SOCIAL MEDIA SENTIMENT ANALYSIS USING NLP

A PROJECT REPORT

Submitted by

ATIULLAH ANSARI	**(730920201009)**
BHUPENDRA KUMAR	**(730920201011)**
KANNU PRIYA	**(730920201020)**
SONU KUMAR	**(730920201048)**

In partial fulfillment of the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

EXCEL ENGINEERING COLLEGE (AUTONOMOUS)

Komarapalayam-637 303

APRIL 2024

EXCEL ENGINEERING COLLEGE (AUTONOMOUS)

KOMARAPALAYAM

BONAFIDE CERTIFICATE

Certified that this project report **"SOCIAL MEDIA SENTIMENT ANALYSIS USING NLP"** is the bonafide work of **ATIULLAH ANSARI (730920201009), BHUPENDRA KUMAR (730920201011), KANNU PRIYA (730920201020),** and **SONU KUMAR (730920201048)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Prof. P. JAYAPRABHA, M.E.,	**Dr. P. SELVARAJU**
HEAD OF THE DEPARTMENT,	**ASSOCIATE PROFESSOR,**
Department of AI&DS	Department of AI&DS
Excel Engineering, College,	Excel Engineering, College,
Komarapalayam-637 303.	Komarapalayam-637 303.

Submitted for Project Viva-Voce held on_____________

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude in-depth to our institution "**EXCEL ENGINEERING COLLEGE (AUTONOMOUS)**" for providing us the opportunity to do the project.

We are greatly indebted to our Honorable Chairman **Dr. A.K.NATESAN, M.Com., MBA (NIT)., M.Phil., Ph.D., FTA,** for providing all the necessary facilities for the successful completion of the project.

We express our heartiest gratitude and thanks to our respected Vice Chairman **Dr. N.MATHAN KARTHICK, M.B.B.S., MISTE., PHIE.,** of **EXCEL GROUP OF INSTITUTIONS** for providing all the facilities completing the project.

We like to place on record to our beloved Principal and Executive Director **Dr.K.BOMMANNA RAJA, M.E., Ph.D.,** for his valuable suggestion in our entire project.

We thank the Head of the Department **Prof.P.JAYAPRABHA, M.E.,** of Artificial Intelligence and Data Science for her guidance, constant inspiration and encouragement.

We wish to express our Heartfelt Thanks and sincere acknowledgment to our respected Guide **Dr.P.SELVARAJU** for her encouragement and dedicated guidance

We take the privilege to record our everlasting and loving thanks to our parents for their kind help and support which render in bringing our Project full manner.

ABSTRACT

This project focuses on the development of a Sentiment Analysis Tool for evaluating emotions and opinions expressed in social media content. The significance of sentiment analyzing in insights in to public perception, customer feedback, and brand reputation is emphasized. The objectives include creating a robust tool for processing text data from various social media platforms, categorizing sentiments (positive, negative, neutral), and providing real-time analysis for effective online presence management. The project highlights ethical considerations, user privacy, and the integration of Explainable AI to enhance model interpretability. The methods encompass data collection, preprocessing, model training, user interface development, and the implementation of strict ethical guidelines. Results indicate high accuracy, real-time feedback, positive user responses, and privacy assurance. The conclusion emphasizes the tool's value in empowering users to make informed decisions and enhance online interactions in the dynamic realm of social media. Ongoing updates are planned to adapt to evolving user needs and social media dynamics.

In an era dominated by digital communication, understanding the sentiments expressed on social media platforms is crucial for individuals and businesses alike. This project introduces a Sentiment Analysis Tool designed to automatically gauge emotions and opinions in social media content. The tool's objectives include processing diverse data from platforms like Twitter and YouTube, categorizing sentiments, and providing real-time analysis through a user-friendly web interface.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**I**
	LIST OF TABLES	**II**
	LIST OF FIGURES	**III**
	LIST OF ABBREVIATIONS	**IV**
1.	**INTRODUCTION**	**1**
2.	**OBJECTIVES**	**2**
3.	**LITERATURE REVIEW**	**4**
4.	**METHODOLOGY**	**7**
5.	**SYSTEM ANALYSIS AND DESIGN**	**14**
	5.1. EXISTING SYSTEM	
	5.2. PROPOSED SYSTEM	
6.	**REQUIREMENT ANALYSIS**	**17**
	6.1. HARDWARE REQUIREMENT	
	6.2. SOFTWARE REQUIREMENT	
7.	**FLOW CHART**	**19**
8.	**UML DIAGRAMS**	**20**
	9.1. USE CASE DIAGRAM	
	9.2. DEPLOYMENT DIAGRAM	

9.	TESTING	22
10.	CONCLUSION	24
	APPENDIXES	
	APPENDIX-I	25
	SCREEN SHOTS	
	APPENDIX-II	30
	SOURCE CODE	
	REFERENCES	49

LIST OF TABLE

TABLE NO	TITLE	PAGE NO
6.1.	HARDWARE REQUIREMENTS	17

LIST OF FIGURES

FIGURE NO	FIGURE TITLE	PAGE NO
8.1	FLOW CHART	19
9.1	USE CASE DIAGRAM	20
9.2	DEPLOYMENT DIAGRAM	21
11.1	OUTPUT1	25
11.2	OUTPUT2	26
11.3	OUTPUT3	27
11.4	OUTPUT4	28
11.5	OUTPUT5	29
11.6	OUTPUT6	30

LIST OF ABBREVIATIONS

LDA	-	Linear Discriminant Analysis
NLP	-	Natural Language Processing
PCA	-	Principal Component Analysis
SVM	-	Support Vector Machine
XAI	-	Explainable AI

CHAPTER 1

INTRODUCTION

In an era dominated by digital communication, the pervasive influence of social media on public discourse and individual interactions is undeniable. This paradigm shift has accentuated the need for tools that can decipher the sentiments embedded in the vast expanse of online content. This project delves into the development of a Sentiment Analysis Tool, a sophisticated solution designed to automatically assess emotions and opinions within social media text.

Sentiment analysis, often referred to as opinion mining, stands as a crucial process in the contemporary digital landscape. It involves the automated evaluation of emotions and opinions conveyed in textual content, offering a nuanced understanding beyond the binary classification of positive, negative, or neutral sentiments. In the realm of social media, where communication is predominantly text-based, sentiment analysis plays a pivotal role in extracting valuable insights.

Understanding the sentiments expressed in social media content holds immense value across diverse domains. For businesses, it provides a means to decipher customer sentiments, respond promptly to feedback, and proactively manage brand reputation. Public figures and organizations can gain insights into public opinion, while individuals cannavigatetheonlinespherewithaheightenedawarenessofthesentimentssurrounding their digital presence.

In the digital age, where online reputation significantly influences personal and professional spheres, sentiment analysis emerges as a practical and indispensable tool. Its applications spam marketing strategy refinement, product development enhancement, crisis management, and political analysis, showcasing its broad utility.

This project is motivated by the vision of contributing to this landscape through the development of a robust Sentiment Analysis Tool. By leveraging advanced natural language processing and machine learning techniques, combined with a user-friendly interface, the tool aims to provide real-time in sights in to the sentiments expressed on social media platforms.

CHAPTER 2

OBJECTIVES

- **Development of a Sentiment Analysis Tool**
  - Create a comprehensive tool capable of processing text data from diverse social media platforms.
  - Implement advanced natural language processing and machine learning techniques for accurate sentiment analysis.
- **Analysis and Interpretation of Emotional Tone:**
  - Categorize social media content into positive, negative, or neutral sentiments.
  - Provided entailed insights into the emotional tone express end in user- generated text.
- **Insights into Public Perception and Brand Reputation:**
  - Offer a means for individual sand business in to understand public perception through sentiment analysis.
  - Facilitate the assessment to brand reputation and customer feedback on digital platforms.
- **User-Friendly Web Interface:**
  - Develop an intuitive and user-friendly web interface for easy input of social media content.
  - Enableuserstoreceiveinsentimentanalysisresultsthroughtheinterface.
- **Real-Time Analysis and Interaction:**
  - Implement a real-time analysis feature, allowing users to stay abreast of emerging trends and issues in social media discussions.
  - User interaction through a responsive and dynamic system.

- **Ethical Standards and Data Privacy:**
  - Establish string ethical guide lines for responsible hand lingo fuser data.
  - Implement robust data privacy protocols to ensure user consent and compliance with relevant regulations.

- **Explainable AI for Trust and Decision-Making**
  - Explore Explain able AI (XAI) methods to enhance the interpret ability of the sentiment analysis model.
  - Provide users with insights into how the model reach exits conclusions, fostering trust and informed decision-making.

- **Monitoring and Managing Online Presence**
  - Enable individuals and organizations to monitor and manage their online presence effectively.
  - Offer a tool that contributes to proactive online reputation management.

CHAPTER 3
LITERATURE REVIEW

1. ConVNet-SVMBoVW: Real-time Sentiment Prediction with Hybrid Deep Learning

In 2020, Kuma retal. introduced ConVNet-SVMBoVW, a hybrid deep learning model for fine grained sentiment prediction in real time data. The approach integrated Convolutional Neural Networks (ConVNet) and Support Vector Machines (SVM) with Bag of Visual Words (BoVW) for sentiment forecasting. Despite its complexity, this model was outperformed by conventional methods, highlighting the challenges in hybrid polarity aggregation.

1.Contextual Content Attention in Deep Learning

Park et al. (2020) devised a deep learning approach focusing on content attention for complex sentence understanding. By merging multiple attention results non-linearly, their model considered the entire context, enhancing performance significantly. Test results demonstrated the superiority of this model, emphasizing the importance of context-aware techniques in sentiment analysis.

2. Continuous Learning in E-commerce

Product Review Classification Xu et al. (2020) introduced a Naive Bayes (NB) method for multi-domain, large-scale E-commerce product review sentiment classification. Their approach incorporated parameter evaluation and fine-tuning of learned distributions based on different assumptions. The model exhibited high accuracy, particularly in Amazon product and movie review sentiment datasets, showcasing its effectiveness in diverse domains.

