Twitter Sentiment Analysis

LOVELY
PROFESSIONAL
UNIVERSITY

Project Report

Version 1.0

Developed By

Manish Das 12100139

Shahid Sadiq 12100459

Submitted To

Assistant Professor Mr Girish Kumar

Acknowledgment

The project we had under the STAR COURSE – CAP776 (PROGRAMMING IN PYTHON) was a great chance for learning and professional development. Therefore, we consider ourselves as a very lucky individual as we were provided with an opportunity to be a part of it.

We express our deepest thanks to our course instructor Mr. Girish Kumar (Assistant Professor) SCA, LPU for allowing us to grab this opportunity. We choose this moment to acknowledge his contribution gratefully by giving necessary advice and guidance to make our internship a good learning experience.

Manish Das - 12100139
Shahid Sadiq - 12100459

Table of Contents

1. Executive Summary

1.1 Project Overview

The goal of the project (Twitter Sentiment Analysis) is Identifying the general sentiment of a given document, which is also known as opinion mining, which is a subtask of NLP. We can extract the subjective information from a text and attempt to categories it according to its polarity, such as positive, neutral, or negative, using natural language processing. It is a very helpful analysis since we might maybe ascertain the general perception of a selling item or forecast stock prices for a specific company, for example, if most people have a favorable opinion of it, perhaps its stock prices would rise, and so on. Given the complexity of the language (objectivity/subjectivity, negation, lexicon, grammar), sentiment analysis is still far from being solved. In this project, we have opted to use a probabilistic model to try and categorize tweets from Twitter as having "positive" or "negative" sentiment. Twitter is a microblogging website where users can post 140-character tweets to express their feelings rapidly and spontaneously. By including the @ target symbol and the hashtag, you can contact someone directly in a tweet or join a discussion. Twitter is a wonderful source of information to ascertain the current general opinion about anything due to its popularity.

1.2 Purpose and Scope of this Specification

This effort will benefit businesses, political parties, and regular citizens alike. It will be beneficial to the political party to assess the programs that they have implemented or that they intend to implement. Like consumers, businesses can also receive feedback on brand-new hardware or software. The film's creator can also solicit feedback on a film that is already in theatres. By analyzing the tweets, the tweet analyzer can determine whether individuals are more favorable or negative about something.

1.3 Additional functionality

- Can Fetch up to 10000 Tweets.
- Gives a detailed Analysis of the tweets.
- Gives the most trending #tag, links and @mentions of the fetched tweets.
- Gives the Overall Sentiment Analysis of the Tweets in form of a graph.
- You can download/export the tweets in form of CSV file.

2. Product/Service Description

2.1 User Characteristics

- User Type - The intended user will be a member of the general public who is interested in the sentiment of the Twitter population with respect to various topics.
- experience – Basic computer knowledge
- technical expertise - Users are not expected to have a very high level of technical expertise.

2.2 Assumptions

- Sentiment Analysis An assumption is that it is possible to accurately determine the sentiment for a 140 character string of English text.

- Internet access is required for each analysis session to function properly. An assumption Users should have an operating system with modern browsers which support the latest version of JS (ES6+).
- If any developer wants to modify the source code he/she must have twitter developer account otherwise he/she cannot get API key from twitter.

2.3 Constraints

Individual Data Twitter does not provide any user-provided data that has not already been made public. Any personal data gathered via Twitter won't be saved or utilized in any other way.

The terms of service for Twitter Developer must be followed by the application. The following is included in this:

● Defining an application privacy policy (what we do with tweets, user data, etc.)

● Not redistributing Tweets

● Providing a link to Twitter sign-up if user does not have a registered Twitter account

2.4 Dependencies

- Internet connection must be there in order to fetch the real-time data from twitter.

3. Requirements

3.1 Functional Requirements

Retrieving Input

The software will receive three inputs: keywords, analysis session duration, and Tweets.

● Keywords will be entered by the user for each topic.

● The analysis session duration will be set by the user before each session.

● Tweets will be retrieved with the Twitter Streaming API.

Real-Time Processing

Real-time input, processing, and output display are all features of the software. This will guarantee that the simple gauge's picture of the present state of the Twitter community about the selected subject is accurate.

Sentiment Analysis

The user-specified keywords in the Tweet will be subject to sentiment analysis to ascertain the overall mood of the Tweet in relation to the subject. A negative, neutral, or positive numerical sentiment value will be presented as a result of the sentiment analysis.

Output

Real-time data must be output by the software in the form of a straightforward gauge. The software can also generate extra information about a subject and a graph showing mood trends over time (average sentiment over all analysis sessions and total number of tweets processed). This output ought to be understandable and clear.

3.2 User Interface Requirements

3.2.1 Software Interfaces

Inputs

There will be two sources of input for the software. The user interface comes first, followed by the Twitter API. While the Twitter API provides the Tweet text, the user interface provides the keywords and the number of tweets the user wants to conduct an action on.

Outputs

In the form of a straightforward gauge, the output will show the current attitude of the Twitter community toward a particular subject. If accessible, a graph will show historical data.

Operating System

The software will run on any operating system as long it has an updated browser with proper JavaScript support, specifically ES6+ and above.

3.2.2 User Interfaces

To satisfy user needs, the interface must adhere to the following standards. It will be clear and straightforward to comprehend. Clear controls that imply their functioning within the application will be provided for the user to engage with. The user inputs and two visuals that make up the interface are shown below. The user will get a visual depiction of the output created in the graphics displayed to them.

