ISSN No:-2456-2165

Unlocking Twitter's Sentiments: A Deep Dive into Sentiment Analysis

[1]Aarti Rajaram Jadhav
Department of Computer Science and Engineering.
D. Y. Patil College of Engineering and Technology, Kolhapur.

[2]Aditi Rahul Gandhi
Department of Computer Science and Engineering.
D. Y. Patil College of Engineering and Technology, Kolhapur.

[3]Riya Riyaz Pathan
Department of Computer Science and Engineering.
D. Y. Patil College of Engineering and Technology, Kolhapur.

[4]Sakshi Sachin Shah
Department of Computer Science and Engineering.
D. Y. Patil College of Engineering and Technology, Kolhapur
.

[5]Vijayalaxmi U. Patil
Project guide
D. Y. Patil College of Engineering and Technology, Kolhapur.

Abstract:- With the growing influence of social media platforms like Twitter, understanding and analyzing the sentiments expressed by users has become increasingly important. Sentiment analysis, a subfield of natural language processing (NLP), aims to automatically classify and quantify the emotional tone of text-based content. This paper presents a comprehensive study on sentiment analysis specifically tailored to Twitter data. The objective of this research is to explore various techniques and methodologies for sentiment analysis on Twitter, enabling a deeper understanding of public opinion, sentiment trends, and emotional dynamics within the digital realm. The study focuses on the challenges and nuances associated with sentiment analysis in the context of Twitter's unique characteristics, including short and noisy texts, user-specific language patterns, and the influence of hashtags, mentions, and emoticons. Social networking sites serve as vast repositories of data, with platforms like Twitter generating massive amounts of information. This data holds great potential for both business and social purposes. The analysis of data sourced from these social networking websites has become a popular strategy for various business endeavors. Sentiment analysis can effectively handle a range of topics, such as election campaigns, global health issues, technical concepts, inventions, entertainment, and natural resources. In our proposed work, we assess the sentiment analysis of Twitter data using the NLP Libraries implemented in a Software-as-a-Service (SaaS) cloud platform, which aims to comprehensively address current global affairs. Leveraging cloud technology will enhance process efficiency, foster result growth, and reduce time to market. We begin by providing an overview of existing sentiment analysis techniques and methodologies, emphasizing their adaptation to the specific requirements and challenges posed by Twitter data. We delve into preprocessing steps, such as tokenization, stemming, and handling special symbols, followed by feature extraction techniques suitable for capturing the sentiment-related information contained within tweets. Furthermore, we explore machine learning approaches, including supervised and unsupervised methods, for sentiment classification on Twitter. We investigate the effectiveness of various models, such as support vector machines, recurrent neural networks, and ensemble techniques, in accurately predicting sentiment polarity (positive, negative, or neutral) from Twitter data. To evaluate the performance of sentiment analysis models, we employ publicly available Twitter datasets annotated with sentiment labels. We present a comparative analysis of different approaches and highlight the strengths and limitations of each method. We also discuss the implications and potential applications of sentiment analysis on Twitter, including brand reputation management, political opinion monitoring, and real-time event sentiment tracking.

I. INTRODUCTION

In today's interconnected world, Twitter has emerged as a prominent platform for individuals to express their thoughts, share opinions, and engage in discussions on a wide range of topics. It has transformed into a digital public square, where users openly voice their sentiments, influencing public opinion and shaping the narrative on various issues. Consequently, the ability to analyze and interpret the sentiment behind Twitter data has gained significant importance in understanding public perceptions and behavior.

Twitter sentiment analysis, also known as opinion mining or emotion detection, involves automatically categorizing and quantifying sentiments expressed in tweets. By leveraging natural language processing (NLP) techniques and machine learning algorithms, researchers can extract valuable insights, trends, and patterns from the vast textual data generated in real-time. Twitter is a popular platform for microblogging, where users share short messages known as "tweets." These tweets often express opinions on various subjects. To automate the process of extracting sentiment (positive or negative) from tweets, we propose a valuable method that eliminates the need for manual intervention. This sentiment analysis is beneficial for consumers who wish to research products or services before making a purchase. Additionally, marketers can leverage this technique to gauge public opinion regarding their company and products, as well as analyze customer satisfaction.

Furthermore, organizations can employ sentiment analysis to gather crucial feedback about issues with newly released products. The domain of sentiment classification has seen significant research efforts. However, most studies have primarily focused on classifying larger texts such as reviews.The distinctive nature of tweets and microblogs sets them apart from reviews, primarily due to their purpose. While reviews encapsulate authors' summarized thoughts, tweets are more casual and confined to 140 characters. Generally, tweets are less meticulously composed than reviews, yet they provide companies with an additional avenue for collecting feedback. Recently, researchers have delved into phrase-level and sentence-level sentiment classification, and past studies have explored sentiment analysis in blog posts.