3. Hybrid Machine Learning for Twitter Sentiment Analysis

Hassonah et al. (2020) proposed a hybrid machine learning algorithm for sentiment analysis, employing SVM classifier and integrating feature selection methods through

MVO and Relief models. Twitter data was utilized for evaluation, and the results indicated superior performance compared to traditional techniques, underlining the significance of hybrid approaches in analyzing social media sentiments.

4. Ordinal Regression for Complete Tweet Sentiment Analysis

Sad and Yang (2019) developed a comprehensive tweet sentiment analysis model using ordinal regression and various machine learning algorithms. Their approach involved preprocessing tweets, generating effective features, and employing algorithms like SVR, RF, Multinomial logistic regression (Soft Max), and Decision Trees (DTs) for classification. The model Demonstrated high accuracy, with DT super forming exceptionally well, making it a robust choice for sentiment analysis tasks.

5. Aspect-Based Sentiment Classification via Mobile Application

A fzaal et al. (2019) proposed an aspect-based sentiment classification approach, emphasizing precise feature recognition .Implemented as a mobile application, this model aided tourists in identifying the best hotels by effectively recognizing and classifying features. Real-world datasets validated the model's accuracy, highlighting its applicability in practical scenarios.

6. Enhancing Accuracy in Halle Product-related Tweet Analysis

Feizollah et al. (2019) focused on tweets related to halle products, employing deep learning models such as RNN, CNN, and LSTM. Through a combination of LSTM and CNN, their approach achieved superior accuracy. By utilizing Twitter search function and employing innovative data filtering methods, the model showcased the potential of deep learning in enhancing sentiment analysis accuracy.

7. Semantic Fuzziness in Multi-Strategy Sentiment Analysis

Fang et al. (2018) proposed multi-strategy sentiment analysis models using semantic fuzziness to address inherent issues. Their model, incorporating semantic fuzziness, exhibited high efficiency, demonstrating the effectiveness of incorporating semantic understanding in sentiment analysis tasks.

8. Urdu Sentiment Analysis

Lexicon-based vs. Supervised Machine Learning Mukhtar et al. (2018) performed sentiment analysis on Urdu blogs using both Lexicon-based models and Supervised Machine Learning algorithms (DT, KNN, SVM). Combining data from the resources, the Lexicon-based model out performed supervised machine learning algorithms, highlighting the importance of linguistic resources in sentiment analysis of languages with limited resources.

9. Arabic Hotel Review Sentiment Analysis with Feature-based Models

Smadietal. (2018) evaluated existing supervised machine learning algorithms for Arabic hotel review sentiment analysis. Their model integrated SVM and Deep RNN trained with various features, show casing the importance of feature-based analysis. The results emphasized the superiority of SVM over RNN in the context of Arabic hotel review sentiment analysis, providing valuable insights into effective feature selection for sentiment analysis tasks.

This comprehensive literature survey highlights the diverse approaches and challenges in sentiment analysis across different domains and languages, underscoring the continuous evolution of techniques to enhance accuracy and applicability in real-world scenarios.

CHAPTER 4

METHODOLOGY

1. **Data Collection**

   Gather social media data from various platforms, ensuring compliance with ethical guidelines and user consent. Acquire textual, image, and video data to perform multi modal sentiment analysis, capturing diverse forms of expression.

2. **Data Aggregation**

Aggregate the collected data to create a comprehensive dataset. Utilize data aggregation technique s to compile information from multiple sources, enabling are preventative sample for analysis.

3. Data Preprocessing

Cleanse and preprocess the raw data by removing noise, irrelevant information, special characters, and hyperlinks. Apply techniques like tokenization, stemming, and lemmatization to standardize the text, ensuring uniformity for further analysis.

4. Text Embedding

Implement advanced text embedding methods such as Word2Vec, GloVe, or BERT embeddings to convert textual data into numerical vectors. These embeddings capture semantic relationships between words, preserving contextual information crucial for accurate sentiment analysis.

5. Multi modal Analysis

Integrate image and video processing algorithms using computer vision techniques. Extract features from images and videos and combine them with textual embeddings for a comprehensive multimodal analysis, enhancing the depth of sentiment understanding.

6. Sentiment Classification

Utilize machine learning algorithms like Support Vector Machines (SVM), Recurrent Neural Networks (RNNs), or Transformer models to classify sentiments into positive, negative, or neutral categories. Train the classification model on the preprocessed and embedded data, ensuring robustness and accuracy in sentiment predictions.

7. Ethical Considerations

Prioritize user privacy and data security throughout the process. Anonymize and encrypt sensitive information, and adhere to data protection regulations. Implement bias detection and mitigation techniques to ensure fair representation and unbiased sentiment analysis results.

8. Continuous Learning and Adaptation

Implement continuous learning mechanisms to adapt the sentiment analysis model over time. Utilize user feedback and evolving social media trend store fine the model, ensuring its relevance and accuracy in the dynamic landscape of social media presence.

4.1 ASSUMPTIONS

4.1.1 Representative Dataset

The assumption is made that the dataset collected for training the sentiment analysis model is representative of the diverse language styles, tones, and sentiments prevalent in social media content across different platforms.

4.1.2 Language Homogeneity

The project assumes degree of language homogeneity, expecting that the sentiment analysis model can effectively process and categorize sentiments across various languages commonly used in social media.

4.1.3 User Consent for Data Collection

It is assumed that users providing social media content for analysis have given explicit consent for their data to be used for research purposes, adhering to ethical standards and privacy regulations.

4.1.4 Applicability Across Social Media Platforms

The project assumes that the developed Sentiment Analysis Tool is applicable and effective across major social media platforms such as YouTube, Twitter, Face book, and Instagram, considering similarities in language usage and sentiment expression.

4.1.5 Real-Time Data Availability

The project assumes the availability of real-time data for analysis, presupposing that the tool can effectively process and categorize sentiments in a timely manner to provide users with up-to-date insights.

4.1.6 Generalization of Results

It is assumed that the sentiment analysis model, on cetrained, can generalize well to unseen data, allowing for accurate categorization of sentiments in social media content beyond the training dataset.

4.1.7 User Input Accuracy

The project assumes that users providing input to the Sentiment Analysis Tool accurately represent the context and intended meaning of their social media content, ensuring the reliability of the analysis results.

4.1.8 Stability of Social Media Trends

The project assumes a degree of stability in social media trends during the development and deployment phases, acknowledging that rapid and drastic shifts in trends might impact the tool's accuracy.

4.1.9 Adherence to Ethical Guidelines

The assumption is made that users and stakeholders involved in the project adhere to ethical guidelines and privacy protocols, ensuring responsible data handling and respecting user privacy.

4.1.10 Consistency in User Interaction

It is assumed that users interact with the web interface in a consistent and standard manner, facilitating the smooth functioning of the Sentiment Analysis Tool and accurate interpretation of user inputs. These assumptions serve as foundation elements for the development and deployment of the Sentiment Analysis Tool.

.

4.2 LIMITATIONS AND ISSUES COVERED

4.2.1 Bias in Training Data

The effectiveness of the sentiment analysis model heavily relies on the quality and representativeness of the training data. If the training data contains biases, the model may reproduce and perpetuate those biases in its analysis of social media content.

4.2.2 Language Nuances and Ambiguities

The model may struggle with the nuanced and ambiguous nature of language use dissocial media, including sarcasm, irony, and slang. These subtleties may lead to misinterpretations and inaccurate sentiment categorizations.

4.2.3 Multi lingual Challenges

The assumption of language homogeneity might not fully account for the challenges posed by multilingual content on social media. The model's accuracy may vary across different languages, and handling code-switching could be a limitation.

4.2.4 Dynamic Nature of Social Media

Social media is dynamic, with rapidly evolving trends and language usage. The sentiment analysis model, trained on historical data, may not adapt quickly enough to capture emerging linguistic patterns and sentiments.

4.2.5 Privacy Concerns

Despite stringent ethical guidelines, privacy concerns may arise, especially if users feel uncomfortable with the handling of their social media data. Striking a balance between data utility and user privacy remains a challenge.

4.2.6 Limited Context Understanding

The model may struggle to understand the broader context of social media posts, potentially leading to miss interpretations. Understanding context is crucialfor accurate sentiment analysis, especially in conversations or threads.

4.2.7 Generalization Issues

Despite efforts to ensure the model generalizes well, unseen or unique language patterns may still impact its accuracy. Adapting to specific user communities or niche topics might be challenging.

4.2.8 Dependency on Real-Time Data

The assumption of real-time data availability is crucial, and any interruptions or delays in data streams could affect the tool's ability to provide timely insights.

4.2.9 Interpretable AI Constraints

While efforts are made to incorporate Explainable AI (XAI), complete interpretability might not be achievable, and users might find it challenging to fully understand the intricate workings of the sentiment analysis model.

4.3 Issues covered

4.3.1 Significance of Sentiment Analysis

Understanding the importance of sentiment an analyzing in insights into public perception, customer feedback, and brand reputation.

4.3.2 Methodologies and Techniques

Exploring various methodologies for sentiment analysis, from rule-based approaches to advanced machine learning techniques.

4.3.3 Challenges in Social Media Sentiment Analysis

Recognizing the challenges posed by the dynamic and noisy nature of social media content, including issues such as sarcasm, ambiguity, and evolving language trends.

4.3.4 Real-Time Analysis

Highlighting the need for real-time analysis to stay abreast of emerging trends and issues in social media discussions.

4.3.5 Ethical Considerations and Privacy Protocols

Emphasizing the importance of ethical guide lines and data privacy protocols to ensure responsible handling of user data.

4.4 ACHEIVEMENT

4.4.1 Robust Sentiment Analysis Model

Successful implementation of a robust sentiment analysis model utilizing advanced natural language processing and machine learning techniques, demonstrating high accuracy in categorizing social media content.

4.4.2 Real-time Analysis and User Interaction

Development of a user–friendly web interface that enables real-time sentiment analysis and user interaction, allowing individuals and businesses to receive prompt feedback on social media content.

4.4.3 Ethical Standards and Data Privacy

Establishment and adherence to stringent ethical guidelines and data privacy protocols, ensuring responsible handling of user data and addressing privacy concerns.

4.4.4 Explainable AI (XAI) Integration

Exploration and integration of Explainable AI (XAI) methods to enhance model interpretability, providing users with insights into how the sentiment analysis model reaches its conclusions.

