in partial fulfilment of the requirements for
the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE and ENGINEERING

By

SYEDA FATIMA ALI 18L51A05F2

UMAIMA SIDDIQUA 18L51A05H1

Under the guidance of
Dr. SATHEESH KUMAR S
Associate Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SHADAN WOMEN’S COLLEGE OF ENGINEERING & TECHNOLOGY
KHAIRATABAD, HYDERABAD – 500 004.

DECEMBER 2021

CERTIFICATE

This is to certify that the Mini Project Report titled **“REAL-TIME TWITTER SENTIMENT ANALYSIS”** is being submitted by **“SYEDA FATIMA ALI, 18L51A05F2; UMAIMA SIDDIQUA, 18L51A05H1”** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad, Hyderabad** is a record of bonafide work carried out by her under my guidance and supervision during the academic year 2021 - 2022.

The results presented in this thesis have been verified and are found to be satisfactory. The results embodied in this thesis have not been submitted for the award of any other degree or diploma to this/any other University.

Dr. SATHEESH KUMAR S
ASSOCIATE PROFESSOR
Department of CSE
Shadan Women’s College of Engineering
and Technology, Khairatabad – 500 004.

Ms. M. SRIVANI M.Tech.
HEAD OF THE DEPARTMENT
Department of CSE
Shadan Women’s College of Engineering
and Technology, Khairatabad – 500 004.

JNTUH University Project viva voce Examination held on ____________________

External Examiner

DECLARATION

I hereby declare that the Mini Project Report titled **“REAL-TIME TWITTER SENTIMENT ANALYSIS”** is a record of bonafide work done by me in the Department of Computer Science and Engineering, **Shadan Women’s College of Engineering and Technology, Khairatabad,** submitted to the **Jawaharlal Nehru Technological University Hyderabad, Hyderabad** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering**.

The results embodied in this thesis have not been submitted for the award of any other degree or diploma to this/any other University.

SYEDA FATIMA ALI

18L51A05F2

UMAIMA SIDDIQUA

18L51A05H1

ACKNOWLEDGEMENT

The satisfaction and euphoria of successful completion of any work could be incomplete without mentioning the persons who made it possible, whose constant guidance and encouragement crown our efforts with success.

I take this opportunity to express my grateful acknowledgement to **The Management,** Shadan Women's College of Engineering and Technology, Khairatabad for their kind encouragement and granting permission to do this thesis.

I also express my thanks to **Dr. K. Palani, Principal,** Shadan Women's College of Engineering and Technology, Khairatabad for providing infrastructure and facilities.

I express my sincere gratitude to **Ms. M. Srivani, Head of the Department of CSE,** for her extended help, concern and persistent encouragement and support.

I wish to express my heartfelt thanks and sincere acknowledgement to my guide **Dr. Satheesh Kumar S, Associate Professor** for his/her concern and help to direct me for every move in this thesis.

I acknowledge with grateful thanks to the authors of the references and other literatures referred in this thesis.

I finally thank my parent, friends and relatives who render their help directly or indirectly for the completion of this thesis.

SYEDA FATIMA ALI

18L51A05F2

UMAIMA SIDDIQUA

18L51A05H1

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Sentiment analysis alludes to the application for processing natural language, text analysis, computational linguistics, and biometrics to methodically recognize, extract, quantify, and learn affective states and subjective information in source material. Twitter, being one among several popular social media platforms, is a place where people often choose to express their emotions and sentiments about a brand, a product or a service. Analysing sentiments for tweets is very helpful in determining people's opinion as positive, negative or neutral. This thesis evaluates a person’s tweets to know whether he/she builds a positive or negative impact on people. Twitter API is used to access the tweets directly from twitter and build a sentiment classification for the tweets. The outcome of the analysis is depicted for positive, negative and neutral remarks about their opinions using visualization techniques such as a histogram and a wordcloud.

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Social media platforms are now an inherent part of our lives, boasting a whopping 4.55 billion active users. The dynamics of communication in all spheres of life has changed. The reason for the rapid growth of these platforms is they enable their users to share their thoughts, ideas, and insights all the while being extremely fast, open, free, and easy to access. Social media provide a platform through which users can freely share information simultaneously with a significantly larger audience than traditional media. It gives us the ability to discover what's happening in the world in real-time and to have access to endless amounts of information at our fingertips. In many senses, social media has helped many individuals find common grounds with others online, making the world seem more approachable.

Online social platforms including Twitter have become a significant platform for information sharing. They offer organizations a fast and effective way to analyse customers' perspectives which are critical to their success in the market place. Twitter-enabled analytics do not only constitute a valuable source of information but provide an uncomplicated extraction and dissemination of subject specific information for government agencies, businesses, political parties, financial institutions, fundraisers and many others. However, due to their explosive spreading nature, they turn into mediums for wrongdoers to spread different forms of hate or prejudice. In a recent study, 10 million tweets from 700,000 Twitter accounts were examined. The collected accounts were linked to 600 fake news and conspiracy sites. Therefore, it becomes apparent that being able to classify large amounts of data based on user sentiment is extremely beneficial and important.

Sentiment analysis is the process of detecting positive or negative sentiment in text. The objective of sentiment analysis is to analyse the attitude of various individuals to a particular subject using (Natural Language Processing) NLP. This is done by determining the polarity of the given text which ranges between -1 to +1, where -1 denotes a negative sentiment and +1 denotes positive. Subjectivity in sentiment analysis (Range: 0-1) shows whether the sentence is subjective or not, where subjective sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information.

1.2 MOTIVATION

We have chosen to work with Twitter since we feel it is a better approximation of public sentiment as opposed to conventional internet articles and web blogs. The reason is that the amount of relevant data is much larger for twitter, as compared to traditional blogging sites. Moreover, the response on Twitter is more prompt and also more general (since the number of users who tweet is substantially more than those who write web blogs on a daily basis). Sentiment analysis of public is highly critical in macro-scale socioeconomic phenomena like predicting the stock market rate of a particular firm. This could be done by analysing overall public sentiment towards that firm with respect to time and using economics tools for finding the correlation between public sentiment and the firm's stock market value. Firms can also estimate how well their product is responding in the market, which areas of the market is it having a favourable response and in which a negative response (since Twitter allows us to download stream of geo-tagged tweets for particular locations). If firms can get this information, they can analyse the reasons behind geographically differentiated response, and so they can market their product in a more optimized manner by looking for appropriate solutions like creating suitable market segments. Predicting the results of popular political elections and polls is also an emerging application to sentiment analysis.

1.3 SCOPE

Sentiment Analysis is an exceptionally versatile tool with its applications spanning diverse fields like commerce, politics, education, finance etc.

1.3.1 Sentiment Analysis for Businesses

- Social Media Monitoring**:**

Social media posts often present some of the most truthful points of view about products, services, and businesses because users offer their opinions unsolicited. They are simply compelled to tell the world how they feel. Sentiment analysis of social data will keep an eye on customer opinion 24/7 and in real time. Businesses will be able to quickly respond when something negative starts circulating and boost their image when they receive positive mentions. And, they will get regular, dependable insights about their customers, which they can use to monitor their progress from one quarter to the next.

- Customer Support:

Customer support management presents many challenges due to the sheer number of requests, varied topics, and diverse branches within a company – not to mention the urgency of any given request. Sentiment analysis with natural language processing (NLP) reads regular human language for meaning, emotion, tone, and more, to understand customer requests, just as a person would. Businesses can automatically process customer support tickets, online chats, phone calls, and emails by sentiment, which might also indicate urgency, and route to the appropriate team. Sentiment analysis can automatically mark thousands of customer support messages instantly by understanding words and phrases that indicate negativity.

- Customer Feedback:

Sentiment analysis can also be used to gain insights from the troves of customer feedback available (online reviews, social media, surveys) and save hundreds of employee hours. Sentiment analysis would classify the second comment as negative, even though they both use words that, without context, would be considered positive. Keeping track of customer comments allows businesses to engage with individual customers in real time. And help target read for new products or specific user issues.

- Brand Monitoring and Reputation Management:

Brand monitoring is one of the most popular applications of sentiment analysis in business. Bad reviews can snowball online, and the longer you leave them the worse the situation will be. With Sentiment analysis tools, businesses will be notified about negative brand mentions immediately. Not only that, they can keep track of their brand’s image and reputation over time or at any given moment, so they can monitor their progress. Whether monitoring news stories, blogs, forums, and social media for information about their brand, they can transform this data into usable information and statistics. They can also trust machine learning to follow trends and anticipate outcomes, to stay ahead and go from reactive to proactive.

- Voice of Customer:

Combine and evaluate all customer feedback from the web, customer surveys, chats, call centres, and emails. Sentiment analysis allows them to categorize and structure this data to identify patterns and discover recurring topics and concerns. Businesses can understand their customer base collectively, then segment them to target directly. For example, using data from a customer survey, they might want to offer free services or promotions to entice unhappy customers. Or offer rewards to those that are extremely happy with their company, encouraging them to spread the word about their product or service. Listening to the voice of customers, and learning how to communicate with them – what works and what doesn't – will help create a personalized customer experience.

- Voice of Employee:

Engage employees, reduce turnover, and increase productivity. Sentiment analysis software allows businesses to analyse employee opinions subjectively, with no human input. "Process unstructured data to go beyond *who* and *what* to uncover the *why*." Creating analysis models for their specific needs, businesses will discover the most common topics and concerns to keep their employees happy and productive.

- Product Analysis:

Finding out what the public is saying about a new product right after launch, or analyse years of feedback they might never have seen. Businesses can search keywords for a particular product feature (interface, UX, functionality) and train sentiment analysis models to find only the information they need. Discovering how a product is perceived by the target audience, which elements of the product need to be improved. Sentiment analysis provides better results than humans and it's not subjective.

- Market and Competitor Research:

Another use case of sentiment analysis is market and competitor research. Finding out who’s trending among their competitors and how their marketing efforts compare. Getting a comprehensive view from the ground, from every aspect of their and their competition’s customer base. Analysing competitor’s content to find out what works with the public that businesses might not have considered. They will understand their strengths and weaknesses and how they relate to that of the competitors.

- Example of Sentiment Analysis Used in Business (Google):

A good showcase of how sentiment analysis application contributes to product improvement can be seen in Google’s output. Taking the Chrome browser for example. Google Chrome’s development team is constantly monitoring user feedback, whether it is direct or indirect (i.e., presented in the open sources, most notably, blogs). But they are not looking at feedback as a message from the user but rather as a sum of its parts: the sentiment itself (positive or negative), Mentions of the specific aspects of the product - whether it is scalability, extensions, security, or UI, Sentiments, wishes, and recommendations regarding the product in general and its specific elements.

- The Result:

These elements provide an additional perspective on the weak and strong points of the product. This subsequently contributes to further research and development of the product.

1.3.2 Sentiment Analysis in Finance

By analysing articles, news and social media info about public companies, data analysts can assign scores for trading systems – such as Stock Sonar – and form stock prices. Making investments, especially in the business world, is quite tricky. The stocks and market are always on the edge of risks, but they can be condensed if you do correct research before investing. Now if you are looking to invest in the automobile industry and are confused about choosing between company X and company Y, you can look at the sentiments analysis from the company for their latest products. It will help you to find the one that is performing better in the market.

1.3.3 Sentiment Analysis in Politics

Many leaders use Twitter as their primary mode of communication to the public rather than addresses or newsletters. Therefore, we have a treasure trove of data for analysis. By analysing sentiments regarding a particular politician or political party, one can make projections as to who has the highest probability to win in an election and this can be a useful political advice. Campaign managers use sentiment analysis to understand how people feel about certain issues in the politician's program or rhetoric, how they react to speeches and actions, etc.

1.3.4 Sentiment Analysis in Public relations

Sentiment analysis gives PR professionals the ability to analyse millions of tweets and other written social media posts to identify main topics of discussion or to measure audience sentiment (positive, negative or neutral) – helping them assess public opinion about their brands or clients. By analysing social media content and news feeds, PR professionals can optimize traffic flow, ensure better public security and resolve issues before they become too pressing. According to the Chartered Institute for Public Relations study, "Humans Still Needed," social media analysis is one of the most common PR activities impacted by artificial intelligence. This explains why almost half (44%) of the PR professionals who participated in the 2019 USC Annenberg Global Communication Report chose media monitoring as the most relevant tool for their current work.

- Need of Sentiment Analysis in PR**:**

Public sentiment toward an organization can change overnight – or even within hours as negative comments spread virally online. In some scenarios, the news media promptly reports their complaints. When a passenger was forcibly removed from a United Airlines aircraft, other passengers posted videos of the violent removal. Videos and damaging comments spread virally within hours, leading to extensive critical media coverage the following day. Real-time sentiment analysis could have alerted United Airlines about the imminent PR crisis and depth of public outrage, and encouraged it to respond more humbly rather than issue a feeble apology. Real-time sentiment analysis combined with email notifications can alert PR personnel of a spike in negative sentiment that portends a coming PR crisis. PR can then take swift actions to respond on social media and to media outlets, if needed, to limit negative reactions.

CHAPTER 2

LITERATURE SURVEY

TITLE: Un-Compromised Credibility: Social Media Based Multi-Class Hate Speech Classification for Text.

AUTHORS: Khubaib Ahmed Qureshi, Muhammad Sabih, DHA Suffa University, Karachi, Pakistan.

YEAR: 2021

DESCRIPTION:

In this study, major challenges are identified first and the complex problem of multi-class automated hate speech classification for text is accomplished with much better results. Ten separate binary classified datasets consisting of different hate speech categories are constructed. Each dataset was annotated by experts with the strong agreement of annotators under comprehensive, clear definition and well-defined rules. Datasets were well balanced and broad. They were also supplemented with language subtleties. Compilation of such dataset was achieved as necessary requirement for filling the gap of the field. After the development of high-quality datasets, a list of effective, commonly used and recommended features extracted from related studies under the field of text mining were identified. In addition to these features our own potential features were also proposed. These features were then explored and identified with respect to their problem objective. It is found that character 2 to 4-grams, word 1 to 5-grams, dependency tuples, sentiment scores, and count of 1st, and 2nd person pronouns were very effective. Latent Semantic Analysis (LSA) as a dimensionality reduction algorithm was also applied and found much effective in such high dimensional classification problems. [1]

Datasets were completely explored through tSNE multi-dimensional plots. These plots identified issues like the need for appropriate discriminating features, complex data overlaps, and non-linearity. Therefore complex, non-linear models were used for classification, and the most popular and advanced machine learning model

CATBoost was found top-performing over all datasets. CATBoost has shown the best average scores, in terms of Accuracy, F1, and AUC, which were 89.03, 87.74, and 88.88, respectively. These results seem quite appreciating, considering the context of the hate speech problem's criticality. Similarly, the Gradient Boosting model performed next to CATBoost with minor difference, which scored 88.78, 86.04, and 87.69 under the same measures of Accuracy, F1, and AUC, respectively. Random Forest stood at the top 3rd with slight variation in scores, which are 86.45, 85.53, and 86.76 corresponding to Accuracy, F1, and AUC, respectively. The performance of the final model is also compared with two related studies and our initial baseline. It is worth mentioning that the model outperformed all these.[1]

TITLE: Using a Hybrid-Classification Method to Analyze Twitter Data During Critical Events

AUTHORS: Saadat M. Al Hashmi, Ahmed M. Khedr, Ifra Arif, Magdi El Bannany, University of Sharjah, Sharjah, UAE

YEAR: 2021

DESCRIPTION:

This paper introduced a novel hybrid classification approach to analyze the feelings of tweets using the SVM and BFTAN methods. The method was tested on two Twitter datasets, the COVID-19 and the Expo2020 datasets. It classifies the input tweets into four phases: (i) collecting data, (ii) preprocessing of the tweets, (iii) feature extraction, (iv) proposed hybrid classification approach. Our hybrid-based approach is proposed to address the following challenges: improving accuracy, identifying the polarity of comparative sentences, distinguishing the intensity of opinion words, considering negative comments, and handling Sarcasm. Results demonstrate the efficacy of the suggested approach based on the accuracy and class distribution of each dataset. The approach was compared with other classifiers - BFTAN, TAN, NB, SVM and RF. Accuracy, precision and recall were computed for all the considered datasets. When enough data is available to support training examples, Bayesian classifier shows effective performance. [2]

TITLE: Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis.

AUTHORS: Zhao Jianqiang, Gui Xiaolin, Key laboratory of Computer Network of Shaanxi Province, Xi’an, China.