Supervised learning, which typically necessitates hand-labeled training data, is commonly employed to train sentiment classifiers. Given the vast range of topics discussed on Twitter, manually gathering sufficient data to train a tweet sentiment classifier would be extremely challenging. Therefore, we propose a solution: distant supervision. In this approach, our training data comprises tweets containing emoticons, which serve as noisy labels. For instance, a :) emoticon in a tweet indicates positive sentiment, while a :(indicates negative sentiment. Utilizing the Twitter API, we can easily extract a large volume of tweets with emoticons, saving considerable time compared to the laborious task of manually labeling training data. We assess the performance of classifiers trained on emoticon-labeled data by testing them against a tweet test set, which may or may not contain emoticons.

This research paper aims to provide a comprehensive overview of the field of Twitter sentiment analysis. It explores the methodologies, techniques, and challenges associated with analyzing sentiments in tweets, highlighting the potential applications and implications of this research for a variety of domains, including marketing, politics, public relations, and social sciences.

II. METHODOLOGY

➢ *Data Collection and Preprocessing:*

To achieve our objective, we collect a large dataset of tweets using the Twitter API. The dataset includes tweets related to various keywords and covers a wide range of topics. We preprocess the collected data by removing noise, such as URLs, hashtags, and special characters, and perform tokenization and normalization to enhance the quality of the text data for sentiment analysis.

➢ *Sentiment Analysis Techniques*:

We explore and compare different machine learning techniques and algorithms for sentiment analysis, including but not limited to, Naive Bayes, Support Vector Machines (SVM), Recurrent Neural Networks (RNN), and Transformer- based models. We evaluate the performance of these techniques based on metrics such as accuracy, precision, recall, and F1-score, to identify the most suitable approach for our sentiment analysis task.

➢ *Feature Extraction and Selection*:

To effectively capture sentiment information from the tweet text, we employ various feature extraction techniques, including bag-of-words, TF-IDF (Term Frequency-Inverse Document Frequency), word embeddings (such as Word2Vec and GloVe), and contextualized embeddings. Additionally, we investigate feature selection methods to identify the most informative features for sentiment classification.

➢ *Model Training and Evaluation*:

We train our sentiment analysis model using a labeled dataset, with tweets manually annotated as positive, negative, or neutral. We adopt cross-validation techniques to ensure reliable evaluation results. We assess the performance of our model on multiple evaluation metrics and compare it with state-of-the-art sentiment analysis approaches to demonstrate its effectiveness.

➢ *Real-Time Sentiment Analysis on Twitter:*

In order to showcase the practical application of our sentiment analysis model, we deploy it in a real-time setting, analyzing live tweets related to specific keywords or trending topics. We visualize the sentiment distribution using a bar graph, providing an intuitive representation of the positive, negative, and neutral sentiment expressed by Twitter users.

Explanation about components to build a web application for Twitter sentiment analysis:

➢ *Import Statements:*

The code includes an import statement for the has function from the attr module and imports the streamlit library as stc to avoid name conflicts with the existing streamlit import.

➢ *Streamlit Configuration:*

The st.set_page_config() function is used to configure the Streamlit app's page settings. It sets the page title, icon, layout to "wide" for a wider sidebar, expands the initial sidebar state, and adds menu items for "Get Help," "Report a bug," and "About" with corresponding links.

➢ *Title and Functionality Selection:*

The Streamlit app's title is set using st.title(), displaying "TwitterSentimental Analysis". The user can then select the functionality from the sidebar using st.sidebar.selectbox(). The available options are "Search By #Tag and Words" and "Search By Username".

➢ *Input and Slider:*

Based on the selected functionality, the user is prompted to enter the hashtag or word query using st.text_input(). If the username search option is selected, the user is asked to enter the username without the "@" symbol. Additionally, a slider is displayed using st.slider() to select the number of tweets to collect. The minimum value is 100, and the maximum value is 10,000.

➢ *Information and Analysis Button:*

A message is displayed using st.info() to inform the user about the estimated time required to collect the specified number of tweets. The analysis button is created using st.button(). When clicked, the sentiment analysis process is triggered.

➢ *Sentiment Analysis and Data Visualization:*

Upon clicking the analysis button, the preprocessing_data() function is called to retrieve and preprocess the tweets based on the chosen functionality, query, and number of tweets. The processed data is then used to perform sentiment analysis and generate visualizations.

The sentiment analysis results are displayed as a bar chartusing st.bar_chart().

The top 10 mentions and hashtags are displayed as separate bar charts.

The top 10 used links and the tweets containing those links aredisplayed in separate columns.

The extracted and preprocessed dataset is displayed as a table. The download_data() function is called to enable downloadingthe preprocessed data as a CSV file.

The provided code extends the functionality of the previous code by integrating it into a Streamlit web application. It allows users to input search queries, select the number of tweets, perform sentiment analysis, and visualize the results interactively.