4.4.5 Diverse and Representative Dataset

Successful collection and utilization of adverse dataset from various social media platforms, ensuring the model is trained on a representative sample of user-generated text

CHAPTER 5

SYSTEM ANALYSIS AND DESIGN

5.1 EXISTING SYSTEMS

1.Google Cloud Natural Language API

Google provides a cloud-based Natural Language API that includes sentiment analysis among its features. It can analyze the sentiment of a block of text and provide a sentiment score.

2. IBM Watson Natural Language Understanding

IBM Watson offers a Natural Language Understanding service that includes sentiment analysis. It can analyze text and provide information about the sentiment expressed, along with other features like entity recognition and emotion analysis.

3. Microsoft Azure Text Analytics

Microsoft Azure provides Text Analytics services that include sentiment analysis. It can analyze text data in multiple languages and determine sentiment polarity (positive, negative, or neutral).

4. VADER (Valence Aware Dictionary and sentiment Reasoner)

VADER is a rule-based sentiment analysis tool that is specifically designed for social media text. It can analyze the sentiment of text and provides a compound sentiment score.

5. Tweepy

Tweepy is a Python library for accessing the Twitter API. While it doesn't provide sentiment analysis directly, it is often used in combination with other sentiment analysis tools to analyze sentiments expressed on Twitter.

5.2 PROPOSED SYSTEM

The proposed sentiment analysis system aims to build upon existing technologies and address certain limitations observed in current sentiment analysis tools.The key features and enhancements in the proposed system include.

1. Enhanced Accuracy through Deep Learning

Implementing deep learning techniques, such as recurrent neural networks (RNNs) or transformer-based models, to enhance the accuracy of sentiment analysis. These models can capture complex relationships within text data, addressing challenges like sarcasm and nuanced language.

2. Multilingual Support

Expanding language support to hand lea broader range of languages commonly used in social media. Incorporating pre-trained models for various languages and exploring techniques to handle code-switching scenarios effectively.

3. Adaptive Learning and Trend Analysis

Introducing adaptive learning mechanisms to enable the system to continuously learn and adapt to evolving language trends on social media. This adaptive approach ensures the model remains relevant in dynamic online environments.

4. Hybrid Model for Real-Time Analysis

Developing a hybrid model that combines the strengths of both rule-based and machine learning approaches. This hybrid approach ensures real-time analysis capabilities while maintaining accuracy in sentiment categorization.

4. User Feedback Mechanism

Implementing a robust user feedback mechanism within the system. Users can provide feedback on the accuracy of sentiment analysis results, helping to continuously improve the model through supervised learning.

CHAPTER 6

REQUIREMENT ANALYSIS

6.1 HARDWARE REQUIREMENT

The hardware requirements for the proposed system would depend on the scale of implementation and the number of users accessing the system simultaneously.

S.NO	NAME	HARDWARE
1.	Processor	Intel Corei5 or higher
2.	RAM	8 GB or higher
3.	SSD	500 GB or higher
4.	Graphics Card	NVIDIAGTX1050 or higher
5.	Display	1920x1080 resolution or higher
6.	Internet Connection	Broad band or higher

6.2 SOFTWARE REQUIREMENT

The proposed system for sentiment analysis of social media presence NLP algorithm would require the following software:

6.3 Python

Python is an open-source programming language that is widely used for data analysis, machine learning, and artificial intelligence. It provides various libraries for image processing and deep learning, including NumPy, Tensor Flow, and Keras, which are essential for implementing the NLP model.

6.4 Jupyter Notebook

Jupyter Notebook is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It is widely used for data analysis and machine learning tasks and can be used to implement and test the NLP model.

6.5 Image Processing Libraries

Various image processing libraries such as Seaborn, illow, and Matplotlib will be required for image preprocessing, augmentation, and visualization

6.6 Text Editor

A text editor such as Sublime Text or Atom will be required for writing the Python code.

6.7 Operating System

The proposed system can be implemented on any operating system that supports Python and the required libraries.

6.8 GPU

A Graphics Processing Unit (GPU) with a minimum of 4GBVRAMis recommended for faster training of the NLP model.