YEAR: 2017

DESCRIPTION:

This paper studies that six different pre-processing methods affect sentiment polarity classification in the Twitter. We conduct a series of experiments using four classifiers to verify the effectiveness of several pre-processing methods on five Twitter datasets. Experimental results indicate that the removal of URLs, the removal of stop words and the removal of numbers minimally affect the performance of classifiers; furthermore, replacing negation and expanding acronyms can improve the classification accuracy. Therefore, removing stop words, numbers, and URLs is appropriate to reduce noise but does not affect performance. Replacing negation is effective for sentiment analysis. We select appropriate pre-processing methods and feature models for different classifiers for the Twitter sentiment classification task. [3]

TITLE: Seeing and Believing: Evaluating the Trustworthiness of Twitter Users

AUTHORS: Tanveer Khan, Antonis Michalas, Tampere University, Tampere, Finland

YEAR: 2021

DESCRIPTION:

Contemplating the momentous impact unreliable information has on our lives and the intrinsic issue of trust in OSNs, our work focused on finding ways to identify this kind of information and notifying users of the possibility that a specific Twitter user is not credible. To do so, we designed a model that analyses Twitter users and assigns each a calculated score based on their social profiles, tweets credibility, sentiment score, and h-indexing score. Users with a higher score are not only considered as more influential but also, as having a greater credibility. To test our approach, we first generated a dataset of 50,000 Twitter users along with a set of 19 features for each user. Then, we classified the Twitter users into trusted or untrusted using three different classifiers. Further, we employed the active learner approach to label the ambiguous unlabelled instances. During the evaluation of our model, we conducted extensive experiments using three sampling methods. The best results were achieved by using RFC with the margin sampling. We believe this work is an important step towards automating the users' credibility assessment, re-establishing their trust in social networks, and building new bonds of trust between them. [4]

CHAPTER 3

PROBLEM ANALYSIS

3.1 EXISTING SYSTEM : (LEXICON BASED APPROACH)

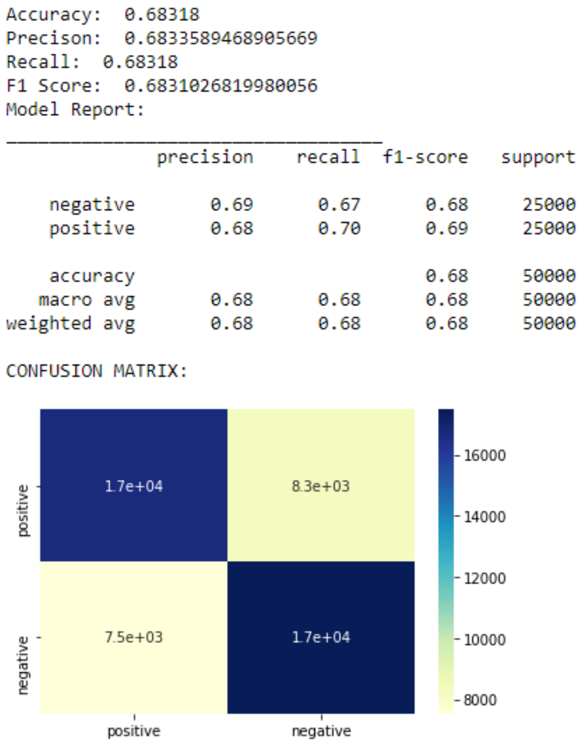
Lexicon means the vocabulary of a person, language or branch of knowledge. Here, in lexicon-based sentiment analysis we already have a given set of dictionaries of words with each labeled as positive negative, neutral sentiments along with polarity, parts of speech and subjectivity classifiers, mood, modality and the like. A sentence is tokenized and each token is matched with the available words in the model to find out its context and sentiment (if any). A combining function such as sum or average is taken to make the final prediction regarding the total text component.

Lexicon-Based Approach uses sentiment lexicon with information about which words and phrases are positive and which are negative. A sentiment lexicon is a list of lexical features which are generally labeled according to their semantic orientation as either positive or negative. Researchers first create a sentiment lexicon through compiling sentiment word lists such as manual approaches, lexical approaches, and corpus-based approaches, then determine the polarity score of the given review based on the positive and negative indicators which are identified in the lexicon. There are some lexicons like LIWC (Linguistic Inquiry and Word Count), GI (General Inquirer) that categorizes the words into positive and negative according to their context free semantic orientation. LIWC consists of almost 4,500 words organized into one of 76 categories, including 905 words in two categories especially related to sentiment analysis. LIWC was well-established and validated in a process spanning tree more than a decade of work by sociologists, psychologists, linguists. Though its extensive use to find sentiment analysis in social media text, LIWC does not include acronyms, initialisms, emoticons, and slang which are important factors for sentiment analysis of social media text.

However, other lexicons like ANEW (Affective Norms for English Words), SentiWordNet, and SenticNet are associated with valence scores for sentiment intensity. SentiWordNet consists of 1,47,306 synsets are annotated with three sentiment scores such as positive negative and objective [5]. Though, it is not a gold standard resource like Word Net, but is useful for wide range of tasks. One of the major advantages of Lexicon-Based approach is, its domain independence, and also it can be easily extended and improved.

Let us consider a few examples of lexicon-based sentiment analysis. IMDB movie reviews dataset is used to make the following predictions.

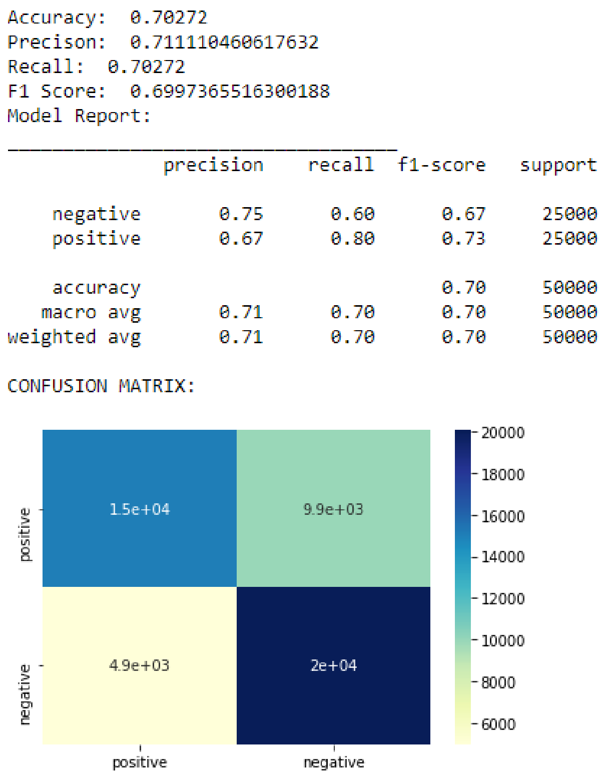
SentiWordNet: It is a lexical resource for opinion mining. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity. [6]



```
Accuracy:  0.68318
Precison:  0.6833589468905669
Recall:  0.68318
F1 Score:  0.6831026819980056
Model Report:
___________________________________
              precision    recall  f1-score   support

    negative       0.69      0.67      0.68     25000
    positive       0.68      0.70      0.69     25000

    accuracy                           0.68     50000
   macro avg       0.68      0.68      0.68     50000
weighted avg       0.68      0.68      0.68     50000

CONFUSION MATRIX:
```

Fig.3.1: SentiWordNet Performance Model

VADER: It is a lexicon and rule-based sentiment analysis tool that is open-sourced and is available in the NLTK package which can be applied directly to unlabelled text data. VADER is capable of detection of polarity and intensity of emotion. [6]



```
Accuracy:  0.70272
Precison:  0.711110460617632
Recall:  0.70272
F1 Score:  0.6997365516300188
Model Report:
___________________________________
              precision    recall  f1-score   support

    negative       0.75      0.60      0.67     25000
    positive       0.67      0.80      0.73     25000

    accuracy                           0.70     50000
   macro avg       0.71      0.70      0.70     50000
weighted avg       0.71      0.70      0.70     50000

CONFUSION MATRIX:
```

Fig.3.2: VADER Performance Model

3.1.1 Limitations of the Existing System

- **Incorrect Scoring:**

Incorrect sentiment scoring of opinion words by the existing lexicons, such as sentiwordnet.[13] The sentiment score of a word is generally dependent on a particular domain and changes when a domain switch occurs. To address this issue, domain specific vocabulary is introduced to improve the efficacy of sentiment classification.

- **Emoticon Handling:**

The emoticon handling issue arises due to insufficient coverage of emoticons expressed by users in their posts. [13] Incorporating emoticon handling features in the proposed setup may increase the accuracy of the model by approximately 10%.

- **Domain Limitation Problem:**

Poor performance can be attributed to the fact that some domain-specific sentiment-bearing terms may not be available from a general knowledge lexicon. [13] Insufficient labeled data leads to the incapability to deal with complex sentences.

3.2 PROPOSED SYSTEM: (MACHINE LEARNING APPROACH)

To overcome the limitations faced in simple lexicon approach we use Textblob and machine learning algorithms and to support live Twitter analysis we use Tweepy.

3.2.1 Machine Learning

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. Recommendation engines are a common use case for machine learning. Other popular uses include fraud detection, spam filtering, malware threat detection, business process automation (BPA) and predictive maintenance.[14] Machine learning is important because it gives enterprises a view of trends in customer behaviour and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm data scientists chooses to use depends on what type of data they want to predict. [15]

- Supervised Learning: In this type of machine learning, data scientists supply algorithms with labelled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.

- Unsupervised Learning: This type of machine learning involves algorithms that train on unlabelled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.

- Semi-Supervised Learning: This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labelled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.

- Reinforcement Learning: Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

3.2.1.1 Classification of Sentiment with Supervised Learning

With supervised learning we get each textual data along with label of their polarity, subjectivity and objectivity. Here we need to build a machine learning model to classify and predict future inputs into different categories of sentiments. [6]

- Text Pre-Processing and Data Normalization: It is the most important thing before training the model. We must have balanced class distribution so that no classes are biased. Furthermore, we have to remove punctuations, html tags, numbers, convert accented characters to ASCII along with lowercasing all texts.
- Feature Engineering: Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. — Dr. Jason Brownlee.

- Model Training, Prediction and Evaluation: This involves using various ML models to train the data and evaluate them to see which is the most appropriate for our use case. We must do hyper parameter tuning to achieve more accurate results.

This supervised learning is done with the help of TextBlob which uses Naive Bayes theorem.

3.2.2 TextBlob

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. [12]

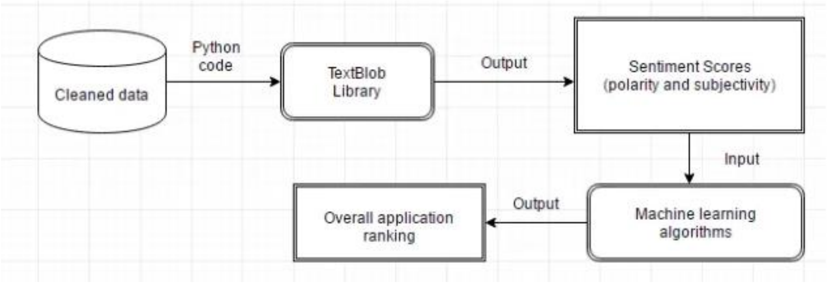

Cleaned data
Python code
TextBlob Library
Output
Sentiment Scores (polarity and subjectivity)
Input
Machine learning algorithms
Output
Overall application ranking

Fig.3.3: TextBlob Architecture

Installing TextBlob:

Step #1: Execute pip install TextBlob on Anaconda/command prompt.

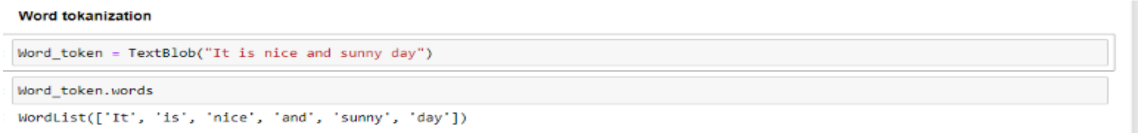
Step #2: Once installed users can provide the data and analyse the sentiments over it.

Features of TextBlob:

- Word Tokenization:

Word tokenization is the process of splitting a large sample of text into words. This is a requirement in natural language processing tasks where each word needs to be captured and subjected to further analysis like classifying and counting them for a particular sentiment etc.

TextBlob which is based on NLTK so, methods are used directly.



Word tokanization

```
Word_token = TextBlob("It is nice and sunny day")
```

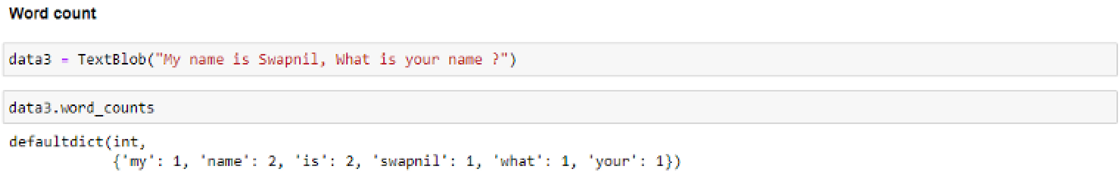
```
Word_token.words
```

```
WordList(['It', 'is', 'nice', 'and', 'sunny', 'day'])
```

Fig.3.4: Tokenization

- Word Count:

Calculating the word count in given message and work on those according to a requirement.



Word count

```
data3 = TextBlob("My name is Swapnil, What is your name ?")
```

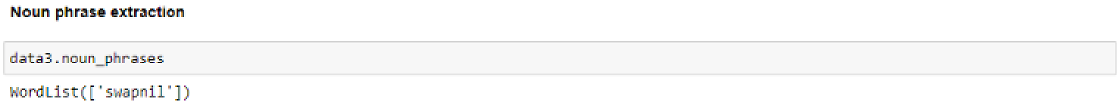
```
data3.word_counts
```

```
defaultdict(int,
            {'my': 1, 'name': 2, 'is': 2, 'swapnil': 1, 'what': 1, 'your': 1})
```

Fig.3.5: Count

- Noun Phrase Extraction:

Noun Phrase extraction is particularly important to analyse the “who” in a sentence.



Noun phrase extraction

```
data3.noun_phrases
```

```
WordList(['swapnil'])
```

Fig.3.6: Noun Phrase Extraction

3.2.3 Sentiment Analysis using TextBlob

Sentiment analysis is basically the process of determining the attitude or the emotion of the writer, i.e., whether it is positive or negative or neutral. The *sentiment* function of textblob returns two properties, polarity, and subjectivity:

- Polarity: is a float value within the range [-1.0 to 1.0] where 0 indicates neutral, +1 indicates a very positive sentiment and -1 represents a very negative sentiment.

- Subjectivity: is a float value within the range [0.0 to 1.0] where 0.0 is very objective and 1.0 is very subjective. Subjective sentence expresses some personal feelings, views, beliefs, opinions, allegations, desires, beliefs, suspicions, and speculations whereas Objective sentences are factual.

Let’s check the sentiment of our blob.

```
print (blob)

blob.sentiment

>> Analytics Vidhya is a great platform to learn data
science.

Sentiment (polarity=0.8, subjectivity=0.75)
```

We can see that polarity is **0.8**, which means that the statement is positive and **0.75** subjectivity refers that mostly it is a public opinion and not factual information.

Analysers in TextBlob: The textblob.sentiments module contains two sentiment analysis implementations, PatternAnalyzer (based on the pattern library) and NaiveBayesAnalyzer (an NLTK classifier trained on a movie reviews corpus).

In this project Naive Bayes Classifier is used.

3.2.4 Naive Bayes Classification

Naive Bayes is the simplest and fastest classification algorithm for a large chunk of data. In various applications such as spam filtering, text classification, sentiment analysis, and recommendation systems, Naive Bayes classifier is used successfully. It uses the Bayes probability theorem for unknown class prediction. The Naive Bayes classification technique is a simple and powerful classification task in machine learning. The use of Bayes’ theorem with a strong independence assumption between the features is the basis for naive Bayes classification. When used for textual data analysis, such as Natural Language Processing, the Naive Bayes classification yields good results. Simple Bayes or independent Bayes models are other names for nave Bayes models. All of these terms refer to the classifier’s decision rule using Bayes’ theorem. In practice, the Bayes theorem is applied by the Naive Bayes classifier. The power of Bayes' theorem is brought to machine learning with this classifier.

Naive Bayes Algorithm Intuition:

The Bayes theorem is used by the Naive Bayes Classifier to forecast membership probabilities for each class, such as the likelihood that a given record or data point belongs to that class. The most likely class is defined as the one having the highest probability. The Maximum A Posteriori is another name for this (MAP).

For a hypothesis with two occurrences A and B, the MAP is

MAP (A)

= max (P (A | B))

= max (P (B | A) * P (A))/P (B)

= max (P (B | A) * P (A)

P(B) stands for probability of evidence. It's utilized to make the outcome more normal. It has no effect on the outcome if it is removed. All of the features in the Naive Bayes Classifier are assumed to be unrelated. A feature's presence or absence has no bearing on the presence or absence of other features. We test a hypothesis given different evidence on features in real-world datasets. As a result, the computations become fairly difficult. To make things easier, the feature independence technique is utilized to decouple various pieces of evidence and consider them as separate entities. [8]

Applications of Naïve Bayes Algorithm:

Naïve Bayes is one of the most straightforward and fast classification algorithms. It is very well suited for large volumes of data. It is successfully used in various applications such as :

- Spam Filtering
- Text Classification

- Sentiment Analysis
- Recommender Systems

It uses the Bayes theorem of probability for the prediction of unknown classes. [8]

In this project we are analysing data in the form of tweets, these tweets are extracted from Twitter using the python library Tweepy.