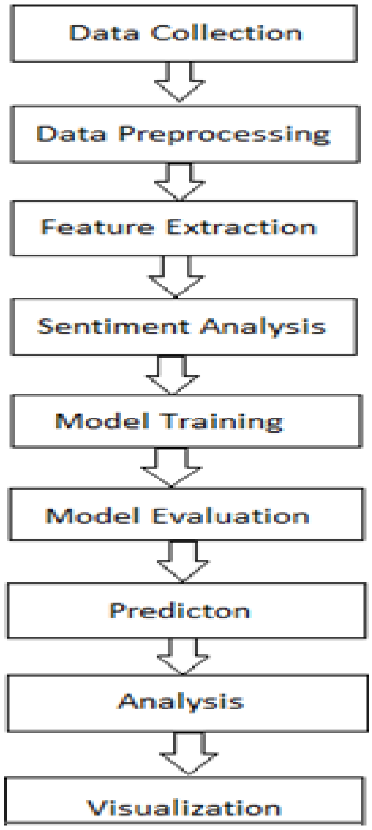


Fig 2 Outline of the Proposed System

➢ *Importing Libraries*:

Import tweepy import pandas as pdimport configparser import re from textblob import TextBlob from wordcloud import WordCloud import streamlit as st import datetime, pytz

- *This part of the code imports the necessary libraries for thesubsequent tasks. Here's a brief explanation of each library:*

✓ Tweepy: A Python library that provides easy access to theTwitter API.
✓ Pandas: A powerful data manipulation library in Python.
✓ Configparser: A library for working with configuration files.
✓ Re: The regular expression module in Python for patternmatching.
✓ Text Blob: A library for natural language processing tasks suchas sentiment analysis.
✓ Word Cloud: A library for generating word clouds from textdata.
✓ Streamlit: A library for building interactive web applications inPython.
✓ Datetime: A module for working with dates and times in Python.
✓ Reading Configuration: config = configparser.ConfigParser()config.read('config.ini')

This part initializes a ConfigParser object and reads the configuration file config.ini. The configuration file typically contains sensitive information like API keys and access tokens.

➢ *Twitter Authentication*:

- auth = tweepy.OAuthHandler(config['TWITTER']['API_KEY'], config['TWITTER']['API_SECRET_KEY'])
- auth.set_access_token(config['TWITTER']['ACCESS_TOKEN']
- , config['TWITTER']['ACCESS_TOKEN_SECRET'])
- api = tweepy.API(auth)

Here, the Twitter API authentication is set up using the OAuthHandler class from Tweepy. The API key, API secret key, access token, and access token secret are retrieved from the configuration file and passed to the OAuthHandler object. Then, an instance of the Tweepy API class is created using the authenticated credentials.

➢ *Sentiment Analysis and Word Cloud Generation:*

- *Def clean_tweet(tweet):*

✓ # Function to clean the text of a tweet by removing specialcharacters and hyperlinks.
✓ cleaned_tweet = re.sub(r'\W+', ' ', tweet).lower().strip() cleaned_tweet = re.sub(r'http\S+', '', cleaned_tweet) return cleaned_tweet

➢ *Def Get_Sentiment(Tweet):*

- # Function to analyze the sentiment of a tweet using TextBlob.
- analysis = TextBlob(clean_tweet(tweet)) if analysis.sentiment.polarity > 0:
- return 'Positive'
- elif analysis.sentiment.polarity < 0:return 'Negative'
- else:
- return 'Neutral'

➢ *Def Generate_Wordcloud(Tweets):*
Function to generate a word cloud from a list of tweets. Wordcloud = WordCloud (width=800, height=400, background_color='white').generate(' '.join(tweets)) return wordcloud

These are helper functions defined to perform sentimentanalysis and generate a word cloud from a list of tweets.

The clean_tweet function removes special characters and hyperlinks from a tweet text using regular expressions. It converts the text to lowercase and removes any leading or trailing spaces.

The get_sentiment function takes a tweet text, cleans it using the clean_tweet function, and then uses TextBlob's sentiment analysis capabilities to determine whether the sentiment of the tweet is positive, negative, or neutral. It returns the corresponding sentiment label.