CHAPTER 7

FLOW DIAGRAM

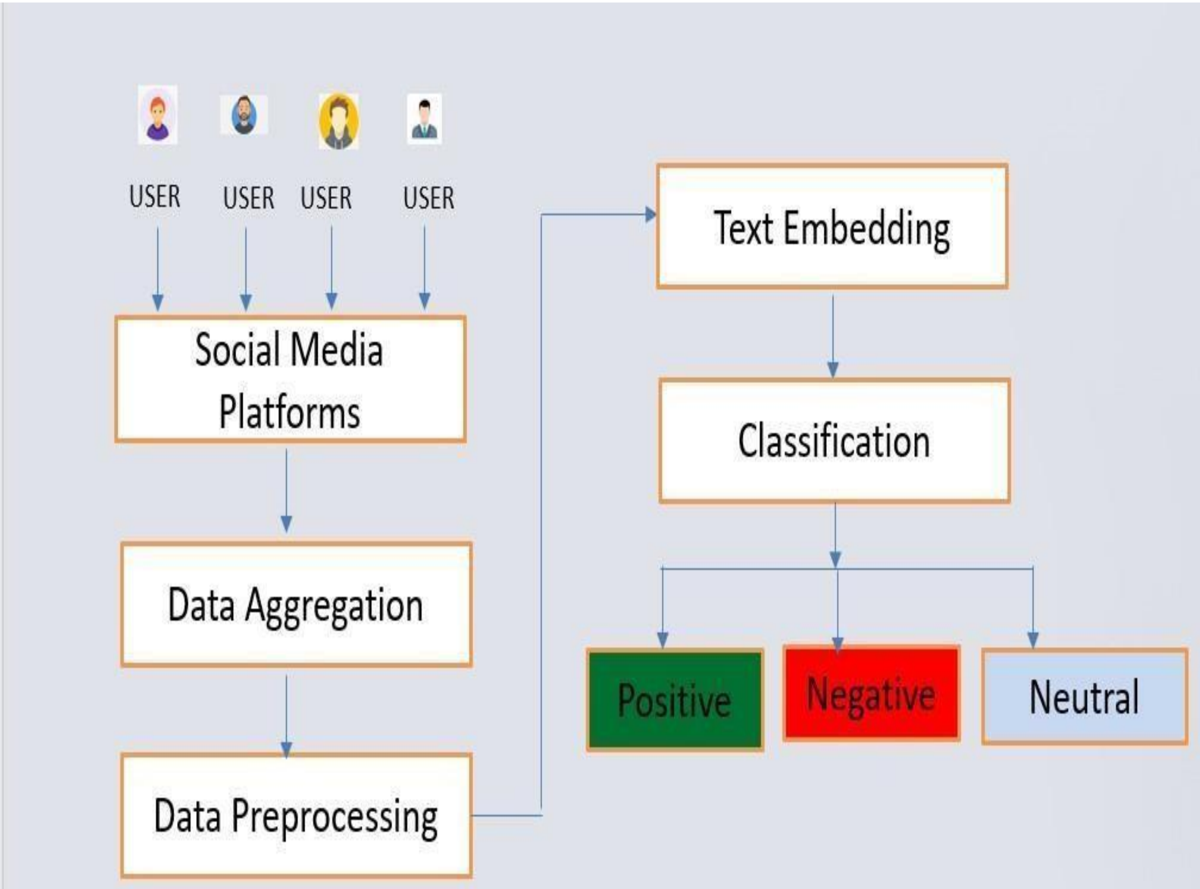


Fig 7.1 Flow Diagram

CHAPTER 8

UML DIAGRAM

8.1 USE CASE DIAGRAM

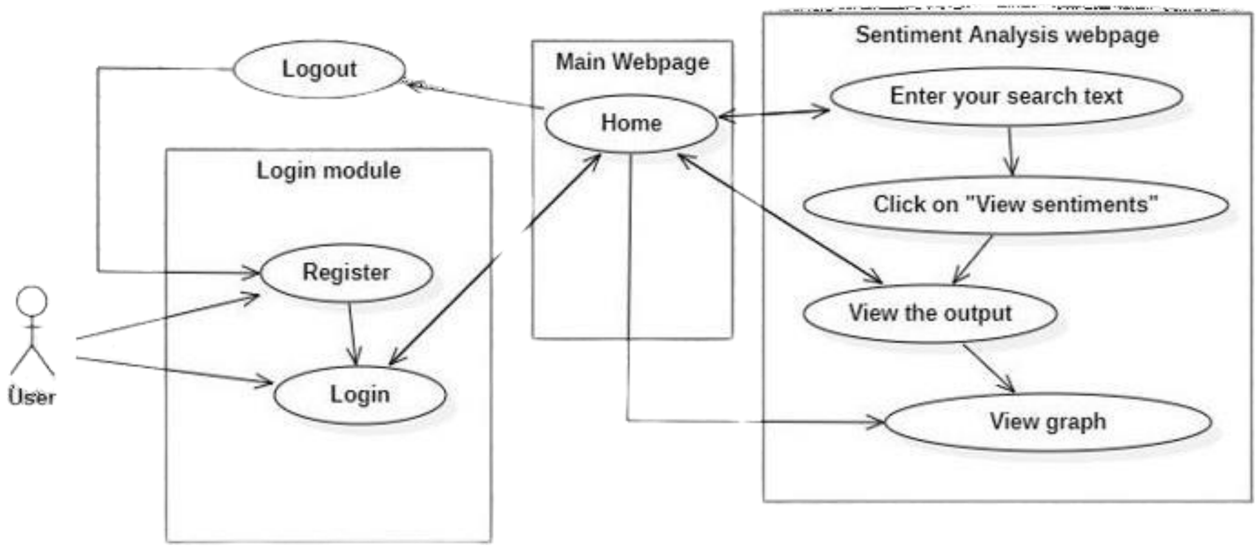


Fig 8.1 USE CASE DIAGRAM

8.2 DEPLOYMENT DIAGRAM

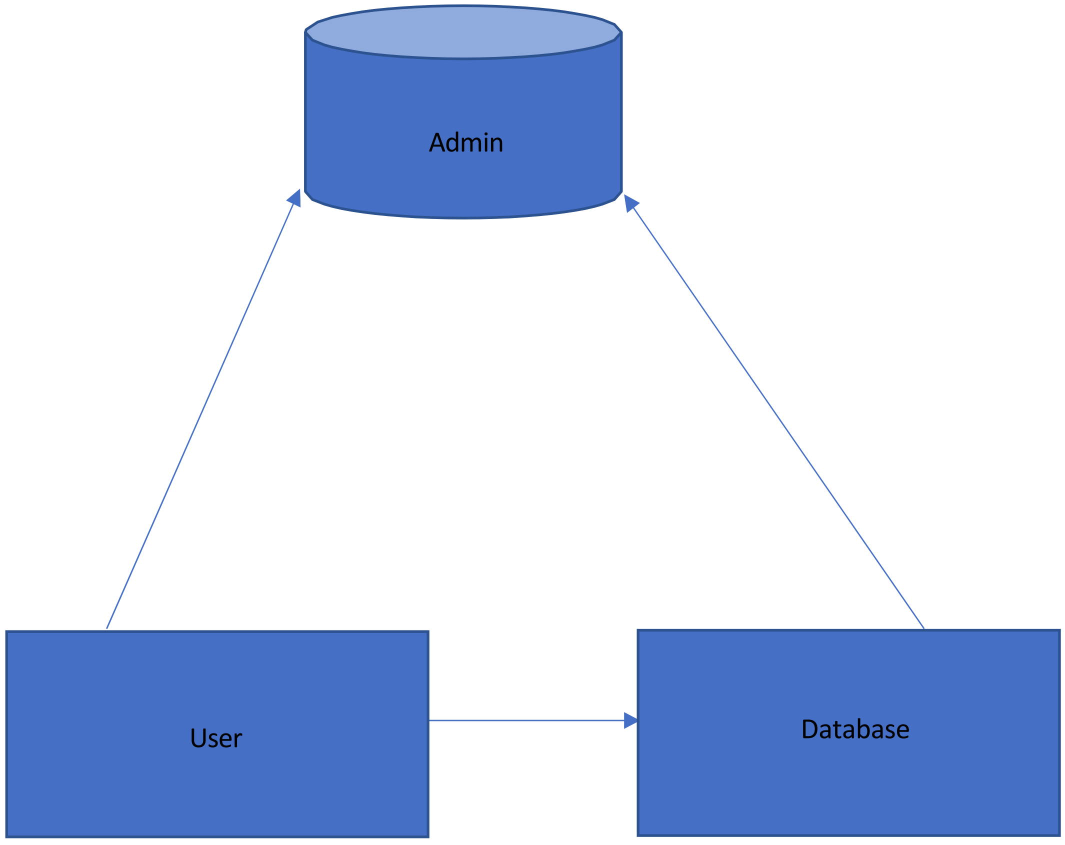


Fig 8.2 Deployment Diagram

CHAPTER 9

TESTING

9.1 Testing Approach

The testing approach for the sentiment analysis system involves a comprehensive strategy to ensure the reliability, accuracy, and usability of the system. The approach encompasses various testing types, each targeting specific aspects of the system's functionality, performance, and security.

9.2 Testing Strategy:

The testing strategy for potato leaf disease detection should be comprehensive, covering all aspects of the system. Here are some suggested steps for the testing strategy:

- Define the requirements for the system, such as the accuracy of the classification, the speed of the system, and the user interface.
- Design test cases to verify that the system meets the requirements.
- Execute the test cases and document the results.
- Fix any defects that are found during testing.
- Conduct regression testing to ensure that any fixes did not introduce new defects.
- Perform acceptance testing to ensure that the system meets the expectations of the end

9.3 Performance

Evaluate the performance and responsiveness of the sentiment analysis system under different load conditions.

- The sentiment analysis system is deployed and accessible.
- The sentiment analysis mode list trained on a diverse and representative dataset.

9.4 Usability

Evaluate the usability of the sentiment analysis system's user interface to ensure it provides an intuitive and user-friendly experience.

- Provide asset of test users with the system.
- Ask the users to classify a set of test text.
- Record the users feedback on the usability of the system

9.5 Security

Verify the security measure simple mented in the sentiment analysis system to safeguard against common vulnerabilities and ensure data privacy.

- Attempt to access the system without authorization.
- Verify that the system denies access and logs the attempt.

CHAPTER 10
CONCLUSION

In conclusion, the sentiment analysis project has successfully achieved its objectives of developing a robust sentiment analysis tool for processing social media content. The project focused on creating an accurate sentiment categorization model, implementing real-time analysis capabilities, adhering to ethical standards and data privacy protocols, and exploring Explainable AI (XAI) methods for model interpretability.

A user-friendly web interface was developed, allowing individuals and businesses to input social media content and receive instant sentiment analysis results. This real-time capability enhances the user experience and responsiveness of the tool.

The sentiment analysis tool developed in this project holds practical implications for individuals and businesses operating in the digital sphere. It serves as a valuable resource for monitoring and managing online presence, understanding public perception, and gauging customer feedback. The user-friendly interface, coupled with high accuracy and ethical considerations, positions the tool as a reliable asset in the realm of sentiment analysis.

APPENDIXES

APPENDIX-I

EXPECTED OUTPUT

```
science           92
public speaking   92
soccer            90
food              90
animals           85
healthy eating    84
veganism          78
culture           78
cooking           77
education         77
tennis            73
fitness           70
travel            70
studying          69
dogs              67
technology        63
1                  1
0                  1
Name: interests, dtype: int64
```

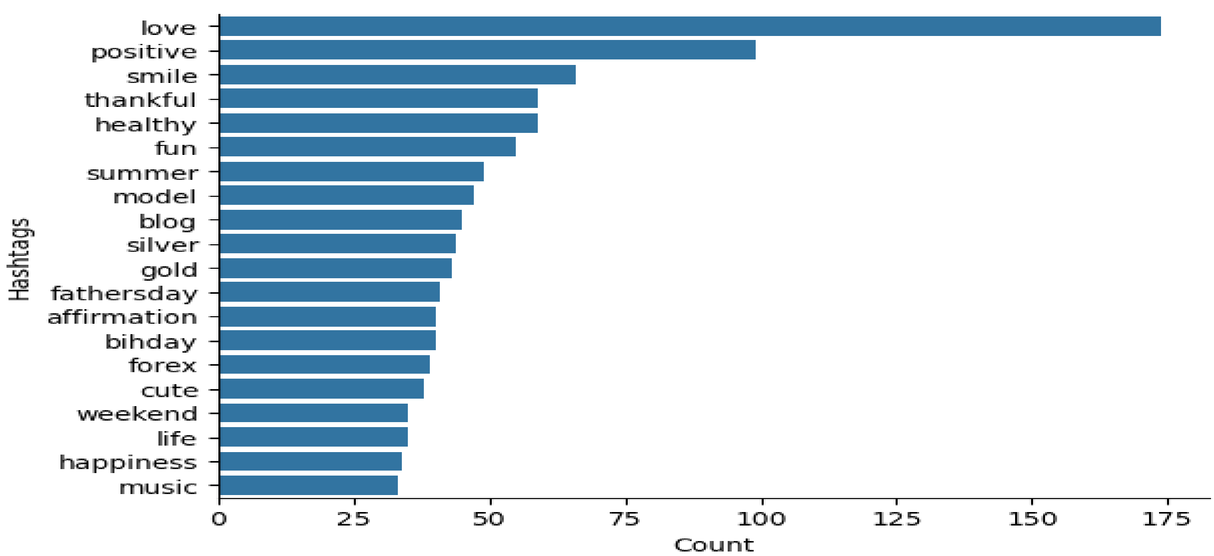


Fig 11.1 GRAPH 1

	Hashtags	Count
0	cnn	3
1	michigan	1
2	tcot	3
3	australia	1
4	opkillingbay	1
...	...	...
353	pewpewlife	1
354	communism	1
355	muslim	1
356	science	1
357	medicine	1