3.2.5 Tweepy

Twitter is a popular social network where users share messages called tweets. Twitter allows users to mine the data of any user using Twitter API or Tweepy. The data will be tweets extracted from the user. The first thing to do is get the consumer key, consumer secret, access key and access secret from twitter developer available easily for each user. These keys will help the API for authentication.

Steps to Obtain Keys:

– Login to twitter developer section
– Go to “Create an App”
– Fill the details of the application.
– Click on Create your Twitter Application
– Details of your new app will be shown along with consumer key and consumer secret.
– For access token, click ” Create my access token”. The page will refresh and generate access token.

Tweepy is one of the libraries that should be installed using pip. Now in order to authorize our app to access Twitter on our behalf, we need to use the OAuth Interface from our developer’s account.

Tweepy provides the convenient Cursor interface to iterate through different types of objects. Twitter allows a maximum of 3200 tweets for extraction.

The following is a piece of code that helps communicate with the Twitter API through the set of tokens provided in the developer’s account’s dashboard:

```
import tweepy

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)

auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)

public_tweets = api.home_timeline()

for tweet in public_tweets:

print(tweet.text)
```

This example will download users home timeline tweets and print each one of their texts to the console. Twitter requires all requests to use OAuth for authentication. [11]

API

The API class provides access to the entire twitter RESTful API methods. Each method can accept various parameters and return responses.

Models

When we invoke an API method, most of the time what is returned back to us will be a Tweepy model class instance. This will contain the data returned from Twitter which we can then use inside our application. [11]

```
# Get the User object for twitter...

user = api.get_user(screen_name='twitter')
```

Models contain the data and some helper methods which we can then use:

```
print(user.screen_name)

print(user.followers_count)

for friend in user.friends():

 print(friend.screen_name)
```

3.2.6 Advantages of the Proposed System

Using Machine Learning:

The major advantage of Machine Learning is the ability to ‘train’ the algorithms. Using Natural Language Processing (NLP) alongside sentiment libraries, sentiment corpora and other human-annotated sentiment rules, algorithms are continuously enriched, becoming faster and more accurate.

The benefits of using Machine Learning for sentiment analysis are:

- **Speed:**

With Machine Learning you can automate the sentiment analysis of millions of product reviews without the need of involving human reviewers, thereby reducing the time and effort involved in collecting, storing and classifying customer reviews.

- **Accuracy:**

The use of Machine Learning algorithms, allows you to classify the sentiment of a customer review with precision and accuracy without the limitations of traditional sentiment analysis tools.

- **Agility:**

By identifying trends and changes in customer sentiment in near real-time, you can become more proactive and competitive, with more opportunities to increase your customers’ satisfaction, improve their experiences and prevent attrition.

Using Tweepy:

- Well written documentation, with a very active community
- Provides easy access to twitter API
- Provides many features about a given tweet (e.g., information about a tweet’s geographical location, etc.)

Using TextBlob:

- Can be interfaced with NLTK. It is similar to Python's string functions.
- It allows to easily swap to a pre-trained implementation from the NLTK library for sentiment analysis

- It is easy to learn and offers a lot of features like sentiment analysis, pos-tagging, noun phrase extraction, etc.

CHAPTER 4

SOFTWARE REQUIREMENT ANALYSIS

4.1 OVERALL DESCRIPTION

Sentiment analysis is a growing area of Natural Language Processing with research ranging from document level classification to learning the polarity of words and phrases. Given the character limitations on tweets, classifying the sentiment of Twitter messages is most similar to sentence level sentiment analysis however, the informal and specialized language used in tweets, as well as the very nature of the microblogging domain make Twitter sentiment analysis a very different task.

4.1.1 Hardware Requirements

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

The following are minimum hardware requirements required to perform the project:

• PROCESSOR	:	DUAL CORE 2 DUOS.
• RAM	:	4GB RAM
• HARD DISK	:	250 GB

4.1.2 Software Requirements:

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

The following are the minimum software requirements to perform the project:

• OPERATING SYSTEM	: Windows 7/8/10/11
• IDE	: Visual Studio code
• PROGRAMMING LANGUAGE	: Python 3.9.2
• FRONT-END	: Streamlit

4.1.3 Functional Requirements:

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behaviour. Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture. The system should be able to extract tweets. Then the system should be able to classify sentiments and lastly visualize result.

4.1.4 Non-Functional Requirements:

The major non-functional Requirements of the system are as follows:

- **Usability**

The system interface is designed to be simple and user friendly so that the user gets to the end result as fast as possible.

- **Reliability**

The system is more reliable because of the use of TextBlob which makes performing all NLP tasks easier.

- **Performance**

This system is developing in the high-level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

- **Supportability**

The system can run in various web browsers which support the system environment.

- **Implementation**

The system is implemented in web environment using Streamlit.

4.2 MODULES

Module 1:

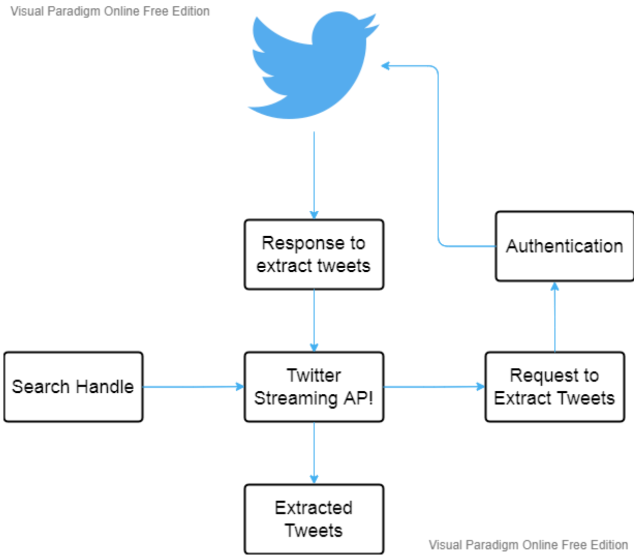

Visual Paradigm Online Free Edition
Response to extract tweets
Authentication
Search Handle
Twitter Streaming API
Request to Extract Tweets
Extracted Tweets
Visual Paradigm Online Free Edition

Fig.4.1: Twitter Interaction Module

The web application requests Twitter through the Twitter Streaming API to get access to tweets, Twitter then authenticates the APIKeys and sends a response to the application with desired tweets.

Module 2:

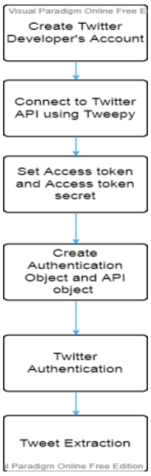

Visual Paradigm Online Free E
Create Twitter Developer's Account
Connect to Twitter API using Tweepy
Set Access token and Access token secret
Create Authentication Object and API object
Twitter Authentication
Tweet Extraction
Paradigm Online Free Edition

Fig.4.2: Authentication Module

Sign up for a Twitter Developer account to start exploring and building on the Twitter API v2 using essential access. Once getting access to the developer account, you will find or generate API Key and Secret, A set of user Access Tokens and Bearer token which will be embedded in the project. Through these Tokens the Twitter API allows you to perform various actions such as searching/extracting tweets using code to be performed on the web application.

Module 3:

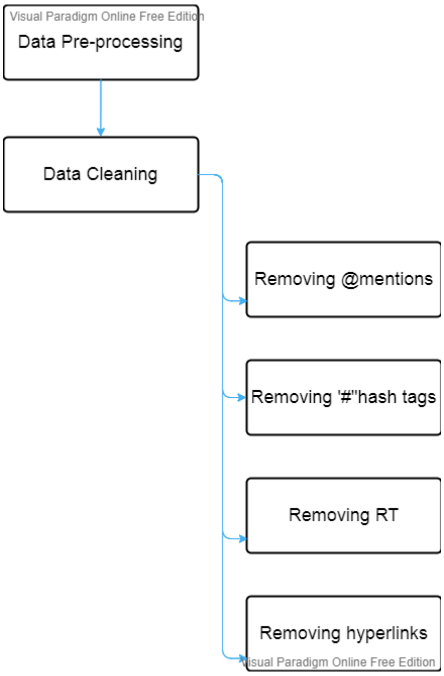


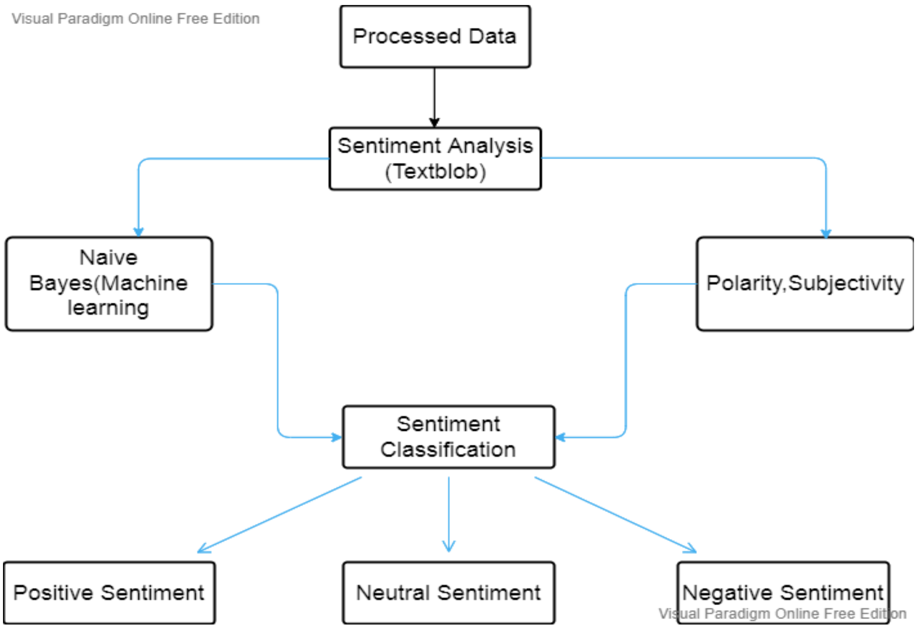
Fig.4.3: Data Pre-Processing Module

Data preprocessing can refer to manipulation or dropping of data before it is used in order to ensure or enhance performance, and is an important step in the data mining process.

It takes raw data and transforms it into a format that can be understood and analysed by computers and machine learning. In our project the data processing steps involved are:

- Removing @mentions
- Removing hashtags '#'
- Removing ReTweets
- Removing hyperlinks
- Performing tokenization and lemmatization

Module 4:

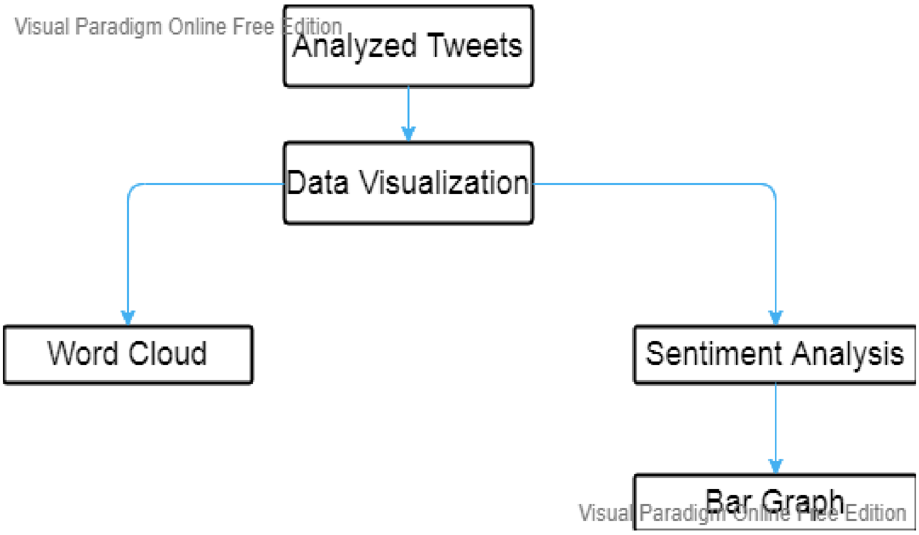


Visual Paradigm Online Free Edition
Processed Data
Sentiment Analysis (Textblob)
Naive Bayes(Machine learning
Polarity,Subjectivity
Sentiment Classification
Positive Sentiment
Neutral Sentiment
Negative Sentiment
Visual Paradigm Online Free Edition

Fig.4.4: Sentiment Analysis Module

The pre-processed tweets are taken in for analysis. Textblob consists of an in-built trained naive bayes model of a movie review dataset through which it tests the pre-processed tweets and obtains their polarity and subjectivity. Using the polarity of the tweet's sentiment classification is performed and the tweet sentiments are visualized in terms of Positive, Negative and Neutral.

Module 5:



Visual Paradigm Online Free Edition
Analyzed Tweets
Data Visualization
Word Cloud
Sentiment Analysis
Bar Graph
Visual Paradigm Online Free Edition

Fig.4.5: Visualization Module

The analysed tweets are then used to generate a word cloud which gives a visual representation of words that have greater prominence to words that appear frequently. These tweets are also used to visualize sentiments in terms of positive negative and neutral in the form of a bar graph based on their polarity.

4.3 SYSTEM ARCHITECTURE

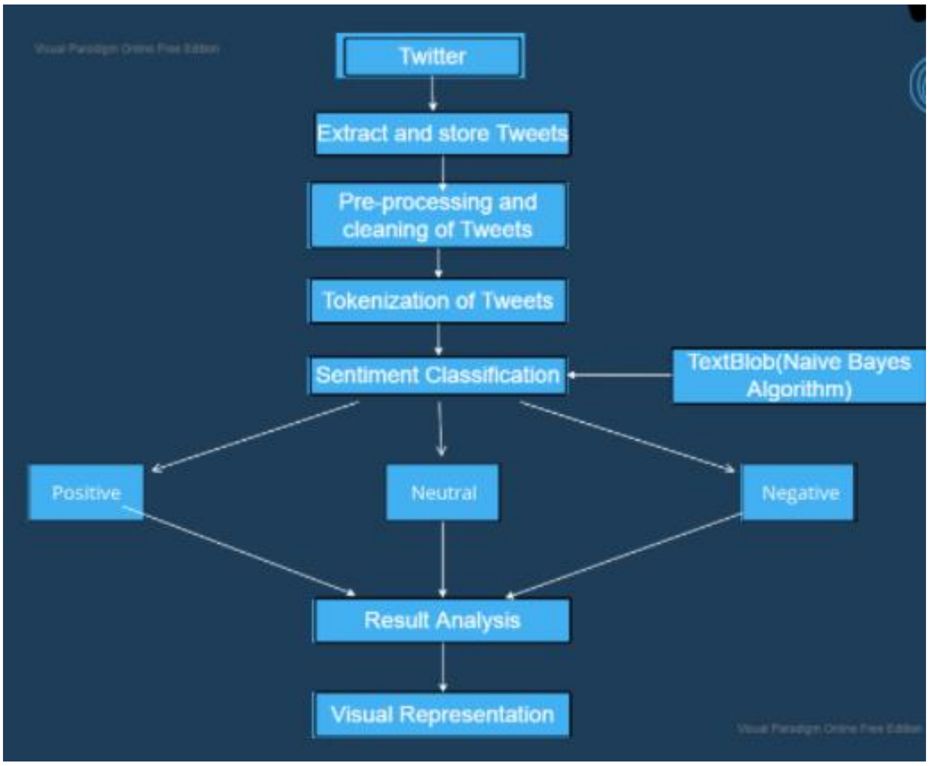

Twitter
Extract and store Tweets
Pre-processing and cleaning of Tweets
Tokenization of Tweets
Sentiment Classification
TextBlob(Naive Bayes Algorithm)
Positive
Neutral
Negative
Result Analysis
Visual Representation
Visual Paradigm Online Free Edition
Visual Paradigm Online Free Edition

Fig.4.6: System Architecture

- Authenticate Tweepy tool using the Developer Account
- Request for tweets, extract and send them for Pre-processing, Tokenization and lemmatization.
- Perform sentiment classification using the TextBlob tool.
- Analyze the classified tweets based on polarity.
- Visual representation of sentiment analysis is obtained

CHAPTER 5

DESIGN

5.1 DATAFLOW DIAGRAM

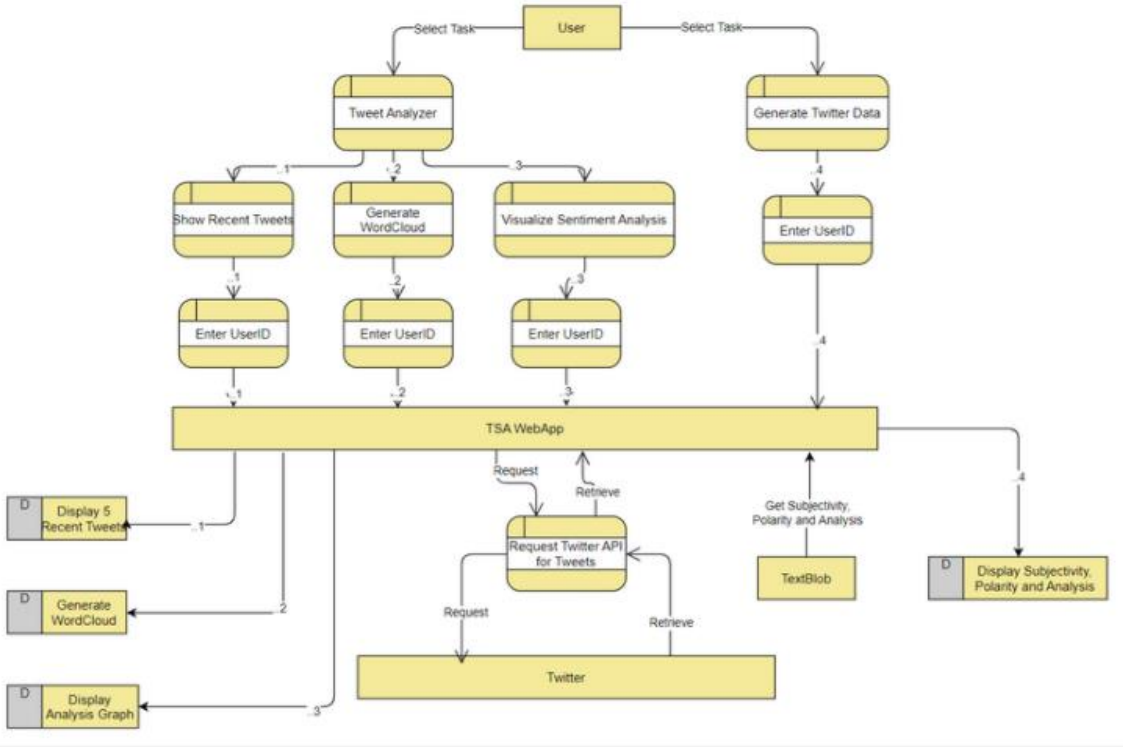


Fig.5.1: DFD

A data flow diagram (DFD) maps out the flow of information for any process or system. They can be used to analyze an existing system or model a new one.