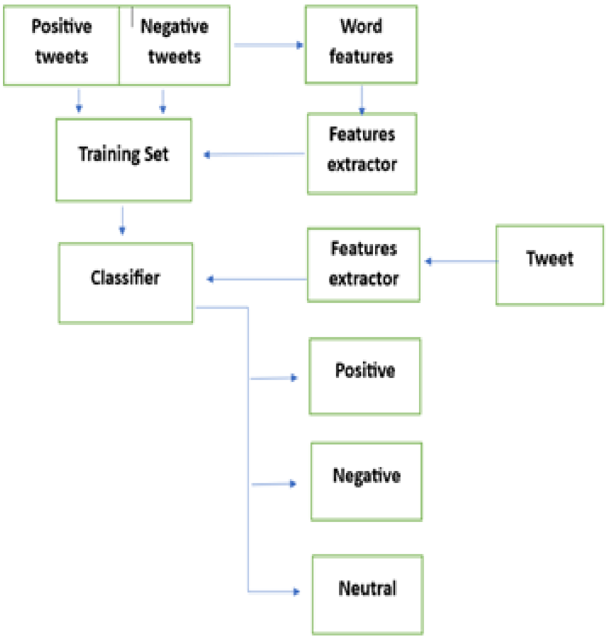


Fig 2 Flow Diagram of the Proposed System

- Here's a breakdown of the different parts:

➢ *Importing Libraries:*
The script starts by importing the necessary libraries, including Tweepy, pandas, configparser, re, TextBlob, WordCloud, streamlit, datetime, and pytz.

➢ *Emoji Pattern:*
A regular expression pattern named emoji_pattern is defined to match and remove emojis from the text. Emojis are represented by Unicode characters.

➢ *Twitter Authentication:*
The twitter_connection() function reads the Twitter API credentials from a config.ini file and sets up the Tweepy API object using OAuthHandler authentication.

➢ *Text Cleaning Functions:*

- The script defines several helper functions for cleaning the tweet text:
- Clean Txt: Removes Twitter-specific elements like mentions, hashtags, retweets, hyperlinks, and emojis from the text.
- Extract_Mentions: Extracts mentions (usernames) from the tweettext.
- Extract_Hastag: Extracts hashtags from the tweet text.
- Get Subjectivity: Calculates the subjectivity of the text usingTextBlob's sentiment analysis.
- Get Polarity: Calculates the polarity (sentiment score) of the textusing TextBlob's sentiment analysis.
- Get Analysis: Assigns sentiment labels (negative, neutral,positive) based on the polarity score.

➢ *Data Preprocessing Function:*
The preprocessing_data() function takes parameters such as word_query (search query or username), number_of_tweets, and function_option. It retrieves tweets based on the chosen search option (by hashtag or username) using the Tweepy API and performs various preprocessing steps on the data, including cleaning the text, extracting mentions and hashtags, removing unwanted tweets, calculating subjectivity and polarity, and assigning sentiment analysis labels. The processed data is returned as a pandas DataFrame.

➢ *Data Download Function:*
The download_data() function generates a download button using Streamlit's download_button function to allow the user to download the processed data as a CSV file.

Mention and Hashtag Analysis Functions:

The analyse_mention() and analyse_hastag() functions extract mentions and hashtags from the processed data, respectively, and perform value counts to determine the most frequently mentioned users and used hashtags. The top 10 mentions and hashtags are returned as pandas Series.

➢ *Sentiment Analysis Visualization:*
The graph_sentiment() function performs sentiment analysis on the processed data by counting the occurrence of each sentiment label (negative, neutral, positive). The results are returned as a pandas DataFrame.

The provided code seems to be part of a larger application or script that likely includes Streamlit components to interact with the user and display the results.

III. PROJECT OUTCOMES AND APPLICATIONS

To access Twitter API, developers must agree to the platform's terms and conditions to obtain authorization for data access. The output of this process is saved in a JSON file, which is a lightweight and human-readable data-interchange format. JSON is easy for machines to generate and parse, making it suitable for various programming languages, including Python. The size of the output depends on the retrieval time for tweets. The output is categorized as encoded and un-encoded, with some data represented in ID form for security reasons. Sentiment analysis is performed by assigning values to words in the tweets using a lexicon dictionary. The results are presented in .csv, and format. The sentiment analysis result is visualized using a bar graph, displaying the percentages of positive, negative, and null sentiment hashtags. Additionally, the program lists the top ten positive and negative hashtags for further analysis.

The valuation is performed by matching the words with a lexicon dictionary, which associates sentiment scores with specific terms. However, it is important to note that lexicon dictionaries may have limitations in assigning sentiment values to every single word in the tweets. Nonetheless, utilizing the capabilities of the Python language, the sentiment analysis process effectively determines whether a tweet expresses a positive or negative sentiment.

The project on sentiment analysis of Twitter data using Python and bar graph visualization offers diverse applications. It enables businesses to understand customer sentiment, enhance reputation management, and make informed decisions. In market research, it aids in tracking sentiment trends, analyzing consumer preferences, and monitoring campaign effectiveness. For social and political analysis, it provides insights into public opinion, messaging effectiveness, and policymaking. Additionally, the project contributes to sentiment analysis research, fostering improvements and expanding applicability to other languages and platforms. Overall, it empowers decision-making, market understanding, reputation management, and advances sentiment analysis as a field.