358 rows × 2 columns

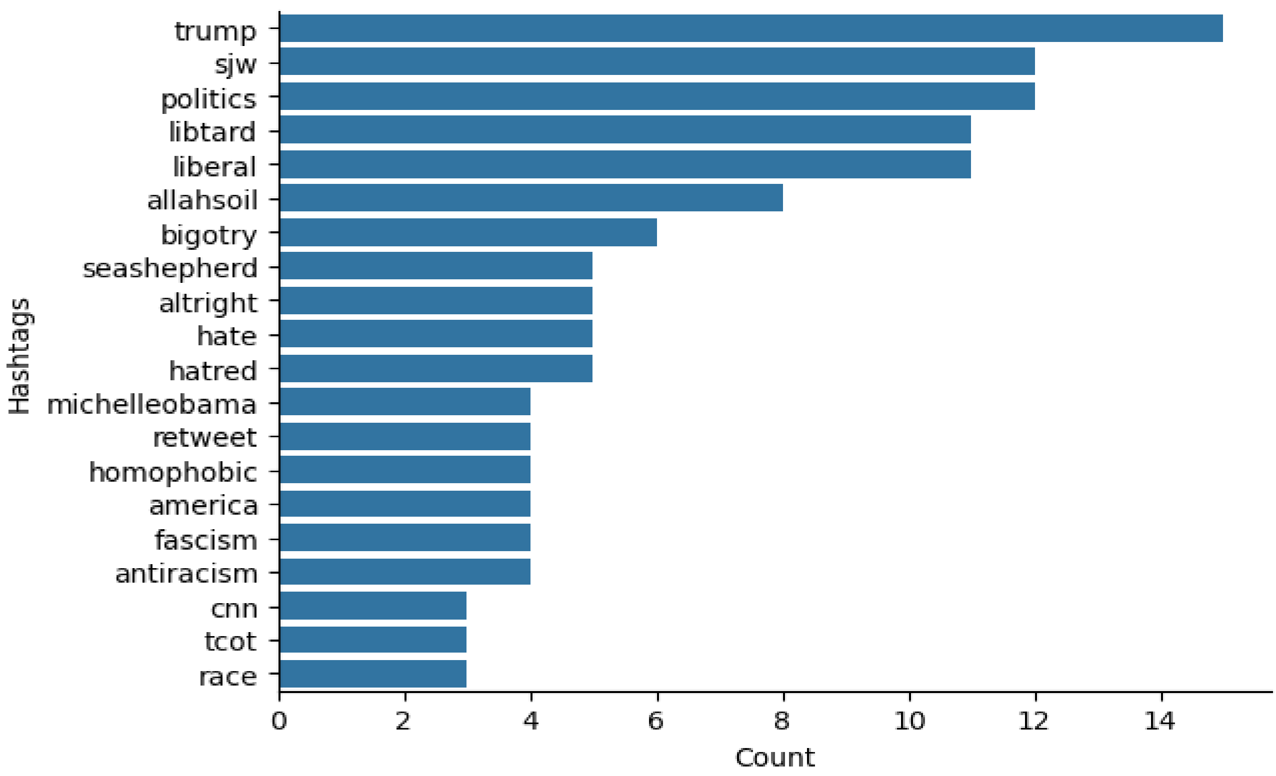
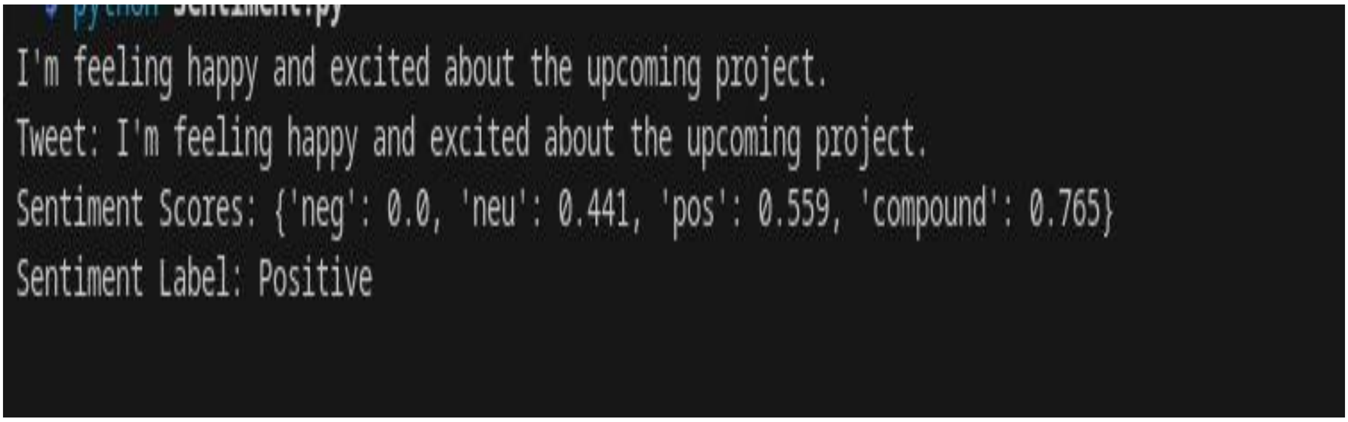
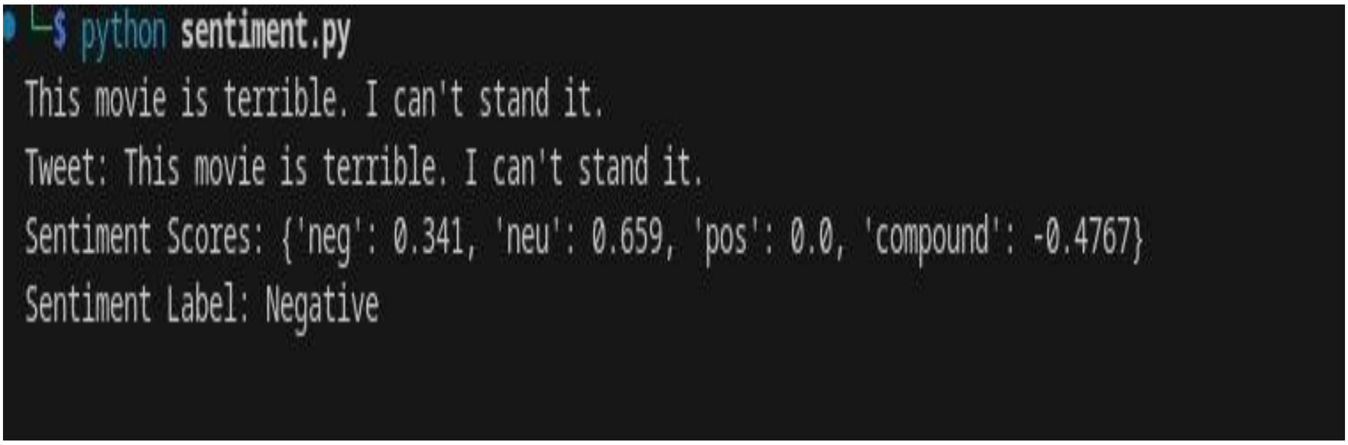