Each process should have at least one input and an output. Each data store should have at least one data flow in and one data flow out. Data stored in a system must go through a process. All processes in a DFD go to another process or a data store.

5.2 UML DIAGRAMS:

A Unified Modelling Language (UML) diagram provides a visual representation of an aspect of a system. UML diagrams illustrate the quantifiable aspects of a system that can be described visually, such as relationships, behaviour, structure, and functionality. UML diagrams can help system architects and developers understand, collaborate on, and develop an application. High-level architects and managers can use UML diagrams to visualize an entire system or project and separate applications into smaller components for development.

System developers can use UML diagrams to specify, visualize, and document applications, which can increase efficiency and improve their application design. UML diagrams can also help identify patterns of behaviour, which can provide opportunities for reuse and streamlined applications.

There are two broad categories of diagrams and they are again divided into subcategories −Structural Diagrams, Behavioural Diagrams.

- Structural Diagrams:

These diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram, which forms the main structure and are therefore stable.

Example: Class Diagram, Component Diagram, Deployment Diagram

- Behavioural Diagrams:

These diagrams basically capture the dynamic aspect of a system. Dynamic aspect can be further described as the changing/moving parts of a system.

Example: Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram

5.2.1 Sequence Diagram:

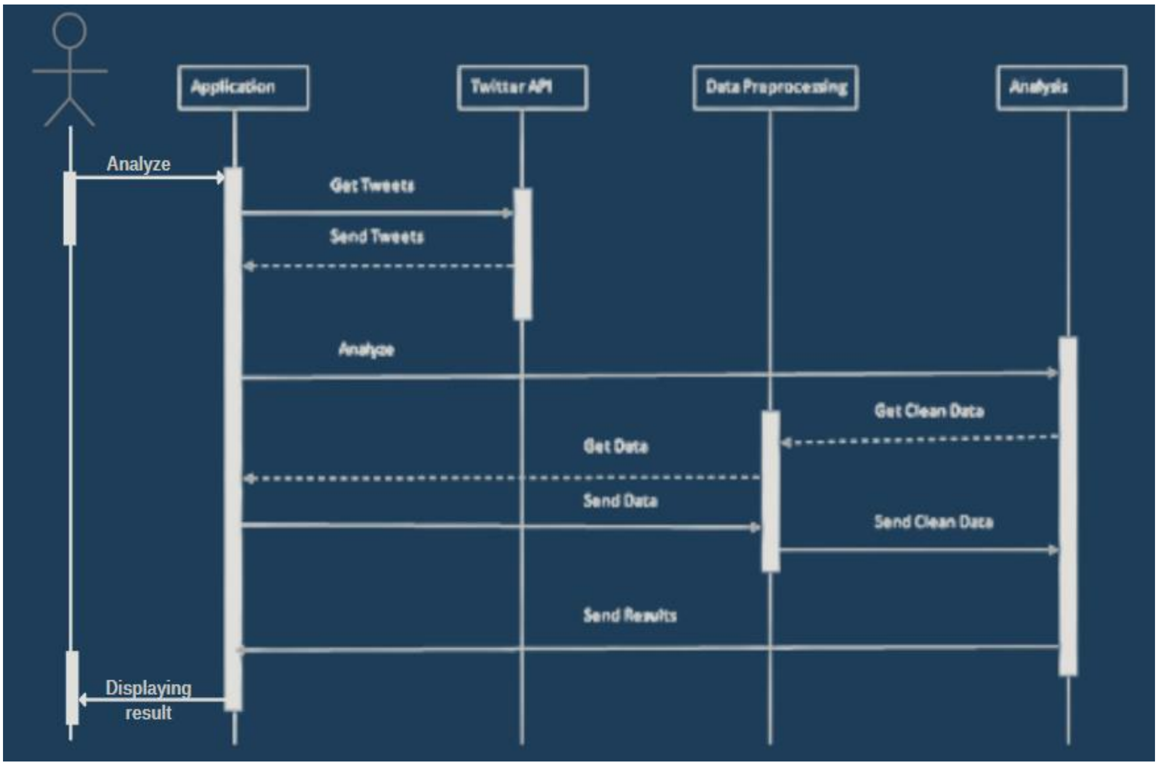


Fig.5.2: Sequence Diagram

A sequence diagram or system sequence diagram shows object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario.

When a user requests the application to analyze a twitter handle, the application sends a request to Twitter through the Twitter API, which in turn sends a response to the application with the desired tweets. The application then performs data preprocessing on the tweets extracted and obtains clean data, the clean data is then visualized based on polarity of sentiments in terms of positive negative and neutral in the form of a bar graph. The bar graph is then displayed as result on the application and the user can then view it.

5.2.2 Use-case Diagram

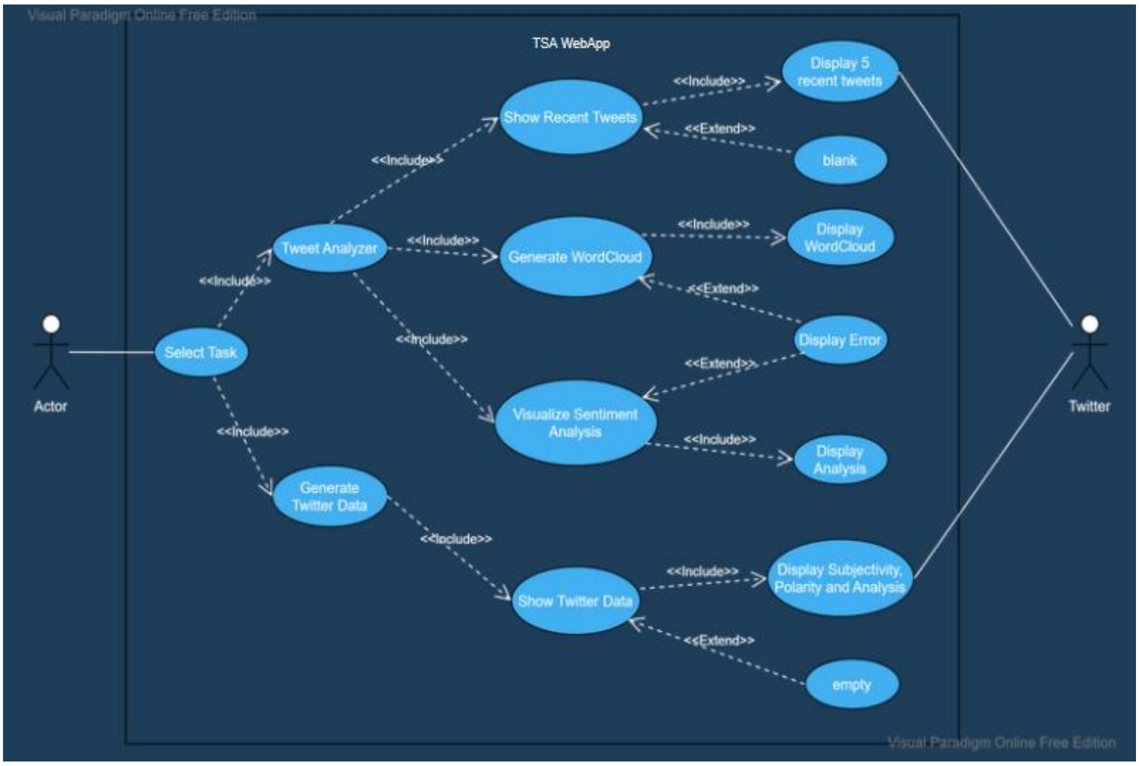


Fig.5.3: Use-Case Diagram

In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.

The user requests the authentication service to retrieve twitter data. On retrieving the tweets, the user performs various activities such as "Show Recent Tweets”: the application fetches 5 recent tweets from the twitter handle, "GenerateWordCloud": application displays a wordcloud which gives a visual representation of words that have greater prominence to words that appear frequently, "Sentiment Analysis" Analysis’ which provides a bar graph that tells us how positively or negatively the tweeter has been tweeting, and "Generate Twitter Data" which fetches last 100 tweets to provide a dataframe that obtains the subjectivity-polarity-analysis of the tweets

5.2.3 Deployment Diagram:

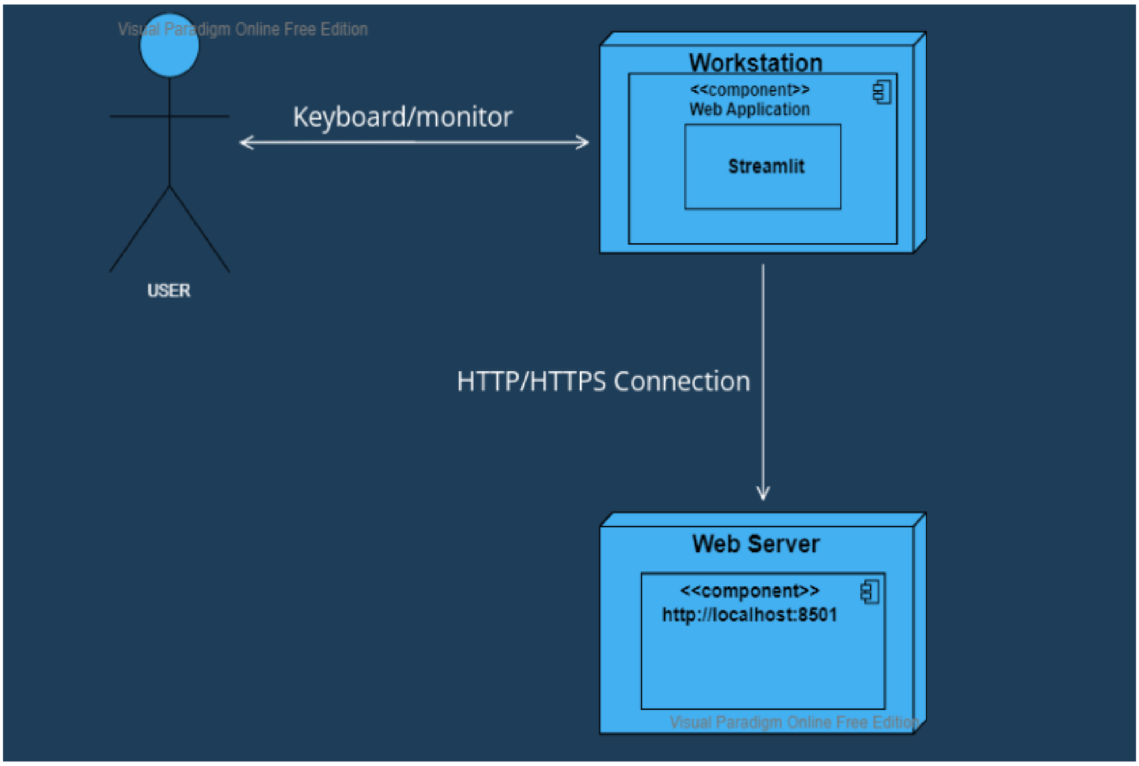


Fig.5.4: Deployment Diagram

A deployment diagram in the Unified Modelling Language models the physical deployment of artifacts on nodes. It describes what hardware components exist, what software components run on each node and how different nodes are connected. These nodes are physical entities where the components are deployed.

Deployment diagrams are used for visualizing the deployment view of a system. This is generally used by the deployment team. The user interacts with the web application using his PC components which opens up a localhost through http/https connection on which he can perform all his tasks.

5.2.4 Class Diagram

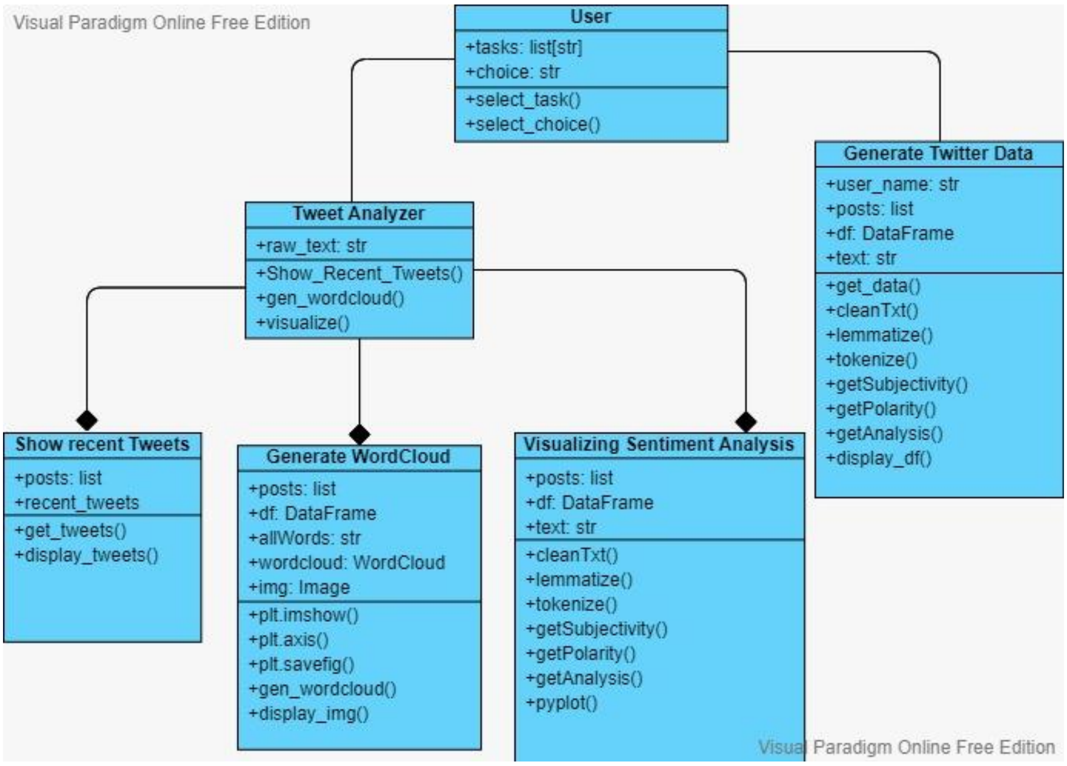


Fig.5.5: Class Diagram

Class diagrams are the most common diagrams used in UML. Since classes are the building block of objects, class diagrams are the building blocks of UML. The various components in a class diagram can represent the classes that will actually be programmed, the main objects, or the interactions between classes and objects. The class shape itself consists of a rectangle with three rows. The top row contains the name of the class, the middle row contains the attributes of the class, and the bottom section expresses the methods or operations that the class may use. Classes and subclasses are grouped together to show the static relationship between each object. Class diagram represents the object orientation of a system. Hence, it is generally used for development purpose.