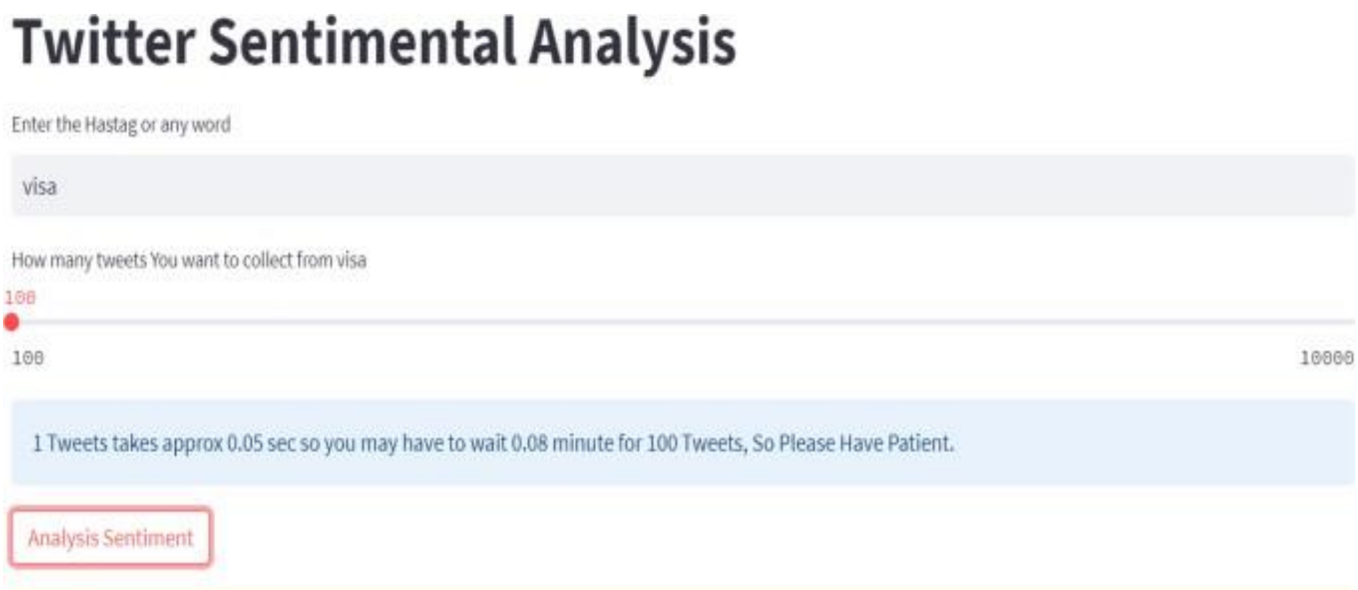
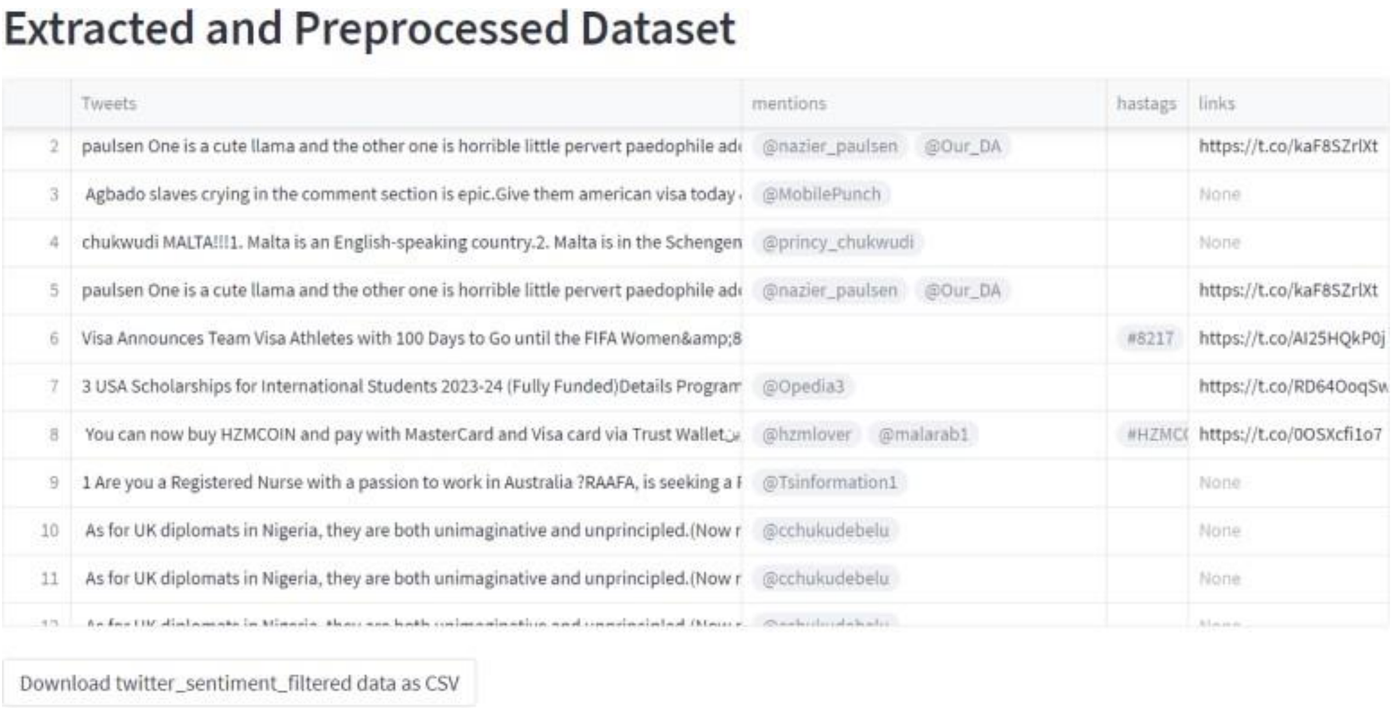