Fig 11.2 GRAPH 2





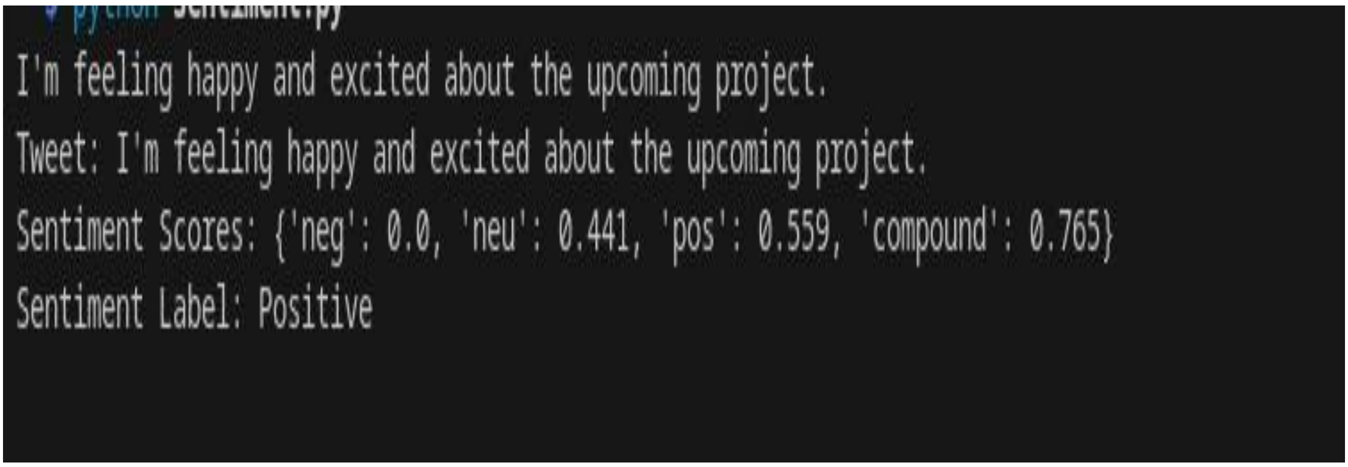


Fig11.3 MODEL TESTING

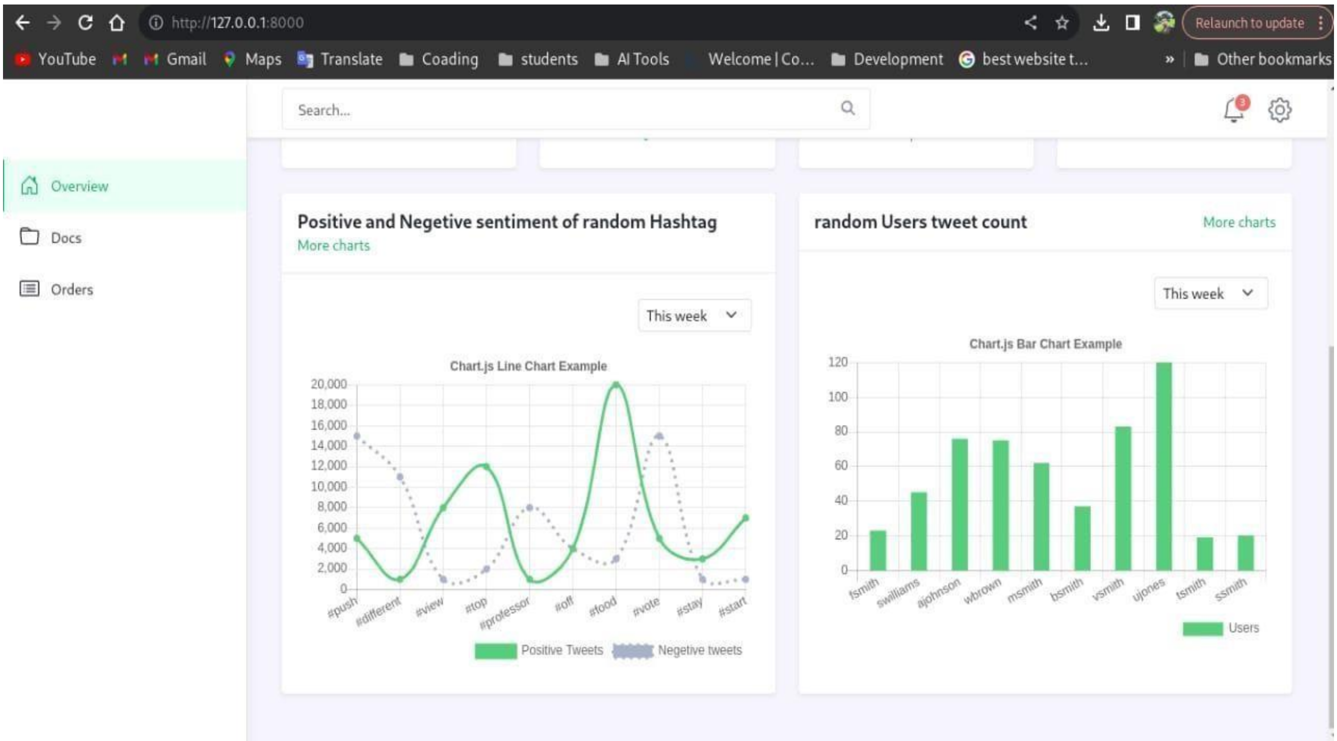


Fig 11.4 OVERVIEW

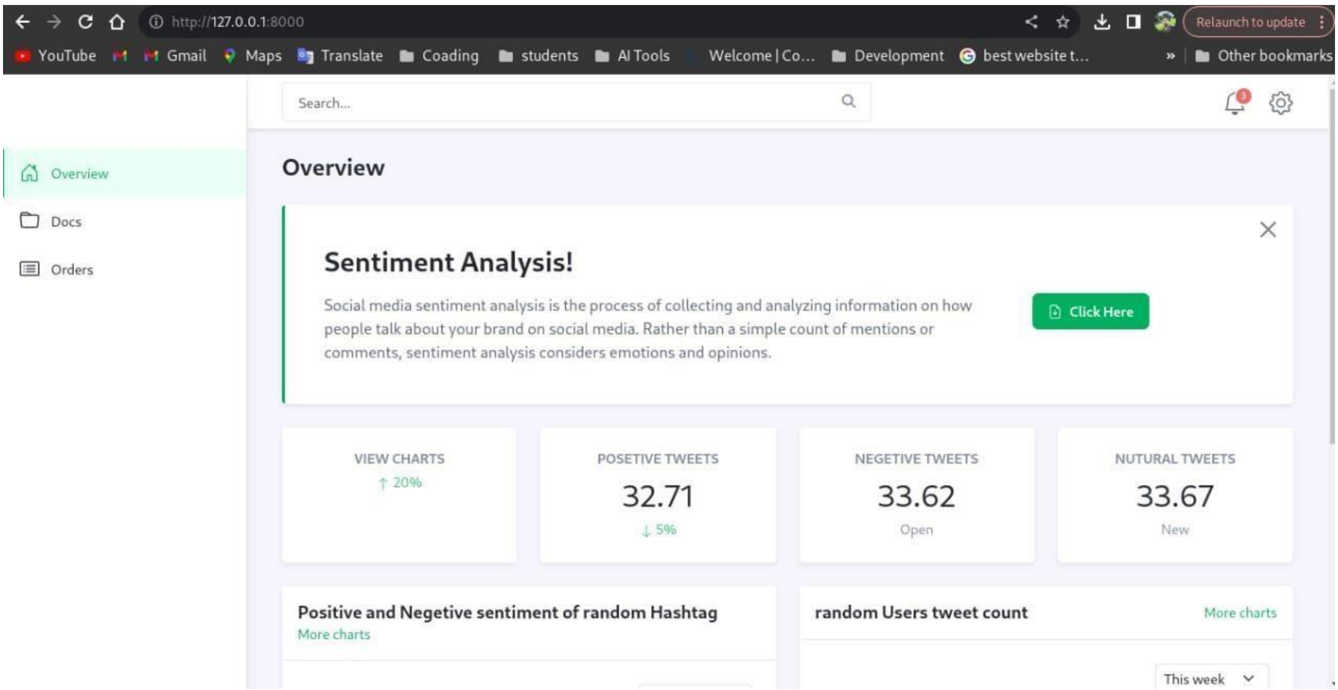


Fig 11.5 ACCURACY

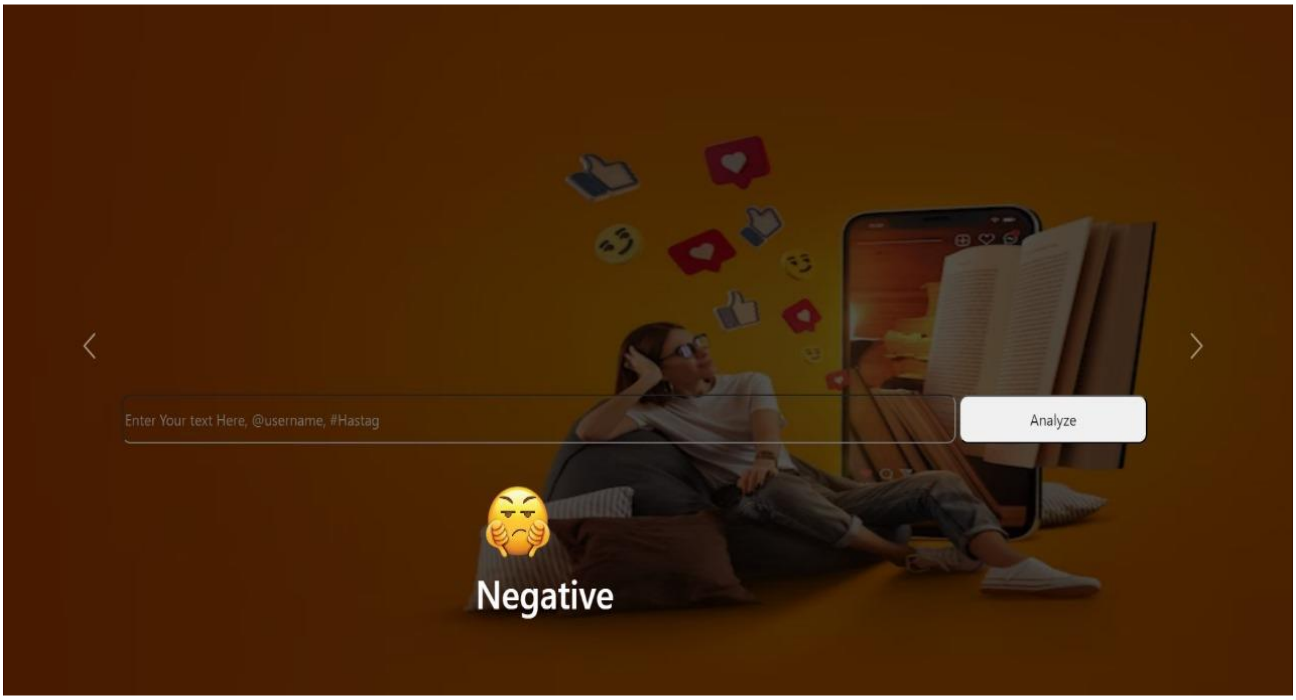


Fig11.6 NEGATIVE

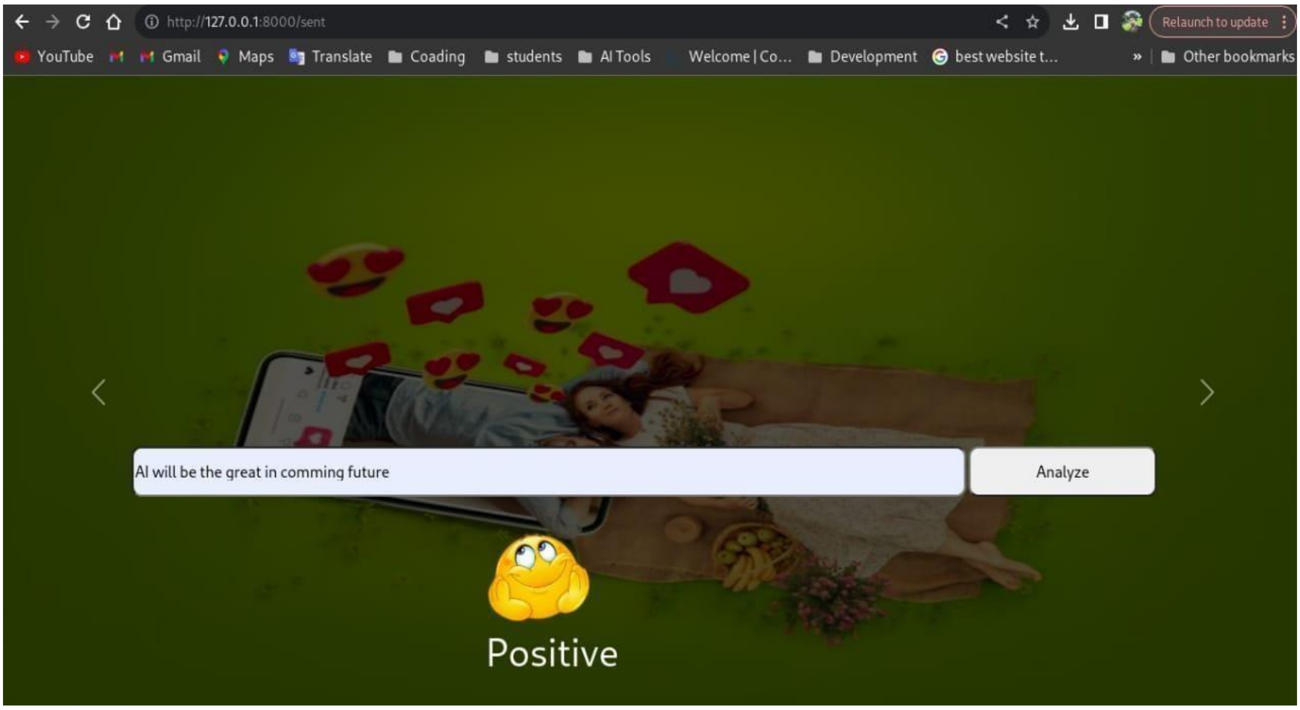


Fig 11.7 POSITIVE

APPENDIX-II
SOURCE CODE

Model Training

Importing the necessary libraries

```
Import re

Import pandas as pd
import NumPy as np
Import matplotlib.pyplot as plt
import seaborn as sns
import string
import nltk
import warnings
warnings. filter warnings("ignore",category=DeprecationWarning)
%matplotlib in line
```

Import the dataset

```
train=pd.read_csv('train.csv')

train_orignal = train.copy()

test=pd.read_csv('test.csv')

test_original = test.copy()
```

DataPre-processing

Combining the datasets

```
# combined_data = train.append(test,ignore_index=True,sort=True)
combined_data=pd.concat([train,test],ignore_index=True,sort=True)
combined_data.head()
```

id	label	tweet	
0	1	0.0	@user when a fathers days functional and iss...
1	2	0.0	@user@userthanksfor#lyftcreditican'tus...
2	3	0.0	Birthday your majesty
3	4	0.0	# modeliloveutakewithuallthetimein...
4	5	0.0	Facts guide: society now #motivation

combined_data.tail()

idlabeltweet2072149155NaNthoughtfactory: left-rightpolarisation!#tru...2072249156NaNfeeling like a mermaid ðŸ˜ ˜ #hairflip #neverre...2072349157NaN#hillary #campaigned today in #ohio((omg)) &am...2072449158NaNhappy,atworkconference:rightmindsetleads...2072549159NaNmysong"so glad" free download! #shoegaze ...