5.2.5 Activity Diagram:

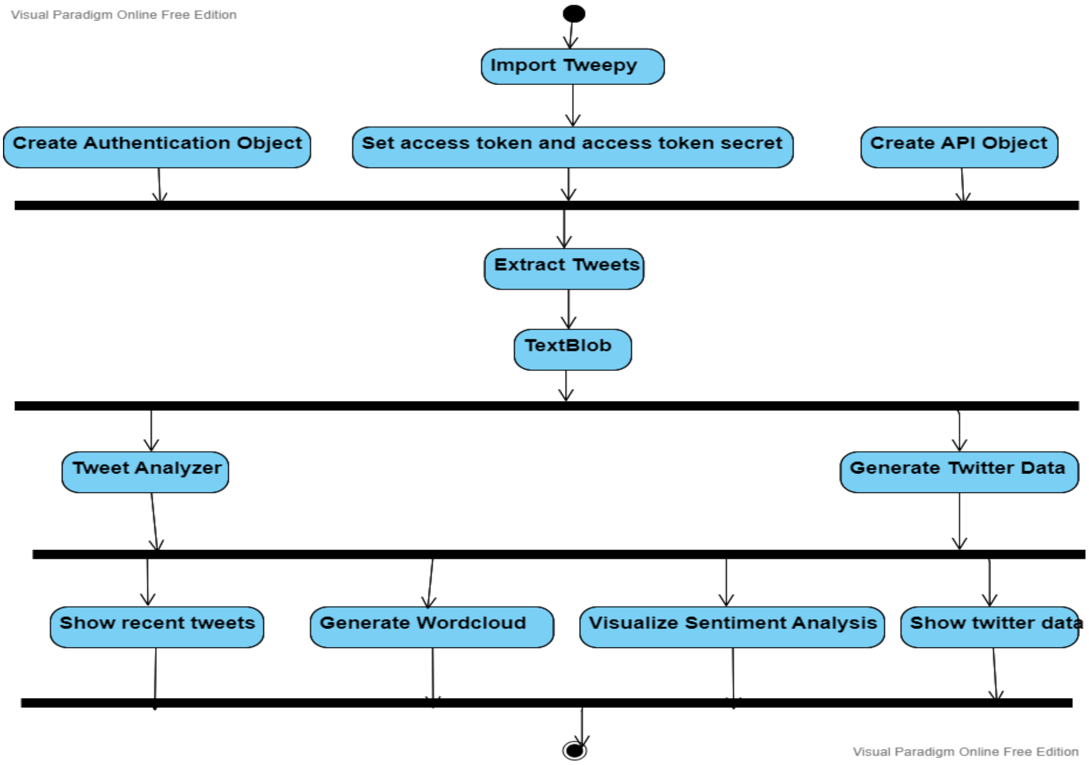

Visual Paradigm Online Free Edition
Import Tweepy
Create Authentication Object
Set access token and access token secret
Create API Object
Extract Tweets
TextBlob
Tweet Analyzer
Generate Twitter Data
Show recent tweets
Generate Wordcloud
Visualize Sentiment Analysis
Show twitter data
Visual Paradigm Online Free Edition

Fig.5.6: Activity Diagram

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. They can depict both sequential processing and concurrent processing of activities using an activity diagram.

Here, we start by importing tweepy and other necessary modules to authenticate our developer's twitter account and generate a set of tokens which are required to extract tweets from a particular user's twitter account The tweets are cleaned and then the following operation are performed on them:

- Tweet Analyzer-: show recent tweets, generate Wordcloud, visualize sentiment analysis
- Generate twitter data: show twitter data

5.2.6 State Machine Diagram

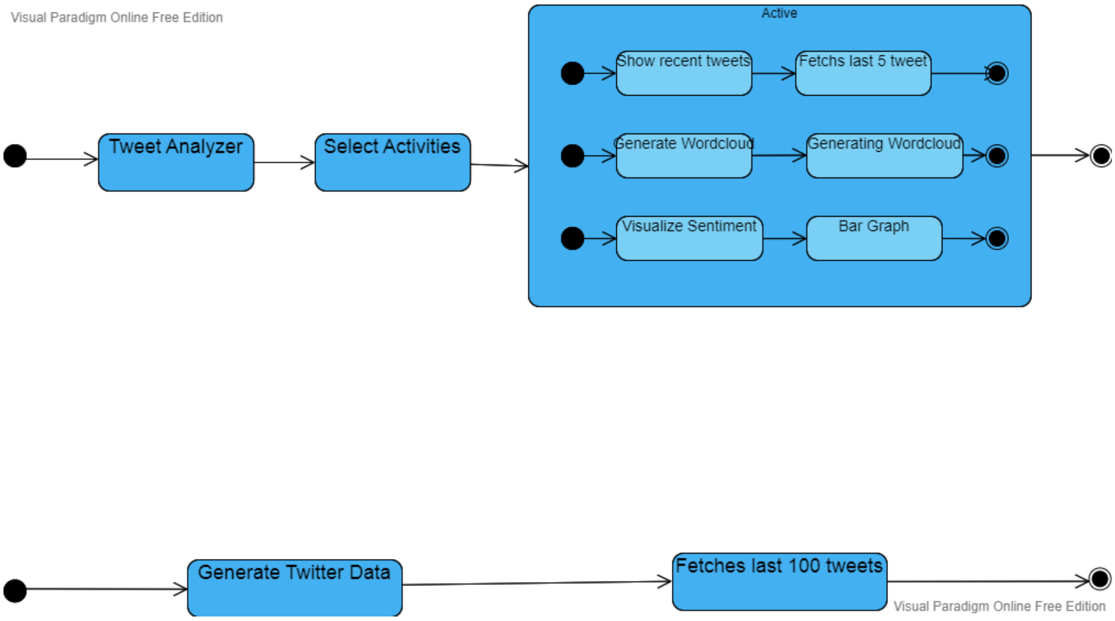


Fig.5.7: State Machine Diagram

A state machine is any device that stores the status of an object at a given time and can change status or cause other actions based on the input it receives. States refer to the different combinations of information that an object can hold, not how the object behaves. In order to understand the different states of an object, you might want to visualize all of the possible states and show how an object gets to each state, and you can do so with a UML state diagram. Each state diagram typically begins with a dark circle that indicates the initial state and ends with a bordered circle that denotes the final state. However, despite having clear start and end points, state diagrams are not necessarily the best tool for capturing an overall progression of events. Rather, they illustrate specific kinds of behaviour—in particular, shifts from one state to another. State diagrams mainly depict states and transitions. States are represented with rectangles with rounded corners that are labeled with the name of the state. Transitions are marked with arrows that flow from one state to another, showing how the states change.

CHAPTER 6

IMPLEMENTATION

6.1 PREREQUISITES

To implement the source code the following essentials are required:

- Familiarity with the English language
- Basic computer literacy
- Access to internet at a minimum of 5Mbps

6.2 CODE

```
import streamlit as st
import tweepy
from textblob import TextBlob
from textblob import Word
import pandas as pd
from wordcloud import WordCloud
import re
import matplotlib.pyplot as plt
from PIL import Image
import seaborn as sns

tsafav = Image.open("tsaf.ico")
st.set_page_config(
 page_title="Twitter Sentiment Analysis",
 page_icon=tsafav,
 layout = "wide")

#--------------------------------------------------------
consumer_key = 'e8P53GlkhDJNtc8d9w1qb5oOb'
consumer_secret =
'fysNKDmd4k46cczfryOu5chh1nBCMSTsORHSVNAv4qJigvvnVF'
```

```
access_token = '1451182626165825538-
YfvuMGo1whpq1dm7xxG1bRHM1ADQgA'
access_token_secret =
'g8MbdXpexp484RTar317BM4EZCCXER9Zzw7c7HyS5XE0l'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)
#--------------------------------------------------------

st.markdown(
 """
 <style>
 .main{
 background-color: #1d9bf0;
 }
 .sidebar .sidebar-content {
 background-color: white;
 }
 </style>
 """,
 unsafe_allow_html=True
)

def main():
 st.title("Twitter Sentiment Analysis")
 tasks=["Tweet Analyzer","Generate Twitter Data"]
 choice = st.sidebar.selectbox("Select Your Task", tasks)
 if choice=="Tweet Analyzer":
  raw_text = st.text_area("Enter the exact twitter handle
of the Personality (without @)")
```

```
  Analyzer_choice = st.selectbox("Select the Activities",
 ["Show Recent Tweets","Generate WordCloud" ,"Visualize
the Sentiment Analysis"])
  if st.button("Analyze"):
   if Analyzer_choice == "Show Recent Tweets":
    st.success("Fetching last 5 Tweets")
    def Show_Recent_Tweets(raw_text):
     # Extract 100 tweets from the twitter user
     posts = api.user_timeline(screen_name=raw_text,
count = 100, lang ="en", tweet_mode="extended")
     def get_tweets():
      l=[]
      i=1
      for tweet in posts[:5]:
       l.append(tweet.full_text)
       i= i+1
      return l
     recent_tweets=get_tweets()
     return recent_tweets
    recent_tweets = Show_Recent_Tweets(raw_text)
    st.write(recent_tweets)
   elif Analyzer_choice=="Generate WordCloud":
    st.success("Generating Word Cloud")
    def gen_wordcloud():
     posts = api.user_timeline(screen_name=raw_text,
count = 100, lang ="en", tweet_mode="extended")
     # Create a dataframe with a column called Tweets
     df = pd.DataFrame([tweet.full_text for tweet in
posts], columns=['Tweets'])
     # word cloud visualization
     allWords = ' '.join([twts for twts in df['Tweets']])
     wordCloud = WordCloud(width=500, height=300,
random_state=21, max_font_size=110).generate(allWords)
     plt.imshow(wordCloud, interpolation="bilinear")
     plt.axis('off')
     plt.savefig('WC.jpg')
     img= Image.open("WC.jpg")
     return img
    img=gen_wordcloud()
    st.image(img)
   else:
    def Plot_Analysis():
     st.success("Generating Visualisation for Sentiment
Analysis")
     posts = api.user_timeline(screen_name=raw_text,
count = 100, lang ="en", tweet_mode="extended")
     df = pd.DataFrame([tweet.full_text for tweet in
posts], columns=['Tweets'])
     # Create a function to clean the tweets
     def cleanTxt(text):
      text = re.sub('@[A-Za-z0-9]+', '', text) #Removing
@mentions
      text = re.sub('#', '', text) # Removing '#' hash
tag
      text = re.sub('RT[\s]+', '', text) # Removing RT
      text = re.sub('https?:\/\/\S+', '', text) #
Removing hyperlink
      text = Word(text)
      text.lemmatize() #lemmatization
      return text
     # Clean the tweets
     df['Tweets'] = df['Tweets'].apply(cleanTxt)
     # Create a function to get the subjectivity
     def getSubjectivity(text):
      return TextBlob(text).sentiment.subjectivity
     # Create a function to get the polarity
     def getPolarity(text):
      return  TextBlob(text).sentiment.polarity
     # Create two new columns 'Subjectivity' & 'Polarity'
     df['Subjectivity'] =
df['Tweets'].apply(getSubjectivity)
     df['Polarity'] = df['Tweets'].apply(getPolarity)
     def getAnalysis(score):
      if score < 0:
       return 'Negative'
      elif score == 0:
       return 'Neutral'
      else:
       return 'Positive'
     df['Analysis'] = df['Polarity'].apply(getAnalysis)
     return df
    df= Plot_Analysis()
    st.write(sns.countplot(x=df["Analysis"],data=df))
    st.set_option('deprecation.showPyplotGlobalUse',
False)
    st.pyplot()
 else:
  user_name = st.text_area("*Enter the exact twitter
handle of the Personality (without @)*")
  def get_data(user_name):
   posts = api.user_timeline(screen_name=user_name, count
= 100, lang ="en", tweet_mode="extended")
   df = pd.DataFrame([tweet.full_text for tweet in
posts], columns=['Tweets'])
   def cleanTxt(text):
    text = re.sub('@[A-Za-z0-9]+', '', text) #Removing
@mentions
    text = re.sub('#', '', text) # Removing '#' hash tag
    text = re.sub('RT[\s]+', '', text) # Removing RT
    text = re.sub('https?:\/\/\S+', '', text) # Removing
hyperlink
    text = Word(text)
    text.lemmatize() #lemmatization
    return text
   # Clean the tweets
   df['Tweets'] = df['Tweets'].apply(cleanTxt)
   # Create a function to get subjectivity
   def getSubjectivity(text):
    return TextBlob(text).sentiment.subjectivity
   # Create a function to get the polarity
   def getPolarity(text):
    return TextBlob(text).sentiment.polarity
   # Create two new columns 'Subjectivity' & 'Polarity'
   df['Subjectivity'] =
df['Tweets'].apply(getSubjectivity)
   df['Polarity'] = df['Tweets'].apply(getPolarity)
   def getAnalysis(score):
    if score < 0:
     return 'Negative'
    elif score == 0:
     return 'Neutral'
    else:
     return 'Positive'
   df['Analysis'] = df['Polarity'].apply(getAnalysis)
   return df
  if st.button("Show Data"):
   st.success("Fetching Last 100 Tweets")
   df=get_data(user_name)
   st.write(df)
 st.caption('A collaborative work of UMAIMA SIDDIQUA and
SYEDA FATIMA ALI')
if __name__ == "__main__":
    main()
```

CHAPTER 7

TESTING

7.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

7.3 TYPES OF TESTS

7.3.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.3.2 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

7.3.3 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.3.4 Performance Test

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

7.3.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

7.3.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updating process

7.3.7 Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

To perform these tests Selenium IDE was used.

7.3.7.1 Selenium

Selenium is an open-source umbrella project for a range of tools and libraries aimed at supporting web browser automation.[3] Selenium provides a playback tool for authoring functional tests without the need to learn a test scripting language (Selenium IDE). It also provides a test domain-specific language (Selenese) to write tests in a number of popular programming languages, including JavaScript (Node.js), C#, Groovy, Java, Perl, PHP, Python, Ruby and Scala. The tests can then run against most modern web browsers. Selenium runs on Windows, Linux, and macOS. It is open-source software released under the Apache License 2.0.

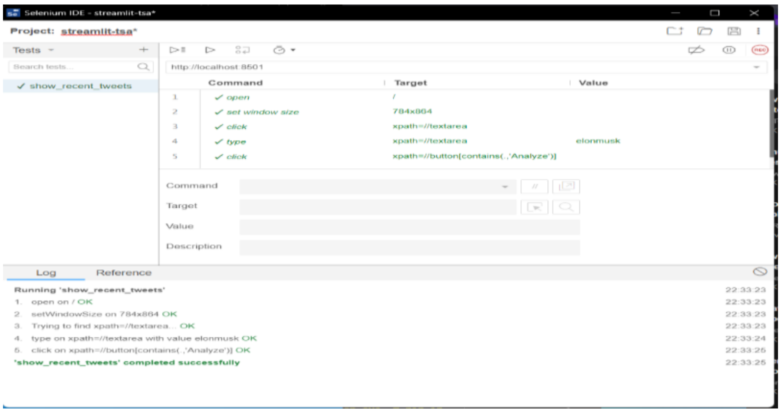
7.3.7.2 Selenium IDE

The Selenium IDE is a browser extension that records and plays back a user's actions. Selenium's Integrated Development Environment (Selenium IDE) is an easy-to-use browser extension that records a user's actions in the browser using existing Selenium commands, with parameters defined by the context of each element. It provides an excellent way to learn Selenium syntax. [16] It's available for Google Chrome, Mozilla Firefox, and Microsoft Edge

So, Test cases were created and run using Selenium IDE

7.3.7.2.1 Test Cases

- Running 1st case: show_recent_tweets

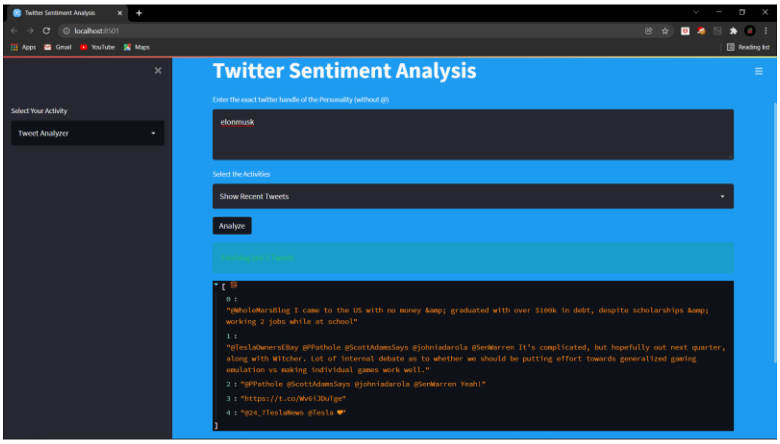


	Command	Target	Value
1	✓ open	/	
2	✓ set window size	784x864	
3	✓ click	xpath=//textarea	
4	✓ type	xpath=//textarea	elonmusk
5	✓ click	xpath=//button[contains(.,'Analyze')]	

```
Running 'show_recent_tweets'                                          22:33:23
1.  open on / OK                                                      22:33:23
2.  setWindowSize on 784x864 OK                                       22:33:23
3.  Trying to find xpath=//textarea... OK                             22:33:23
4.  type on xpath=//textarea with value elonmusk OK                   22:33:24
5.  click on xpath=//button[contains(.,'Analyze')] OK                 22:33:25
'show_recent_tweets' completed successfully                           22:33:25
```

Fig.7.1: show_recent_tweets Test Case

Running this test case opens the following window:



Twitter Sentiment Analysis

Select Your Activity: Tweet Analyzer

Enter the exact twitter handle of the Personality (without @): elonmusk

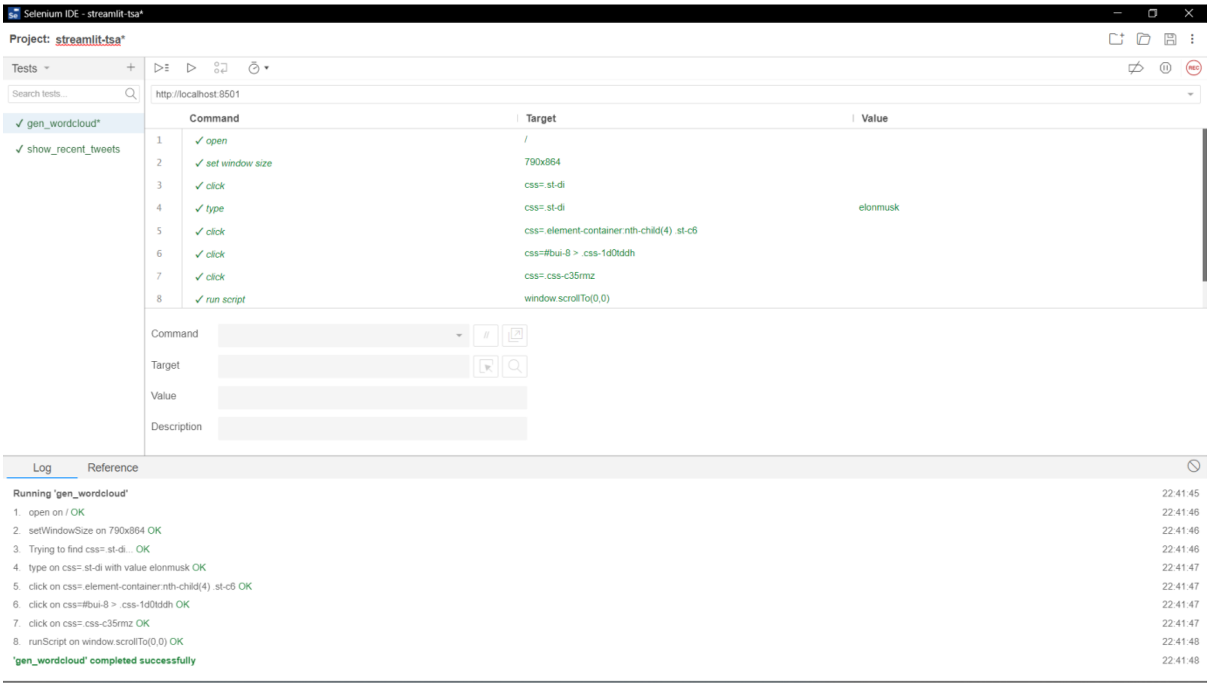
Select the Activities: Show Recent Tweets

Fetching last 5 Tweets

```
[
  0 :
  "@WholeMarsBlog I came to the US with no money &amp; graduated with over $100k in debt, despite scholarships &amp;
  working 2 jobs while at school"
  1 :
  "@TeslaOwnersEBay @PPathole @ScottAdamsSays @johniadarola @SenWarren It's complicated, but hopefully out next quarter,
  along with Witcher. Lot of internal debate as to whether we should be putting effort towards generalized gaming
  emulation vs making individual games work well."
  2 : "@PPathole @ScottAdamsSays @johniadarola @SenWarren Yeah!"
  3 : "https://t.co/Wv6iJDuTge"
  4 : "@24_7TeslaNews @Tesla ❤"
]
```

Fig.7.2: Window 1

- Running 2nd test case: gen_wordcloud



Selenium IDE - streamlit-tsa*

Project: streamlit-tsa*

Tests

✓ gen_wordcloud*

✓ show_recent_tweets

http://localhost:8501

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	790x864	
3	✓ click	css=.st-di	
4	✓ type	css=.st-di	elonmusk
5	✓ click	css=.element-container:nth-child(4) .st-c6	
6	✓ click	css=#bui-8 > .css-1d0tddh	
7	✓ click	css=.css-c35rmz	
8	✓ run script	window.scrollTo(0,0)	

Command

Target

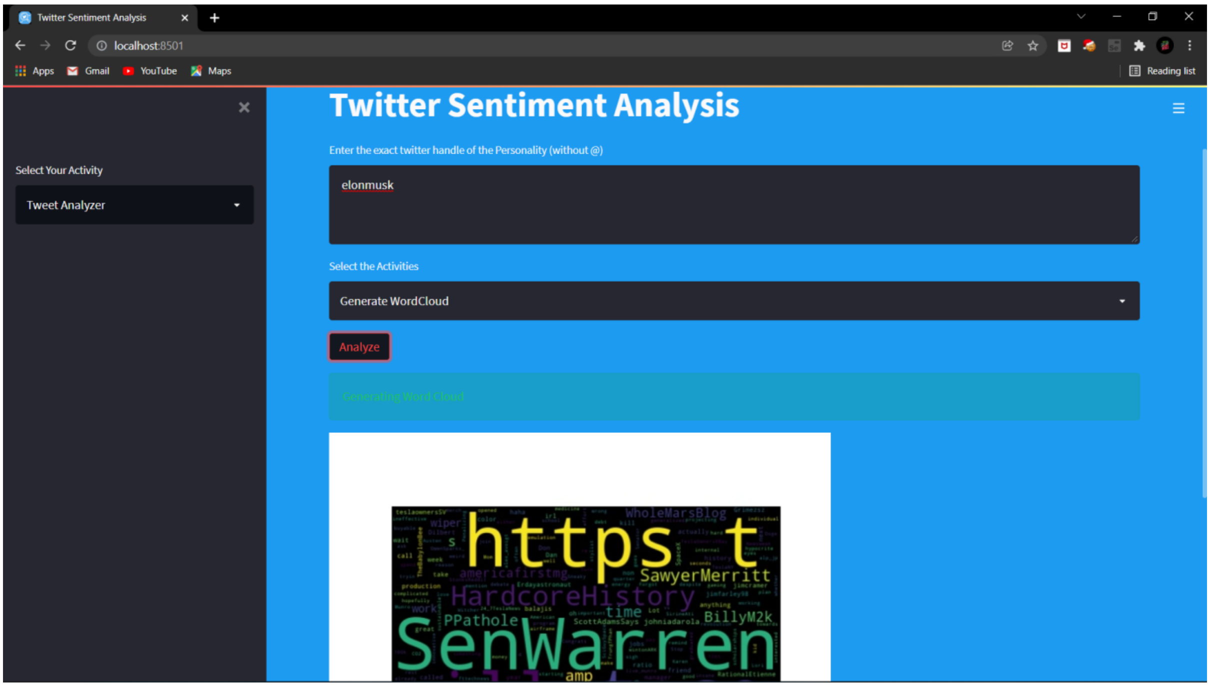
Value

Description

```
Running 'gen_wordcloud'                                                  22:41:45
1.  open on / OK                                                         22:41:46
2.  setWindowSize on 790x864 OK                                          22:41:46
3.  Trying to find css=.st-di... OK                                      22:41:46
4.  type on css=.st-di with value elonmusk OK                            22:41:47
5.  click on css=.element-container:nth-child(4) .st-c6 OK               22:41:47
6.  click on css=#bui-8 > .css-1d0tddh OK                                22:41:47
7.  click on css=.css-c35rmz OK                                          22:41:47
8.  runScript on window.scrollTo(0,0) OK                                 22:41:48
'gen_wordcloud' completed successfully                                   22:41:48
```

Fig.7.3: gen_wordcloud Test Case

Running this test case it opens the following browser:



Select Your Activity: Tweet Analyzer

Twitter Sentiment Analysis

Enter the exact twitter handle of the Personality (without @)

elonmusk

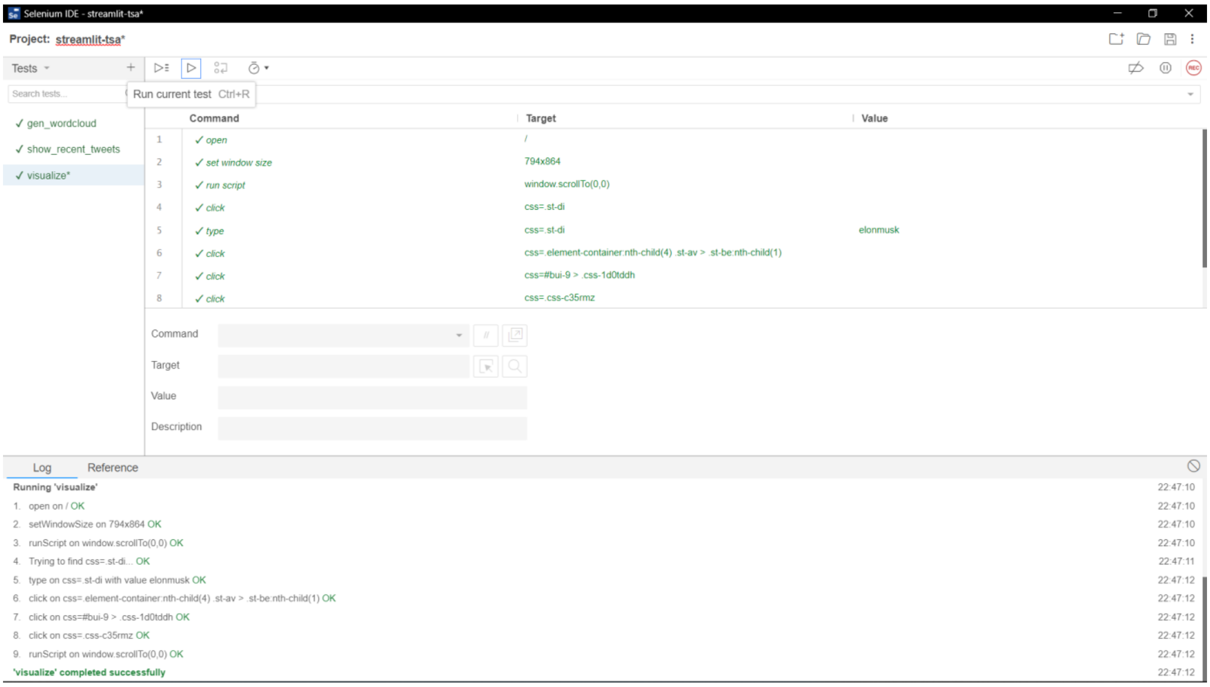
Select the Activities: Generate WordCloud

Analyze

Generating Word Cloud

Fig.7.4: Window 2

- <u>Running the 3rd test case:</u> visualize



Project: streamlit-tsa*

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	794x864	
3	✓ run script	window.scrollTo(0,0)	
4	✓ click	css=.st-di	
5	✓ type	css=.st-di	elonmusk
6	✓ click	css=.element-container:nth-child(4) .st-av > .st-be:nth-child(1)	
7	✓ click	css=#bui-9 > .css-1d0tddh	
8	✓ click	css=.css-c35rmz	

Command

Target

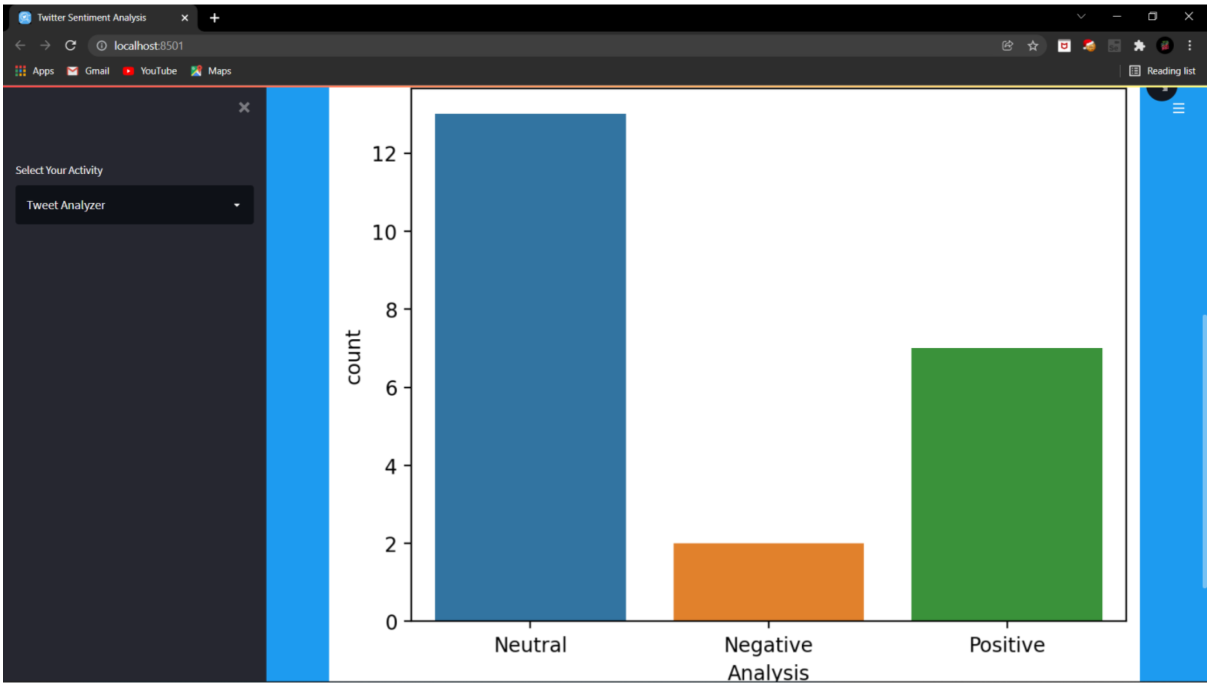
Value

Description

```
Running 'visualize'                                                                  22:47:10
1.  open on / OK                                                                     22:47:10
2.  setWindowSize on 794x864 OK                                                      22:47:10
3.  runScript on window.scrollTo(0,0) OK                                             22:47:10
4.  Trying to find css=.st-di... OK                                                  22:47:11
5.  type on css=.st-di with value elonmusk OK                                        22:47:12
6.  click on css=.element-container:nth-child(4) .st-av > .st-be:nth-child(1) OK     22:47:12
7.  click on css=#bui-9 > .css-1d0tddh OK                                            22:47:12
8.  click on css=.css-c35rmz OK                                                      22:47:12
9.  runScript on window.scrollTo(0,0) OK                                             22:47:12
'visualize' completed successfully                                                   22:47:12
```

Fig.7.5: visualize Test Case

Running this test, following window is opened:

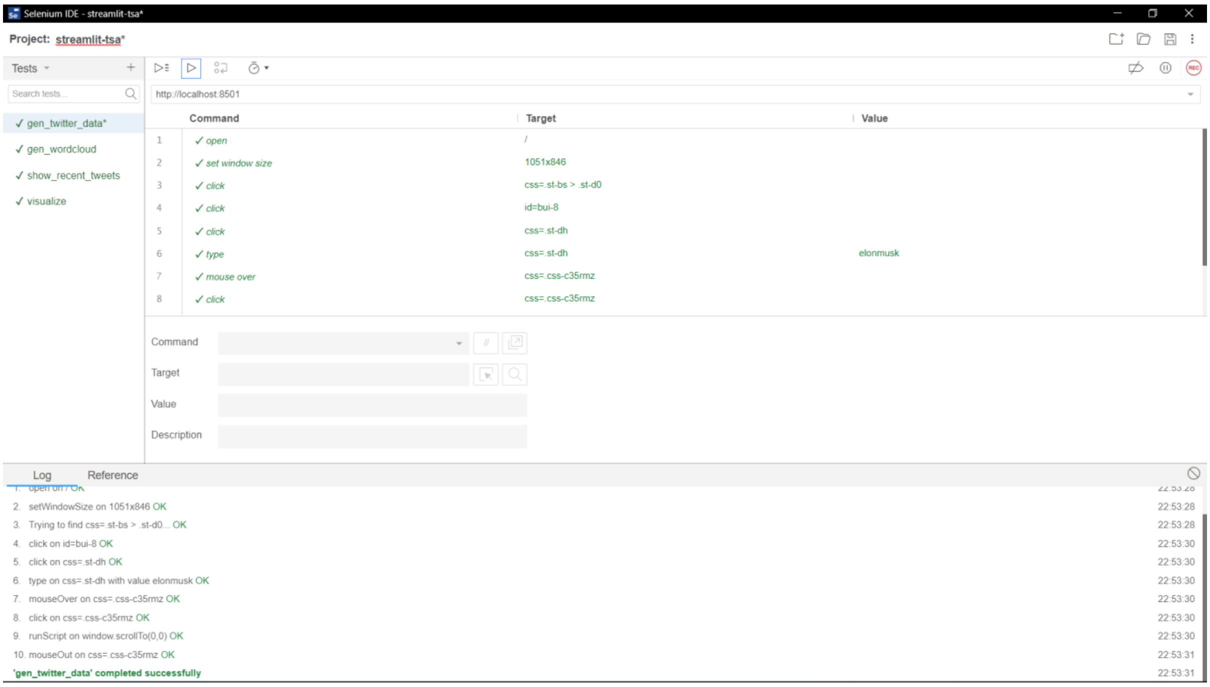


Select Your Activity

Tweet Analyzer

Fig:7.6: Window 3

- Running the 4th test case: generate_twitter_data

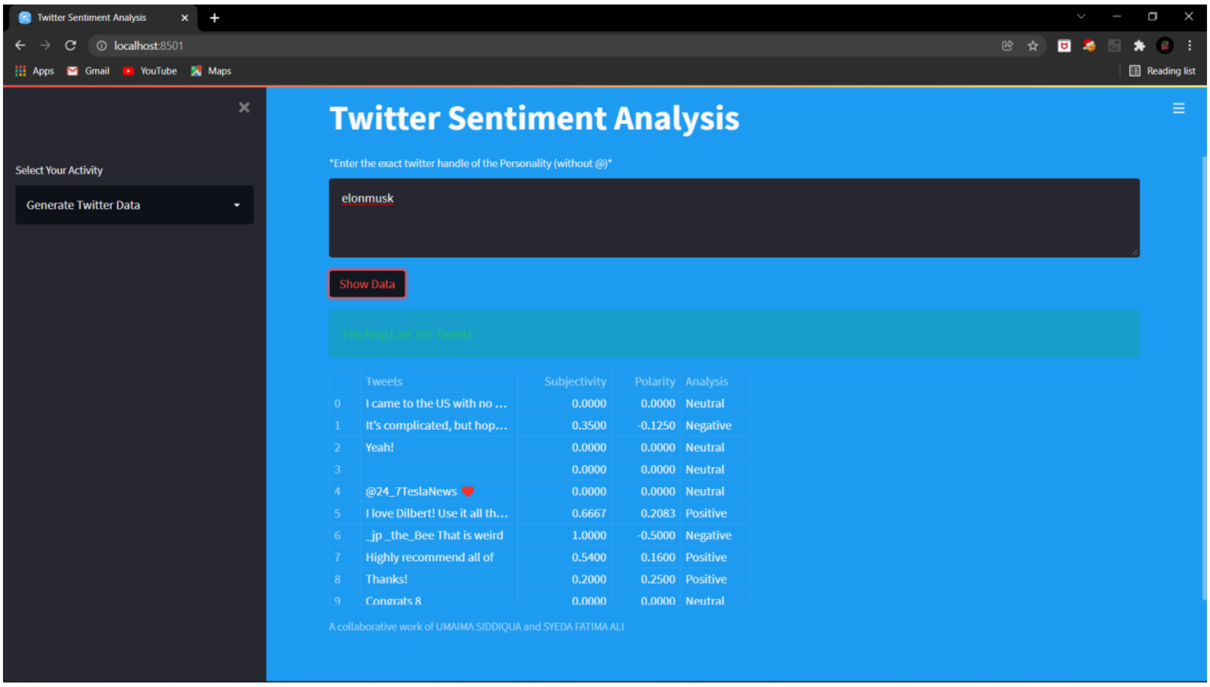


	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1051x846	
3	✓ click	css=.st-bs > .st-d0	
4	✓ click	id=bui-8	
5	✓ click	css=.st-dh	
6	✓ type	css=.st-dh	elonmusk
7	✓ mouse over	css=.css-c35rmz	
8	✓ click	css=.css-c35rmz	

```
1.  open on / OK                                              22:53:28
2.  setWindowSize on 1051x846 OK                              22:53:28
3.  Trying to find css=.st-bs > .st-d0... OK                  22:53:28
4.  click on id=bui-8 OK                                      22:53:30
5.  click on css=.st-dh OK                                    22:53:30
6.  type on css=.st-dh with value elonmusk OK                 22:53:30
7.  mouseOver on css=.css-c35rmz OK                           22:53:30
8.  click on css=.css-c35rmz OK                               22:53:30
9.  runScript on window.scrollTo(0,0) OK                      22:53:30
10. mouseOut on css=.css-c35rmz OK                            22:53:31
'gen_twitter_data' completed successfully                     22:53:31
```

Fig.7.7: generate_twitter_data Test Case

On running this test, the following browser is opened:



Twitter Sentiment Analysis

Enter the exact twitter handle of the Personality (without @)

elonmusk

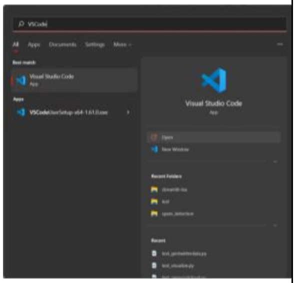
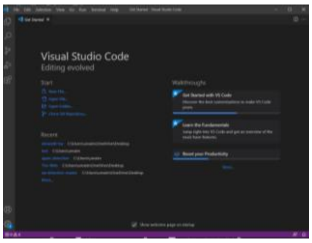
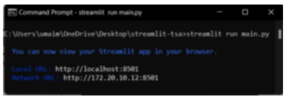
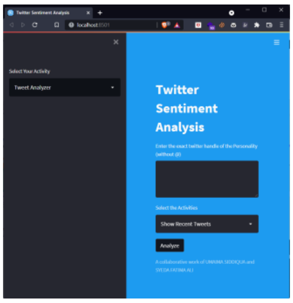
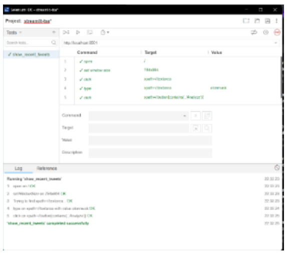
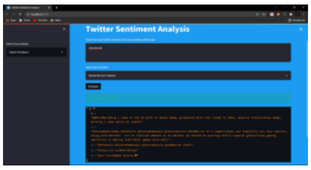
Fetching Last 100 Tweets

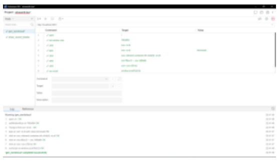
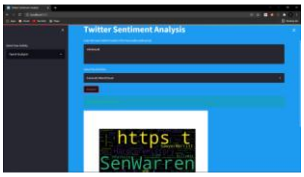
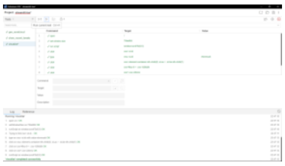
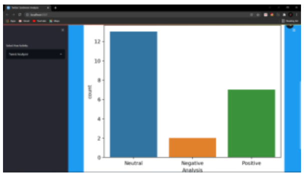
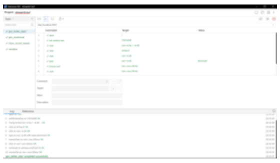
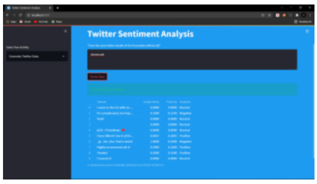
	Tweets	Subjectivity	Polarity	Analysis
0	I came to the US with no …	0.0000	0.0000	Neutral
1	It's complicated, but hop…	0.3500	-0.1250	Negative
2	Yeah!	0.0000	0.0000	Neutral
3		0.0000	0.0000	Neutral
4	@24_7TeslaNews ❤	0.0000	0.0000	Neutral
5	I love Dilbert! Use it all th…	0.6667	0.2083	Positive
6	_jp _the_Bee That is weird	1.0000	-0.5000	Negative
7	Highly recommend all of	0.5400	0.1600	Positive
8	Thanks!	0.2000	0.2500	Positive
9	Congrats 8	0.0000	0.0000	Neutral

A collaborative work of UMAIMA SIDDIQUA and SYEDA FATIMA ALI

Fig.7.8: Window 4

7.4 THE TEST PLAN

SNO	TEST CASE SCENARIO	ACTION	EXPECTED RESULT	ACTUAL RESULT
1	Open VSCode	Launch VSCode from Start menu 	VSCode is opened	VSCode is successfully launched 
2	Run Application	Type the command streamlit run main.py in cmd  Command Prompt - streamlit run main.py C:\Users\umaim\OneDrive\Desktop\streamlit-tsa>streamlit run main.py You can now view your Streamlit app in your browser. Local URL: http://localhost:8501 Network URL: http://172.20.10.12:8501	Web application is launched	Successfully launched WebApp  Select Your Activity: Tweet Analyzer Twitter Sentiment Analysis Enter the exact twitter handle of the Personality (without @) Select the Activities: Show Recent Tweets Analyze A collaborative work of UMAIMA SIDDIQUA and SYEDA FATIMA ALI
3	Show recent tweets	Record and run the Selenium script  Project: streamlit-tsa* show_recent_tweets 1 open 2 set window size 3 click 4 type 5 click 'show_recent_tweets' completed successfully	Web app displays 5 recent tweets	Successful display of tweets 
4	Generating	Record and run the	Web app	Successful display of

	WordCloud	Selenium script 	displays the word cloud	wordcloud  Twitter Sentiment Analysis
5	Visualize the Sentiment Analysis	Record and run the Selenium Script 	Web app displays the sentiment analysis	Successful display of sentiment analysis  12 10 8 6 4 2 0 count Neutral Negative Positive Analysis
6	Generate Twitter Data	Record and run the Selenium Script 	Web app displays 100 tweets along with their subjectivity, polarity and analysis	Successful display of results  Twitter Sentiment Analysis

CHAPTER 8

SCREENSHOTS

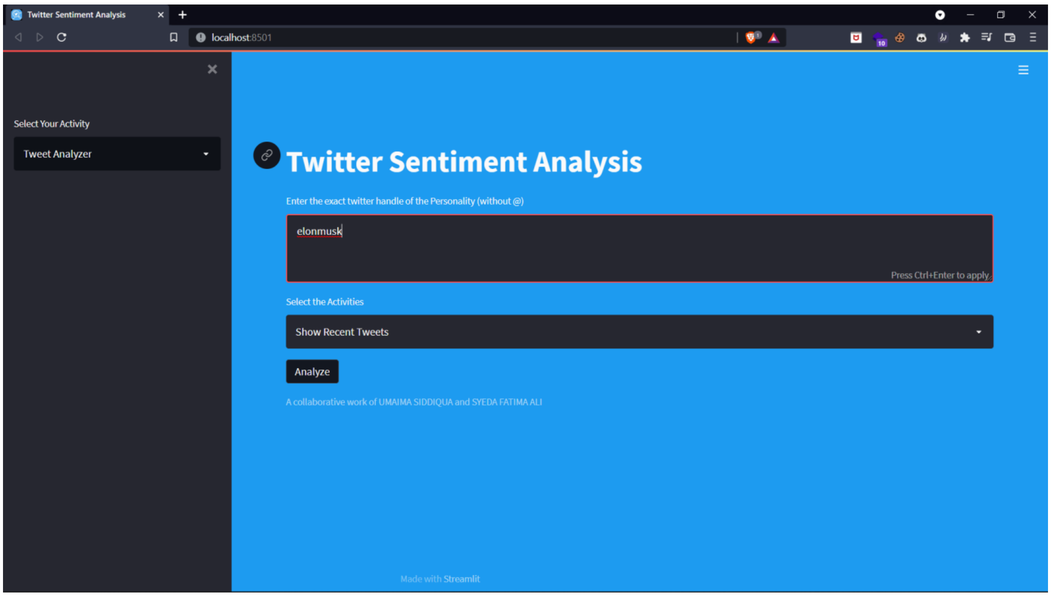
8.1 GENERAL

Sentiment analysis of in the domain of micro-blogging is a relatively new research topic so there is still a lot of room for further research in this area. Decent amount of related prior work has been done on sentiment analysis of user reviews , documents, web blogs/articles and general phrase level sentiment analysis . These differ from twitter mainly because of the limit of 140 characters per tweet which forces the user to express opinion compressed in very short text. The best results reached in sentiment classification use supervised learning techniques such as Naive Bayes.

Here, we analyze the SpaceX's CEO, Elon Musk's twitter account and visualize his tweets into a histogram and a WordCloud.

8.2 VARIOUS SCREENSHOTS

<u>Step 1:</u> Enter the twitter handle without @ in the input field as shown below:



Twitter Sentiment Analysis

localhost:8501

Select Your Activity

Tweet Analyzer

Twitter Sentiment Analysis

Enter the exact twitter handle of the Personality (without @)

elonmusk

Press Ctrl+Enter to apply

Select the Activities

Show Recent Tweets

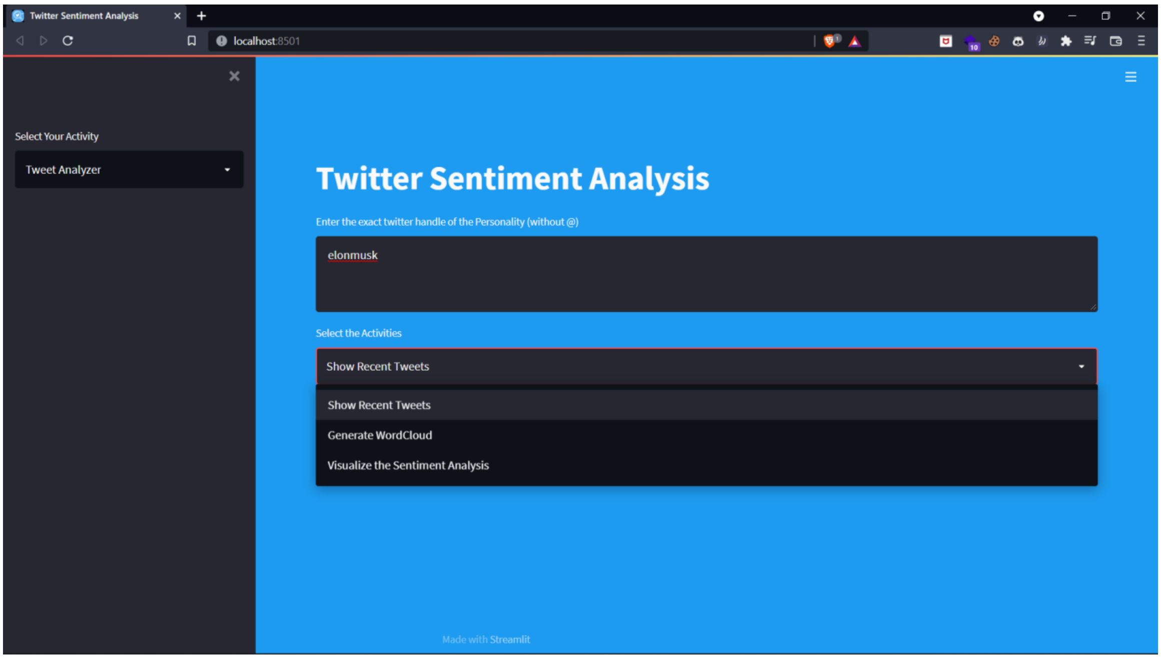
Analyze

A collaborative work of UMAIMA SIDDIQUA and SYEDA FATIMA ALI

Made with Streamlit

Fig.8.1: Screenshot 1

Step 2: Select from the dropdown list to perform preferred activity.



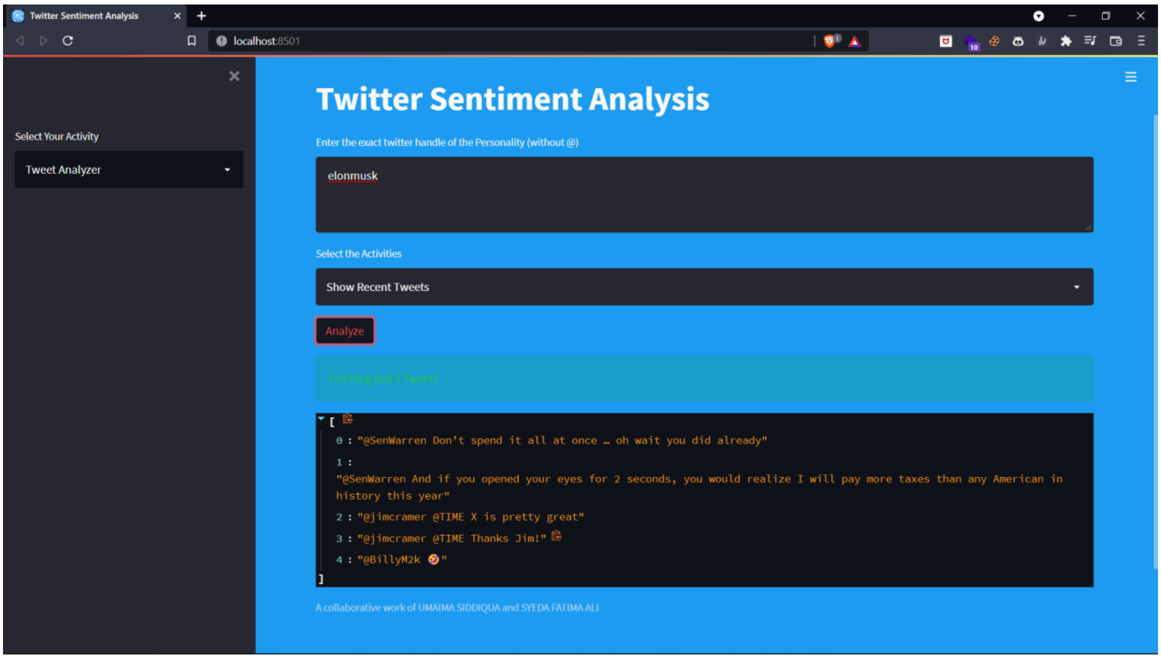
Select Your Activity: Tweet Analyzer

Enter the exact twitter handle of the Personality (without @): elonmusk

Select the Activities: Show Recent Tweets

Fig.8.2: Screenshot 2

Step 3: Selecting the first activity from the list 'Show Recent Tweets'. This activity fetches last 5 tweets .