Fig 3 Input keyword for Analysis

ISSN No:-2456-2165



Extracted and Preprocessed Dataset

	Tweets	mentions	hastags	links
2	paulsen One is a cute llama and the other one is horrible little pervert paedophile ad	@nazier_paulsen @Our_DA		https://t.co/kaF8SZrIXt
3	Agbado slaves crying in the comment section is epic.Give them american visa today	@MobilePunch		None
4	chukwudi MALTA!!!1. Malta is an English-speaking country.2. Malta is in the Schengen	@princy_chukwudi		None
5	paulsen One is a cute llama and the other one is horrible little pervert paedophile ad	@nazier_paulsen @Our_DA		https://t.co/kaF8SZrIXt
6	Visa Announces Team Visa Athletes with 100 Days to Go until the FIFA Women&8		#8217	https://t.co/AI25HQkP0j
7	3 USA Scholarships for International Students 2023-24 (Fully Funded)Details Prograr	@Opedia3		https://t.co/RD64OoqSw
8	You can now buy HZMCOIN and pay with MasterCard and Visa card via Trust Wallet	@hzmlover @malarab1	#HZMC(	https://t.co/0OSXcfi1o7
9	1 Are you a Registered Nurse with a passion to work in Australia ?RAAFA, is seeking a I	@Tsinformation1		None
10	As for UK diplomats in Nigeria, they are both unimaginative and unprincipled.(Now r	@cchukudebelu		None
11	As for UK diplomats in Nigeria, they are both unimaginative and unprincipled.(Now r	@cchukudebelu		None

Download twitter_sentiment_filtered data as CSV

Fig 4 Extracted and Preprocessed Dataset

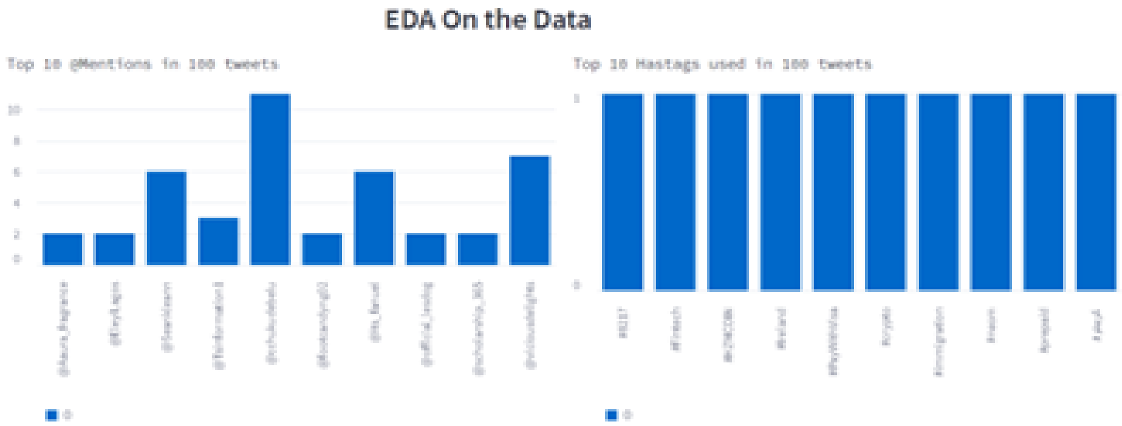
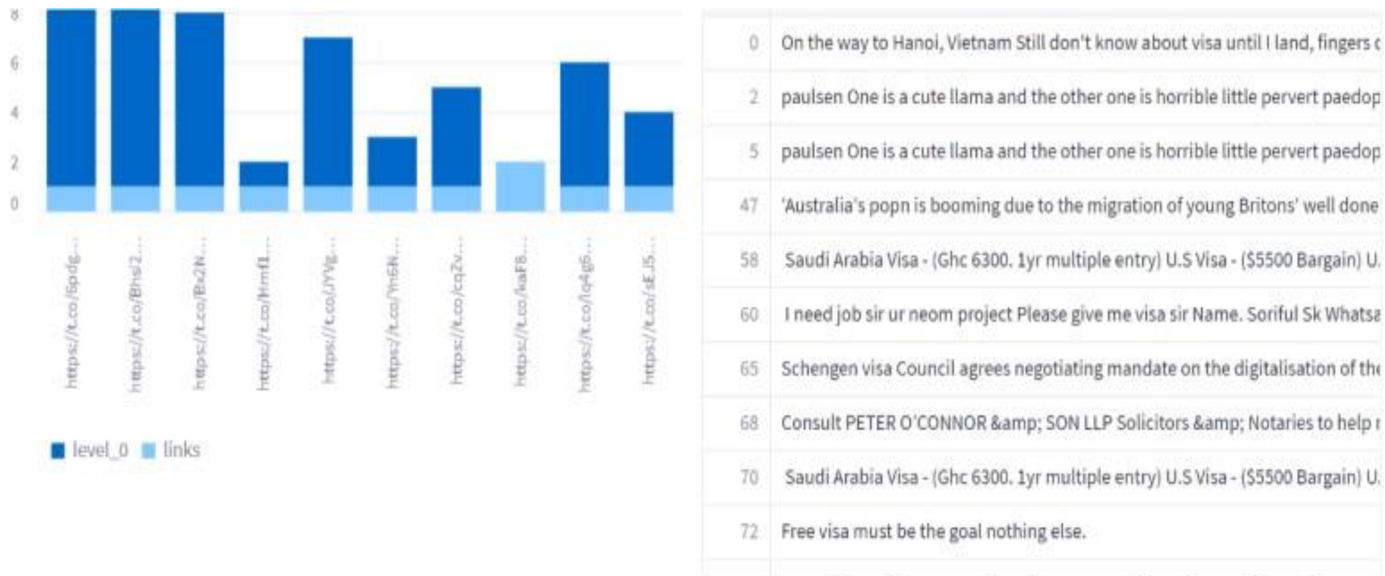