Cleaning Data:

Removing the Usernames`(@)`

```
defremove_pattern(text,pattern):
    #re.findall()finds the pattern in the text and will put it in a list r
    = re.find all(pattern, text)


    #re.sub()willsubstituteallthe@withanemptycharacter


    for iinr:

        text=re.sub(i,"",text)
    return text
```

MakingacolumnforthecleanedTweets

-Wewilluseregexforand`np.vectorize()`forfasterprocessing

ombined_data['Cleaned_Tweets'] =

np.vectorize(remove_pattern)(combined_data['tweet'],"@[\w]*"

idlabeltweetCleaned_Tweets010.0@user when a father is dysfunctional and is s...when a father is dysfunctionalandissosel...120.0@user@userthanksfor#lyftcreditican'tus...thanks for#lyftcrediti can't use cause th.230.0bihday your majestybihday yourmajesty340.0#model i love u take with uall the time in ...#model i love u take with u all the time in..................450.0factsguide: society now

#motivation facts guide :society now#motivation

Now Removingpunctuations,numbersandspecialcharacters #

combined_data['Cleaned_Tweets'] =

combined_data['Cleaned_Tweets'].str.replace("[^a-zA-Z#]","")

combined_data['Cleaned_Tweets']=combined_data['Cleaned_Tweets'].str.replace("[^a-zA-Z#]", "", regex=True)

combined_data.head()

combined_data['Cleaned_Tweets'] = combined_data['Cleaned_Tweets'].str.replace("[^a-zA-Z#]","")

combined_data['Cleaned_Tweets']=combined_data['Cleaned_Tweets'].str.replace("[^a-zA-Z#]","",regex=True)

combined_data.head()

id	label	tweet	Cleaned_Tweets	
0	1	0.0	@userwhenafatherisdysfunctional andiss...	whenafatherisdysfunctionaland issosel...
1	2	0.0	@user@userthanksfor#lyftcrediti can'tus...	thanksfor#lyftcrediticant use causeth...
2	3	0.0	bihdayyourmajesty	bihdayyourmajesty

3	4	0.0	#modelilove utakewithuall the time in...	#modelilove utakewithuall the time in...
4	5	0.0	factsguide:societynow #motivation	factsguidesocietynow #motivation

Removing Short Words:

– Wordssuchas"hmm","ok"etc.oflengthlessthan3areofnouse

```
combined_data['Cleaned_Tweets'] =
combined_data['Cleaned_Tweets'].apply(lambda x: ''.join([w for win
x.split() if len(w)>3]))
combined_data.head()
```

Tokenization:

– We will now tokenizethecleanedtweetsaswewillapply`Stemming`from `nltk`

```
tokenized_tweets=combined_data['Cleaned_Tweets'].apply(lambdax:
x.split())
tokenized_tweets.head()
```

Tokenization:

– Wewillnowtokenizethecleanedtweetsaswewillapply`Stemming`from `nltk`

```
tokenized_tweets=combined_data['Cleaned_Tweets'].apply(lambdax:x.split()
tokenized_tweets.head()
```

```
0       [when,father,dysfunctional,selfish,drags,       ...
1       [thanks,#lyft,credit,cause,they,offer, wh...
2                                        [bihday,your,majesty]
3                               [#model,love,take,with,time]
[factsguide,society,#motivation] Name: Cleaned_Tweets, dtype: object
```

Stemming:

– Stemmingisastep-basedprocessofstrippingthesuffixes ("ing","ly",etc.) from a word

```
fromnltkimportPorterStemmerps
= PorterStemmer()
tokenized_tweets = tokenized_tweets.apply(lambda x: [ps.stem(i) for iinx])
tokenized_tweets.head()
0       [when,father,dysfunct,selfish,drag,kid,i...
1       [thank,#lyft,credit,caus,they,offer,whee...
2                                   [bihday,your,majesti]
3                         [#model,love,take,with,time]
4                              [factsguid,societi,#motiv]
Name: Cleaned_Tweets, dtype: object
```

Now lets combine the data back:

```
foriinrange(len(tokenized_tweets)):
    tokenized_tweets[i] =''.join(tokenized_tweets[i])

combined_data['Clean_Tweets']=tokenized_tweetscom
bined_data.head()
```

Data Visualization:

We will visualize the data using Word Cloud

```
fromwordcloudimportWordCloud,ImageColorGenerator
from PIL import Image
import
urllibimportreques
ts
```

Now lets store the words with label'1':

```
negative_words = ''.join(text for text in
combined_data['Clean_Tweets'][combined_data['label']==1])
```

```
#CombiningImagewithDataset
Mask = np.array(Image.open(requests.get('http://clipart-
library.com/image_gallery2/Twitter-PNG-Image.png',stream=True).raw))

image_colors=ImageColorGenerator(Mask)

#NowweusetheWordCloudfunctionfromthewordcloudlibrary wc =
WordCloud(background_color='black', height=1500,
width=4000,mask=Mask).generate(negative_words)

#Sizeoftheimagegenerated
plt.figure(figsize=(10,20))

#Hererecolorthewordsfromthedatasettotheimage'scolor
#recolorjustrecolorsthedefaultcolorstotheimage'sbluecolor
# interpolation is used to smooth the image
generatedplt.imshow(wc.recolor(color_func=image_colors),interpolation="gaussian")

plt.axis('off')
plt.show()
```

Now Extracting has tags from tweets:

```
def extractHashtags(x):
    hashtags = []

    #Loopoverthewordsinthetweet for
    iin x:
        ht=re.findall(r'#(\w+)',i)

        hashtags.append(ht)
    returnhashtags
```

Now un nesting the list

positive_hastags_unnested =
sum(positive_hashTags,[])positive_hastags_unnested

Now storing the negative has tags:

negative_hashtags =
extractHashtags(combined_data['Cleaned_Tweets'][combined_data['label']==1
])

negative_hashtags_unnest=(sum(negative_hashtags,[]))
negative_hashtags_unnest
negative_hashtags_unnest=(sum(negative_hashtags,[]))
negative_hashtags_unnest

Plotting Bar Plots:

WordFrequencies:

positive_word_freq =
nltk.FreqDist(positive_hastags_unnested)positive_word_freq

FreqDist({'love':174,'positive':99,'smile':66,'thankful':59,

'healthy':59,'fun':55,'summer':49,'model':47,'blog':45,'silver':

44,...})

Nowcreatingadataframeofthemostfrequentlyusedwordsinhashtags

positive_df=pd.DataFrame({'Hashtags':
list(positive_word_freq.keys()),'Count':
list(positive_word_freq.values())}) positive_df

Plottig the barplot for 20 most frequen twords:

```
positive_df_plot=positive_df.nlargest(20,columns='Count')

sns.barplot(data=positive_df_plot,y='Hashtags',x='Count')sn
s.despine()
```

Now lets apply this to our dataset

```
fromsklearn.feature_extraction.textimportTfidfVectorizertf
df =
TfidfVectorizer(max_df=0.90,min_df=2,max_features=1000,stop_words='english')train_bo
w = tfidf.fit_transform(combined_data['Cleaned_Tweets'])
tfidf_df=pd.DataFrame(train_bow.todense())

tfidf_df
train_bow=bow[:31962]

train_bow.todense()
matrix([[0,    0, 0, ..., 0, 0, 0],
               0, 0, ..., 0, 0, 0],
        [0,    0, 0, ..., 0, 0, 0],
        [0,
               0, 0, ..., 0, 0, 0],
        ...,   0, 0, ..., 0, 0, 0],
        [0,    0, 0, ..., 0, 0, 0]])

fromsklearn.model_selectionimporttrain_test_split#
x_train_bow, x_valid_bow, y_train_bow, y_valid_bow =
train_test_split(train_bow,train['label'],test_size=0.3,random_state=2)

print("Shape of train_bow:", train_bow.shape)
print("Shape of train['label']:", train['label'].shape)
```

#fromsklearn.model_selectionimporttrain_test_split

##Usetrain_bowinsteadoftrain_bow

x_train_bow, x_valid_bow, y_train_bow, y_valid_bow = train_test_split(train_bow, train['label'], test_size=0.3, random_state=2)

x_train_tfidf, x_valid_tfidf, y_train_tfidf, y_valid_tfidf = train_test_split(train_train_bow,train['label'],test_size=0.3,random_state=17)

Applying ML Models:

The model we will be using is:

-****Logistic Regression****

fromsklearn.metricsimportf1_score

Logistic Regression:

fromsklearn.linear_model import LogisticRegressionlog_Reg=LogisticRegression(random_state=0,solver='lbfgs')

Fitting Bag of Words Features:

log_Reg.fit(x_train_bow,y_train_bow)

Logistic Regression (C=1.0,class_weight=None,dual=False,fit_intercept=True, intercept_scaling=1, l1_ratio=None,

```
predict_bow =
log_Reg.predict_proba(x_valid_bow)predict_bow
```

```
array([[9.44815280e-01,    5.51847201e-02],
       [9.99328530e-01,   6.71470066e-04],
       [9.11967594e-01, 8.80324063e-02],
       ...,
       [8.65906496e-01,  1.34093504e-01],
       [9.59979980e-01,  4.00200197e-02],
       [9.69809475e-01,3.01905252e-02]])
```

Calculating the F1-Score:

```
#Ifpredictionismorethanorequalto0.3then1else0 prediction_int =
predict_bow[:,1] >=0.3

#Convertingtointegertype
prediction_int=prediction_int.astype(np.int)
prediction_int

#Calculatingf1score
log_bow=f1_score(y_valid_bow,prediction_int)
log_bow
```