Select Your Activity: Tweet Analyzer

Enter the exact twitter handle of the Personality (without @): elonmusk

Select the Activities: Show Recent Tweets

Fetching last 5 Tweets

```
[
  0 : "@SenWarren Don't spend it all at once … oh wait you did already"
  1 :
  "@SenWarren And if you opened your eyes for 2 seconds, you would realize I will pay more taxes than any American in
  history this year"
  2 : "@jimcramer @TIME X is pretty great"
  3 : "@jimcramer @TIME Thanks Jim!"
  4 : "@BillyM2k 🤣"
]
```

A collaborative work of UMAIMA SIDDIQUA and SYEDA FATIMA ALI

Fig.8.3: Screenshot 3

Step 4: Select next activity from the dropdown list, 'Generate WordCloud' which gives a visual representation of words based on prominence frequency.

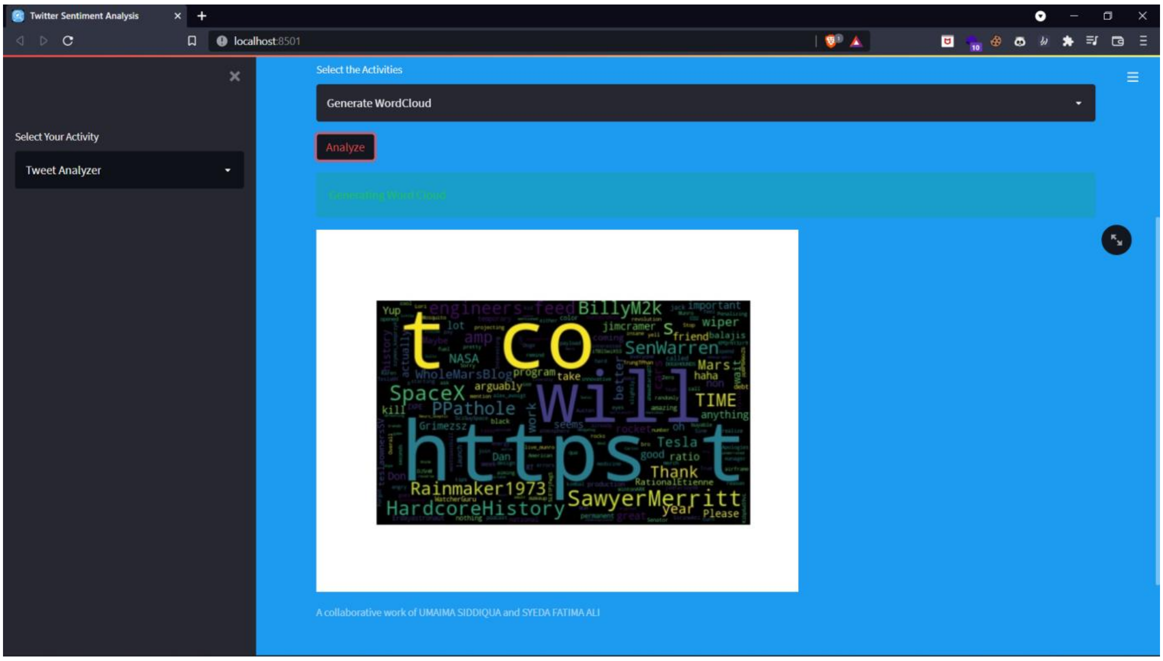


Fig.8.4: Screenshot 4

Step 5: Select next activity, 'Visualize the Sentiment Analysis' which provides a bar graph that tells us how positively or negatively user tweets.

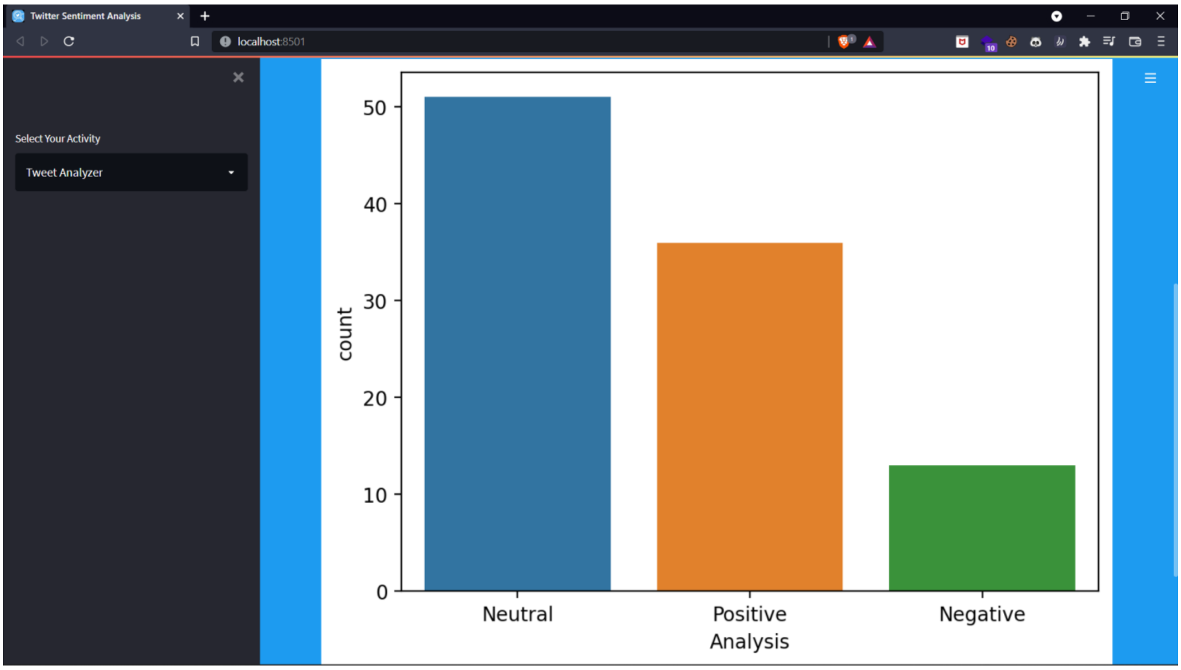
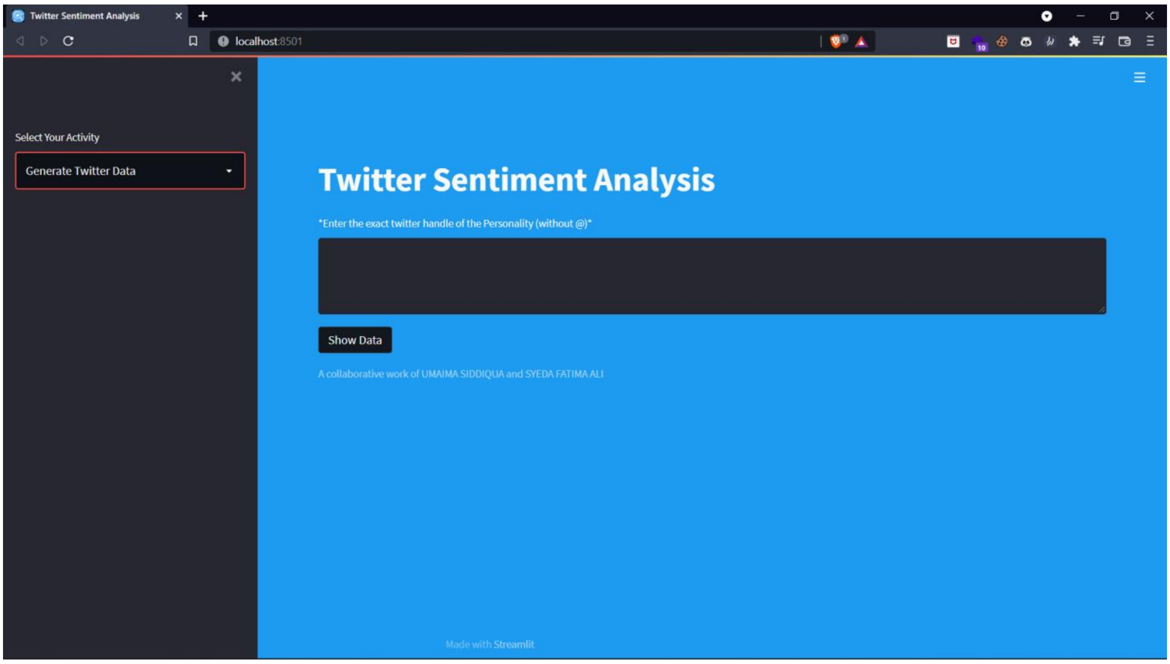


Fig.8.5: Screenshot 5

Step 6: Select ‘Generate Twitter Data’ from the sidebar



Select Your Activity

Generate Twitter Data

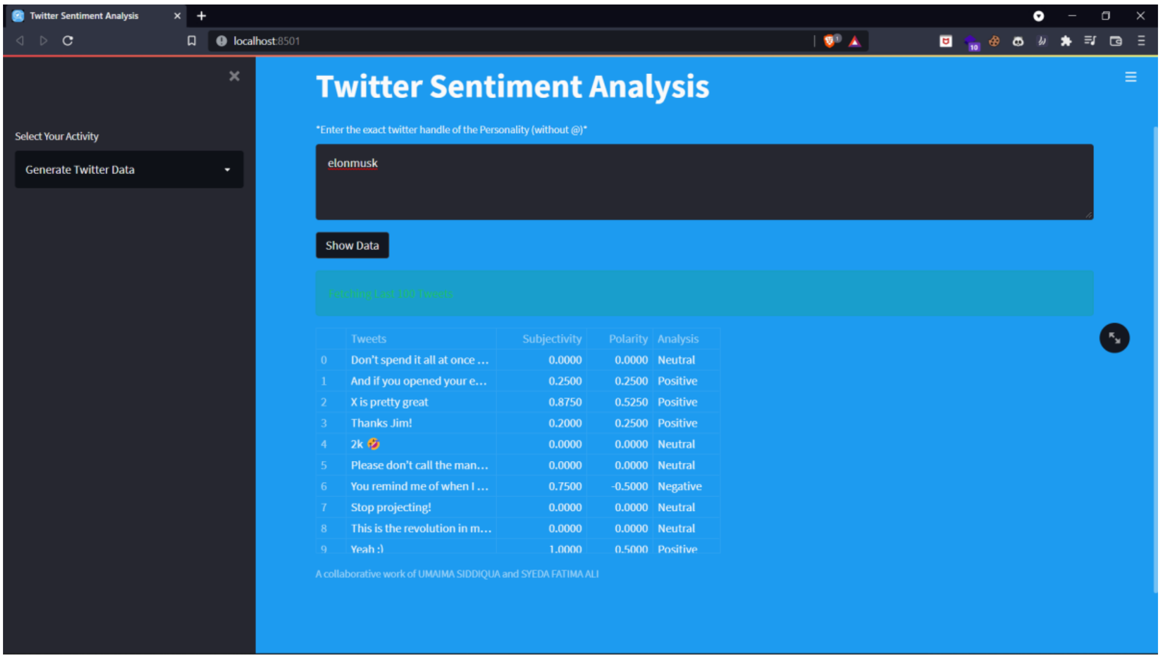
Twitter Sentiment Analysis

Enter the exact twitter handle of the Personality (without @)

A collaborative work of UMAIMA SIDDIQUA and SYEDA FATIMA ALI

Fig.8.6: Screenshot 6

‘Generate Twitter Data’ also takes in the twitter handle of a person and fetches last 100 tweets, tweeted from that particular handle and obtains the sentiment polarity and subjectivity along with its positive/negative analysis.



Select Your Activity

Generate Twitter Data

Twitter Sentiment Analysis

Enter the exact twitter handle of the Personality (without @)

elonmusk

Fetching Last 100 Tweets

	Tweets	Subjectivity	Polarity	Analysis
0	Don’t spend it all at once …	0.0000	0.0000	Neutral
1	And if you opened your e…	0.2500	0.2500	Positive
2	X is pretty great	0.8750	0.5250	Positive
3	Thanks Jim!	0.2000	0.2500	Positive
4	2k 🤣	0.0000	0.0000	Neutral
5	Please don’t call the man…	0.0000	0.0000	Neutral
6	You remind me of when I …	0.7500	-0.5000	Negative
7	Stop projecting!	0.0000	0.0000	Neutral
8	This is the revolution in m…	0.0000	0.0000	Neutral
9	Yeah :)	1.0000	0.5000	Positive

A collaborative work of UMAIMA SIDDIQUA and SYEDA FATIMA ALI

Fig.8.7: Screenshot 7

CHAPTER 9

CONCLUSION AND FUTURE WORKS

9.1 CONCLUSION

As we have seen earlier there are a lot of applications of sentiment analysis over various fields. Though the advantages of sentiment analysis in business sector have ensured its popularity, it has other important uses that are not taken advantage of. The example for this scenario could be detecting violent content spreading across social media platforms. This usually targets a particular community, government, religion, celebrities or politicians and happens dur to the very reason social media is popular, users' ability to exercise their freedom of speech anonymously. Hate directed at an individual or group based on race, caste, religion, ethnic or national origin, sex, handicap, gender identity, or other factors is an abuse of this sovereignty. It actively promotes violence or hate crimes and disrupts society by jeopardising peace, credibility, and human rights, among other things. The aim of this project is to counter this violent extremism and it's spread on Twitter. This project is able to provide a real-time sentiment analysis of any community, government, religion, celebs or politicians across the global anytime using lexical resources.

9.2 FUTURE WORKS

Sarcasm Detection:

In sarcastic text, people express their negative sentiments using positive words. This fact allows sarcasm to easily cheat sentiment analysis models unless they're specifically designed to take its possibility into account.

Sarcasm occurs most often in user-generated content such as Facebook comments, tweets, etc. Sarcasm detection in sentiment analysis is very difficult to accomplish without having a good understanding of the context of the situation, the specific topic, and the environment. [17]

It can be hard to understand not only for a machine but also for a human. The continuous variation in the words used in sarcastic sentences makes it hard to successfully train sentiment analysis models. Common topics, interests, and historical information must be shared between two people to make sarcasm available. While Machine learning algorithms help train data to detect sarcasm, there is still a lot of room for improvements.

Multi-lingual Contribution:

The majority of current sentiment analysis systems address a single language, usually English. However, with the growth of the Internet around the world, users write comments in different languages. Sentiment analysis in only single language increases the risks of missing essential information in texts written in other languages. In order to analyses data in different languages, multilingual sentiment analysis needs to be done. [18] The main problem of multilingual sentiment analysis is the lack of lexical resources. Therefore, further research is needed to make sentiment analysis language independent.

Neutrality Nuance:

Not all documents are either positive or negative; neutral documents also exist. Koppel and Schler (2006) mentioned on their research that Neutral is not a state between positive and negative. Neutral is the lack of sentiment and this category must be detected in sentiment analysis. [19] Neural sentiments are presently classified on the basis of positivity and negativity proportions. Need for a clear-cut parameter for Neutrality detection.

REFERENCES

[1] Khubaib Ahmed Qureshi, Muhammad Sabih,"Un-Compromised Credibility: Social Media Based Multi-Class Hate Speech Classification for Text", IEEE Access, Volume: 9, 02 August 2021

[2] Saadat M. Alhashmi,Ahmed M. Khedr,Ifra Arif,Magdi El Bannany, "Using a Hybrid-Classification Method to Analyze Twitter Data During Critical Events", IEEE Access, Volume: 9, 08 October 2021

[3] Zhao Jianqiang; Gui Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis", IEEE Access, Volume: 5, 22 February 2017

[4] Tanveer Khan; Antonis Michalas,"Seeing and Believing: Evaluating the Trustworthiness of Twitter Users", IEEE Access, Volume: 9, 19 July 2021

[5] Vishal A. Kharde and S.S. Sonawane, "Sentiment Analysis of Twitter Data: A Survey of Techniques", International Journal of Computer Applications (0975 – 8887)
Volume 139 – No.11, April 2016

[6] https://medium.com/nerd-for-tech/sentiment-analysis-lexicon-models-vs-machine-learning-b6e3af8fe746

[7] https://medium.com/analytics-vidhya/future-of-sentiment-analysis-13a9be14218b

[8] https://www.analyticsvidhya.com/blog/2021/07/performing-sentiment-analysis-with-naive-bayes-classifier/

[9] https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/

[10] https://developer.twitter.com/en/portal/apps/22299663/keys

[11] https://docs.tweepy.org/en/latest/api.html

[12] https://textblob.readthedocs.io/en/dev/index.html

[13] https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0171649

[14] https://www.nitj.ac.in/nitj_files/links/Machine_and_Deep_Learnig_for_Research_25102021_98566.pdf

[15] https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML?__cf_chl_managed_tk__=VWldwdFtH8_plC1oS1ytajlntqGee73ybTIG3GdQeP0-1641483428-0-gaNycGzNCCU

[16] https://www.selenium.dev/documentation/ide/

[17] https://www.toptal.com/deep-learning/4-sentiment-analysis-accuracy-traps

[18] https://link.springer.com/article/10.1007/s12559-016-9415-7

[19] https://blog.datumbox.com/the-importance-of-neutral-class-in-sentiment-analysis/