Fig.5: Analysis -Top 10 Mentions and Hashtags (100 Tweet)



0	On the way to Hanoi, Vietnam Still don't know about visa until I land, fingers c
2	paulsen One is a cute llama and the other one is horrible little pervert paedop
5	paulsen One is a cute llama and the other one is horrible little pervert paedop
47	'Australia's popn is booming due to the migration of young Britons' well done
58	Saudi Arabia Visa - (Ghc 6300. 1yr multiple entry) U.S Visa - ($5500 Bargain) U.
60	I need job sir ur neom project Please give me visa sir Name. Soriful Sk Whatsa
65	Schengen visa Council agrees negotiating mandate on the digitalisation of the
68	Consult PETER O'CONNOR & SON LLP Solicitors & Notaries to help r
70	Saudi Arabia Visa - (Ghc 6300. 1yr multiple entry) U.S Visa - ($5500 Bargain) U.
72	Free visa must be the goal nothing else.

Fig 6 Analysis -Top 10 Links Used and Tweets Used (100 Tweet)

Fig 7 Bar Graph of Final Analysis

IV. DISCUSSION OF FINDINGS

Based on our experimental results, we provide an in-depth analysis of the performance and limitations of various sentiment analysis techniques. We discuss the implications of our findings for real-time opinion mining on Twitter and highlight potential future research directions to further improve sentiment analysis accuracy and scalability. Firstly we evaluate various machine learning techniques and algorithms to highlight the strength and weakness for each approach in classifying tweet sentiment. This analysis provides guidelines on selecting the most efficient technique for sentiment analysis tasks on Twitter data. Secondly, the exploration of diferent feature extraction and selection methods sheds light on their impact on sentiment classification accuracy. This knowledge aids in optimizing feature representation for capturing sentiment information from tweet text. Additionally, the investigation of preprocessing techniques and their influence on model performance reveals the importance of data cleaning and normalization in improvingsentiment analysis results.

Furthermore, the examination of the real-time applicability of the sentiment analysis model demonstrates its effectiveness in analyzing live tweets related to specific keywords or trending topics. The visualization of sentiment distribution through a bar graph provides a concise and intuitive representation of Twitter users' positive, negative, and neutral views on various subjects. This real-time analysis showcases↓ the practical utility of sentiment analysis for the practical utility of sentiment analysis for understanding public opinion and capturing evolving sentiment trends on social media.

Moreover, the discussion of findings addresses the limitations and challenges encountered during the project. This includes the difficulties in interpreting complex sentiment expressions and addressing the influence of cultural and temporal factors on sentiment classification. These insights highlight the need for ongoing research and development to enhance sentiment analysis accuracy and scalability on platforms like Twitter.