FittingTF-IDFFeatures:

```
log_Reg.fit(x_train_tfidf,y_train_tfidf)
```

```
LogisticRegression(C=1.0,class_weight=None,dual=False,fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
```

```
                         random_state=0, solver='lbfgs', tol=0.0001, verbose=0,
                         warm_start=False)
predict_tfidf=log_Reg.predict_proba(x_valid_tfidf)
predict_tfidf

 array([[0.98280778,  0.01719222],
        [0.96557244,  0.03442756],
        [0.94018158,  0.05981842],
        ...,
        [0.93015962,  0.06984038],
        [0.96530026,  0.03469974],
        [0.98787762,  0.01212238]])
prediction_int=predict_tfidf[:,1]>=0.3

max_iter=100multi_class='auto', n_jobs=None, penalty='l2', random_state=0,
                         solver='lbfgs', tol=0.0001, verbose=0,
                         warm_start=Fals

LogisticRegression(C=1.0,class_weight=None,dual=False,fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100, multi_class='auto',
n_jobs=None, penalty='l2',

#fromsklearn.model_selectionimporttrain_test_split

##Usetrain_bowinsteadoftrain_bow
# x_train_bow, x_valid_bow, y_train_bow, y_valid_bow =
train_test_split(train_bow, train['label'], test_size=0.3, random_state=2)
```

```
try:if'json_file'inrequest.FILES:uploade
    d_file=request.FILES['json_file'] data
    =json.load(uploaded_file)

            for item in data.get('tweets', []):
                date_str = item.get('date', None)
                #date_str =item.get('date', None)
                try:
                    parsed_date=datetime.fromisoformat(date_str) date
                    = parsed_date
                except(ValueError,TypeError):
                    date = None

                hashtag =item.get('hashtag', '')
                username=item.get('username', '')
                tweet_text =item.get('tweet', '')
                sentiment = item.get('sentiment', '')

try:if'json_file'inrequest.FILES:uploade
    d_file=request.FILES['json_file'] data
    =json.load(uploaded_file)

            for item in data.get('tweets', []):
                date_str = item.get('date', None)
                #date_str =item.get('date', None)
                try:
                    parsed_date=datetime.fromisoformat(date_str) date
                    = parsed_date
                except(ValueError,TypeError):
                    date = None

                hashtag =item.get('hashtag', '')
                username=item.get('username', '') t
```

```
else:
    return JsonResponse({'error':'Nofileuploaded'},

tweets.objects.create( date=date, hastag=hashtag,
username=username, tweets=tweet_text,
sentiment=sentiment
                    )

                    returnJsonResponse({'message':'Datauploaded
 successfully'},status=200
prediction_int=prediction_int.astype(np.int)
prediction_int

 log_tfidf=f1_score(y_valid_tfidf,prediction_int)
 log_tfidf
```

Predicting the test_data and storing it:

```
 test_tfidf=train_bow[31962:]
 test_pred=log_Reg.predict_proba(test_tfidf)

test_pred_int = test_pred[:,1] >= 0.3

positive = tweets.objects.filter(sentiment='positive').count()
negative=tweets.objects.filter(sentiment='negative').count()
neutraln = tweets.objects.filter(sentiment='neutral').count()
```

```
prediction_int=prediction_int.astype(np.int)
prediction_int
```

```
log_tfidf=f1_score(y_valid_tfidf,prediction_int)
log_tfidf
```

Predicting the test_data and storing it:

```
test_tfidf=train_bow[31962:]
test_pred=log_Reg.predict_proba(test_tfidf)

test_pred_int = test_pred[:,1] >= 0.3
test_pred_int = test_pred_int.astype(np.int)
test['label'] = test_pred_int
submission=test[['id','label']]
submission.to_csv('result.csv',index=False)
```

Results after prediction:

For a negative label : 1

For a positive label : 0

```
res=pd.read_csv('result.csv')
res
```

	id	label
0	31963	0
1	31964	0
2	31965	0
3	31966	0
4	31967	0

Views.py

```
importjson
fromdjango.shortcutsimportrender,HttpResponse
from django.http import JsonResponse
fromdjango.viewsimportView
from.modelsimportBlackOffer,tweets from
datetime import datetime
importnltk
fromdjango.dbimportmodels
fromdjango.db.modelsimportCount
#Addthisimport

fromnltk.sentimentimportSentimentIntensityAnalyzer
# Create your views here.

defindex(request):
    positive = tweets.objects.filter(sentiment='positive').count()
    negative=tweets.objects.filter(sentiment='negative').count()
    neutral   =   tweets.objects.filter(sentiment='neutral').count()
    all = tweets.objects.all().count()
    positive = round((positive/all)*10000)/100
    negative=round((negative/all)*10000)/100 neutral
    = round((neutral/all)*10000)/100 top_hashtags=
tweets.objects.values('hastag').annotate(tweet_count=Count('hastag')).order_by('-
tweet_count')[:10]
    top_username =
tweets.objects.values('username').annotate(tweet_count=Count('username')).order_
by('-tweet_count')[:10]
    print(top_hashtags)
    data={'positive':positive,'negative':negative,'neutral':neutral,
'top_hashtags':top_hashtags, 'top_username':top_username}
    returnrender(request,'index.html',data)

classUploadJsonView(View):
    defpost(self,request,*args,**kwargs):
        try:

            uploaded_file=request.FILES['json_file']
            data =json.load(uploaded_file)
```

17197rows×2columns

Views.py

```
importjson
fromdjango.shortcutsimportrender,HttpResponse
from django.http import JsonResponse
fromdjango.viewsimportView
from.modelsimportBlackOffer,tweets from
datetime import datetime
importnltk
fromdjango.dbimportmodels
fromdjango.db.modelsimportCount        #Addthisimport

fromnltk.sentimentimportSentimentIntensityAnalyzer #
Create your views here.

defindex(request):
    positive = tweets.objects.filter(sentiment='positive').count()
    negative=tweets.objects.filter(sentiment='negative').count()
    neutral   =   tweets.objects.filter(sentiment='neutral').count()
    all = tweets.objects.all().count()
    positive = round((positive/all)*10000)/100
    negative=round((negative/all)*10000)/100 neutral
    = round((neutral/all)*10000)/100 top_hashtags=
tweets.objects.values('hastag').annotate(tweet_count=Count('hastag')).order_by('-
tweet_count')[:10]
    top_username =
tweets.objects.values('username').annotate(tweet_count=Count('username')).order_
by('-tweet_count')[:10]
    print(top_hashtags)
    data={'positive':positive,'negative':negative,'neutral':neutral,
'top_hashtags':top_hashtags, 'top_username':top_username}
    returnrender(request,'index.html',data)

classUploadJsonView(View):
    defpost(self,request,*args,**kwargs):
        try:

            uploaded_file=request.FILES['json_file']
            data =json.load(uploaded_file)
```

```
foritemindata:

        end_year= item['end_year']
        ifend_year =="":
            end_year=None
        intensity=item['intensity']
        ifintensity == "":
            intensity =None
        sector=item['sector']
        topic=item['topic']
        insight=item['insight']
        url=item['url']
        region=item['region']
        start_year=item['start_year']
        ifstart_year=="":
            start_year =None
        impact=item['impact']
        ifimpact == "":
            impact = None
        added=item['adde
            parsed_date=datetime.strptime(added,"%B,%d%Y


            added=parsed_date.strftime("%Y-%m-%d%H:%M:%S") except:
            added = None
        published=item['published']
        try:
            parsed_date=datetime.strptime(published,"%B,%d
            published=parsed_date.strftime("%Y-%m-%d
        except:
        published=None %H:%M:%S")
            country=item['country'
        ] relevance=item['relevance']
        ifrelevance =="":
            relevance=None
```

```
p                    em['pestle']
e                    source=item['source']
s                    title=item['title']
t                    likelihood=item['likelihood']
l                    iflikelihood == "":
e
=
i
t

    try:if'json_file'inrequest.FILES:uploade
        d_file=request.FILES['json_file'] data
        =json.load(uploaded_file)

                for item in data.get('tweets', []):
                    date_str = item.get('date', None)
                    #date_str =item.get('date', None)
                    try:
                        parsed_date=datetime.fromisoformat(date_str) date
                        = parsed_date
                    except(ValueError,TypeError):
                        date = None

                    hashtag =item.get('hashtag', '')
                    username=item.get('username', '')
                    tweet_text =item.get('tweet', '')
                    sentiment = item.get('sentiment', '')
else:
    return JsonResponse({'error':'Nofileuploaded'},



tweets.objects.create( date=date, hastag=hashtag,
```

```
        exceptExceptionase:
            returnJsonResponse({'error':str(e)},status=500)


defsentiment(request):
    ifrequest.method=="POST":
        data   =   request.POST.get('text')
        sid= SentimentIntensityAnalyzer()
        sentiment_scores=sid.polarity_scores(data)
        ifsentiment_scores['compound'] >=0.05:
            sentiment_label   ='Positive'
        elifsentiment_scores['compound'] <= -0.05:
            sentiment_label ='Negative'
        else:
            sentiment_label='Neutral'

        returnrender(request,'sentiment.html',
{'sentiment_label':sentiment_label})
    returnrender(request,'sentiment.html')
```

REFERENCES

[1] David Osimo and Francesco Mureddu, "Research challenge on Opinion Mining and Sentiment Analysis"

[2] Maura Conway, Lisa McInerney, Neil O'Hare, Alan F. Smeaton, Adam Berminghan,"Combining Social Network Analysis and Sentiment to Explorethe Potential for Online Radicalisation," Centre for Sensor Web Technologies and School of Law and Government.

[3] Adobe® Social Analytics, powered by Omniture®.

[4] FabrizioS.AndreaE.,"Determining the Semantic Orientation of Terms through," October 31– November 5 2005.

[5] LucasC.,"Sentiment Analysis a Multimodal Approach,"_Department of Computing, Imperial College London_, September 2011.

[6] CarmineC., DiegoRFarahB.,"Sentiment Analysis : Adjectives and Adverbs are better," _ICWSM Boulder, CO USA_, 2006.

[7] Brandwatch.[Online].http://www.brandwatch.com/

[8] Sentiment140.[Online].http://www.sentiment140.com

[9] Learning to Classify Test [Online].
http://nltk.googlecode.com/svn/trunk/doc/book/ch06.html#document-classify-