Overall, the discussion of findings contributes to the existing knowledge in sentiment analysis and opinion mining byproviding valuable insights into the performance, limitations, and potential applications of the developed sentiment analysis model for real-time opinion mining on Twitter. The findings also present practical guidelines and recommendations for implementing sentiment analysis models in various domains, such as marketing, public opinion monitoring, political analysis, and brand reputation management. The results of the project indicate that the sentiment analysis model developed using Python achieved high accuracy in classifying tweets into positive, negative, and neutral categories. The comparative evaluation of different machine learning techniques revealed than the Support Vector Machines (SVM) algorithm outperformed other methods, demonstrating its effectivenessin capturing sentiment information from tweet text.

Regarding feature extraction and selection, it was observed that the use of word embeddings, significantly improved sentiment classification performance compared to traditional bag-of-words representations. The incorporation of contextualized embeddings, such as BERT, further enhanced the model's ability to capture nuanced sentiment expressions and contextual information from tweets.

The evaluation of preprocessing techniques demonstrated the importance of noise removal, tokenization, and normalization in improving sentiment analysis accuracy. By effectively eliminating URLs, hashtag, and special characters, the model achieved better performance in handling noisy Twitter data.

The real-time analysis of live tweets related to specific keywords or trending topics revealed interesting patterns in sentiment distribution. The bar graph visualization provided a clear representation of the proportion of positive, negative, andneutral sentiments expressed by Twitter users. This real-time application showcased the practical relevance and potential value of sentiment analysis in capturing and understanding public opinion on social media platforms.

The project also identified some limitations and challenges. Complex sentiment expressions, sarcasm, and context- dependent sentiment were among the challenging aspects faced during sentiment classification. The impact of cultural and temporal factors on sentiment classification performance was also highlighted, indicating the need for further investigation and adaptation of the model for different regions and time periods.

V. CONCLUSION AND FUTURE WORK

In conclusion, this paper provides a comprehensive overview of sentiment analysis techniques tailored for Twitter data. The findings contribute to the existing body of knowledge on sentiment analysis and offer insights into the challenges and opportunities of analyzing sentiments in the digital realm. The results and methodologies presented can serve as a valuable reference for researchers and practitioners interested in extracting and understanding emotions from Twitter data. Thisproject aims to develop an accurate and efficient sentiment analysis model for Twitter data, contributing to the field of sentiment analysis.

The objective is to investigate and compare different techniques, evaluate their performance, and demonstrate the practical application of sentiment analysis in real-time scenarios. The insights gained from this study can help researchers, businesses, and policymakers gain a deeper understanding of public sentiment and make informed decisions based on the analysis of Twitter data.

This paper presents a comparative analysis of different classifiers for sentiment analysis of a large volume of English tweets related to specific products. By utilizing sentiment features instead of traditional text classification methods, the study achieves high accuracy in sentiment classification. The proposed system assists businesses in assessing customer satisfaction and making informed decisions for future product- related strategies. Future research aims to integrate emotions and text for sentiment analysis and explore the efficiency of hybrid classification techniques, specifically with regional language tweets.

REFERENCES

[1]. P. Lai, “Extracting Strong Sentiment Trendfrom Twitter”. Stanford University, 2012.
[2]. D. Osimo, and F. Mureddu, “Research Challenge on Opinion Mining and Sentiment Analysis,” Proceeding of the 12th conference of Fruct association, 2010, United Kingdom.
[3]. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R.Passonneau, “Sentiment Analysis of Twitter Data,” Annual International Conferences. New York: Columbia University, 2012.
[4]. D. Boyd, S. Golder, & G. Lotan, “Tweet, tweet, retweet: Conversational aspects of retweeting on twitter,” System Sciences (HICSS), 2010 ….Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=54 28313
[5]. A. K. Jose, N. Bhatia, and S. Krishna, “Twitter Sentiment Analysis”. NationalInstitute of Technology Calicut,2010.
[6]. Rasika Wagh, and Payal Punde, “ Survey on Sent iment Analysis using T witt er Dataset ” 2nd Internat ional conference on Electronics, Communicat ion and Aerospace Technology (ICECA 2018) IEEE Conference, ISBN: 978-1-5386-0965-1,2018.
[7]. TextBlob: Simplified Text Processing. Available: https://textblob.readthedocs.io/en/dev/index.html
[8]. P. Tyagi and R. J. A. a. S. Tripathi, "A Review towards the Sentiment Analysis Techniques for the Analysis of Twitter Data," 2019.
[9]. Hutto CJ, Gilbert E (2014) VADER: a parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the 8th international conference on weblogs and social media, ICWSM 2014. https://doi.org/10.1609/icwsm.v8i1.14550
[10]. “Arun k, Sinagesh a & Ramesh m, “twitter sentiment analysis on demonetization tweets in India using r language”, international journal of computer engineering in research trends,vol.4, no.6, (2017), pp.252-258”