User Inputs (Mandatory)

The user can modify the functionality by using the drop-down menu and then searching by the username. The user can also change the functionality by inputting hashtags for the topic and then defining the term (if he wants to search about any specific user).

Graphic 1: Number of tweets the user wants to fetch (Mandatory)

This gauge(slider) can be used by the user to set the number of tweets he/she want to fetch at a time this slider is very easy to use.

Error Notifications (Mandatory)

It will be necessary for the programme to provide error notifications that inform the user of the error that has occurred and display the appropriate messages. Error messages should, if applicable, offer potential fixes for the issue.

Choose the searching mode(Optional)

We have used a dropdown list by which users can either search by # or username according to his/her mood notification.

Submit button (Mandatory)

After entering the #hashtag, keywords or “username” user must click on submit button in order to proceed further.

3.3 Usability

Learnability

- The user documentation and installation help guide

https://github.com/ManishDass/Online-Student-Admission

3.4 Performance

- 1 Tweets takes approx. 0.05 sec in normal working load so you may have to wait 0.08 minute for 100 Tweets.

3.4.1 Capacity

There is no as such user limitation, as long as server supports it will work. It mainly depends upon server performance.

3.4.2 Availability

- As long as the user's device is in good functioning condition, it is always accessible. Any required external services, such as internet access, will affect the software's operation. The user must be informed if those services are not available.

- Unless there are restrictions imposed by the government, accessible everywhere.

- During downtime, users are unable to complete any operations.

- Reliability: The website is operational 24 hours a day, seven days a week. Assuming that there is only one hour of maintenance time per month, the MTBF for that piece of equipment is 672 hours.

3.4.3 Latency

- 1 Tweets takes approx. 0.05 sec in normal working load so you may have to wait 0.08 minute for 100 Tweets.

3.5 Manageability/Maintainability

Maintainability: Software should be written in an understandable and straightforward manner. The code will include thorough documentation. To make maintenance simple, special care will be made in the software's modular design.

3.5.1 Monitoring

Include any requirements for product or service health monitoring, failure conditions, error detection, logging, and correction.

3.5.2 Maintenance

The software should be written clearly and concisely. The code will be well documented. Particular care will be taken to design the software modularly to ensure that maintenance is easy

3.6 System Interface/Integration

Any operating system which supports modern browser will work.

3.7 Security

3.7.1 Protection

The software should never disclose any personal information of Twitter users, and should collect no personal information from its own users.

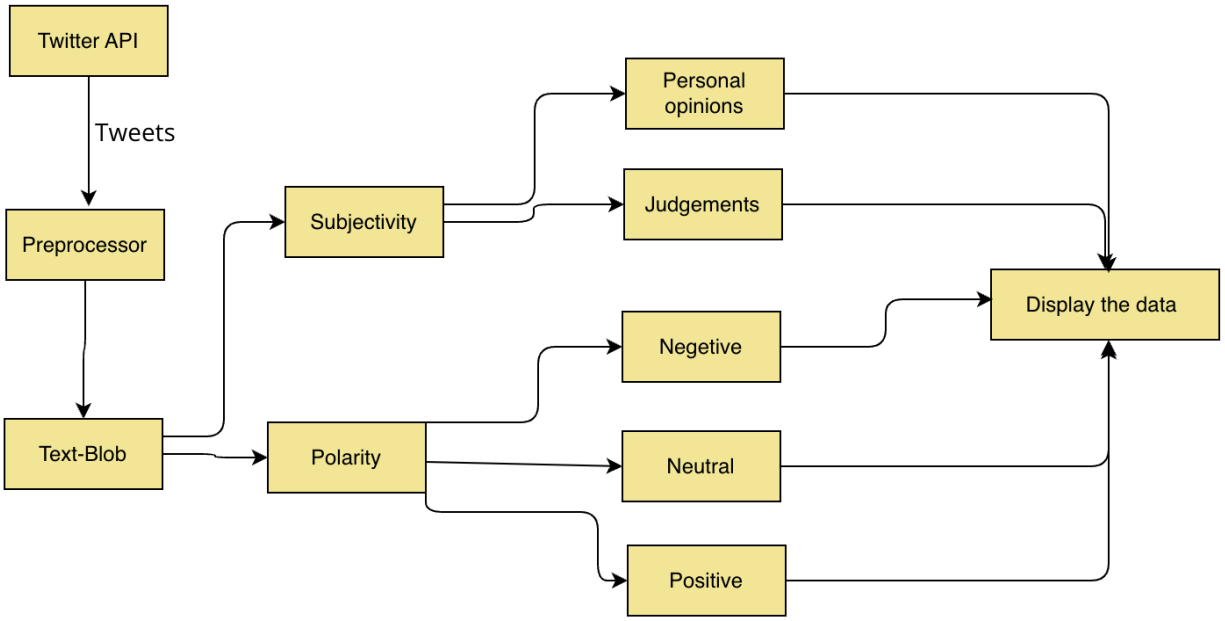
3.8 Data Management

The software will receive three inputs: keywords, number of Tweets.

- Keywords will be entered by the user for each topic.
- Tweets will be retrieved with the Twitter Streaming API.

So there is no database is being used everything is done by fetching Realtime data from twitter using the API.

3.9 DFD



4. User Scenarios/Use Cases

This software will serve as a tool of interest, providing users with the current mood of the Twitter community on any specified topic.

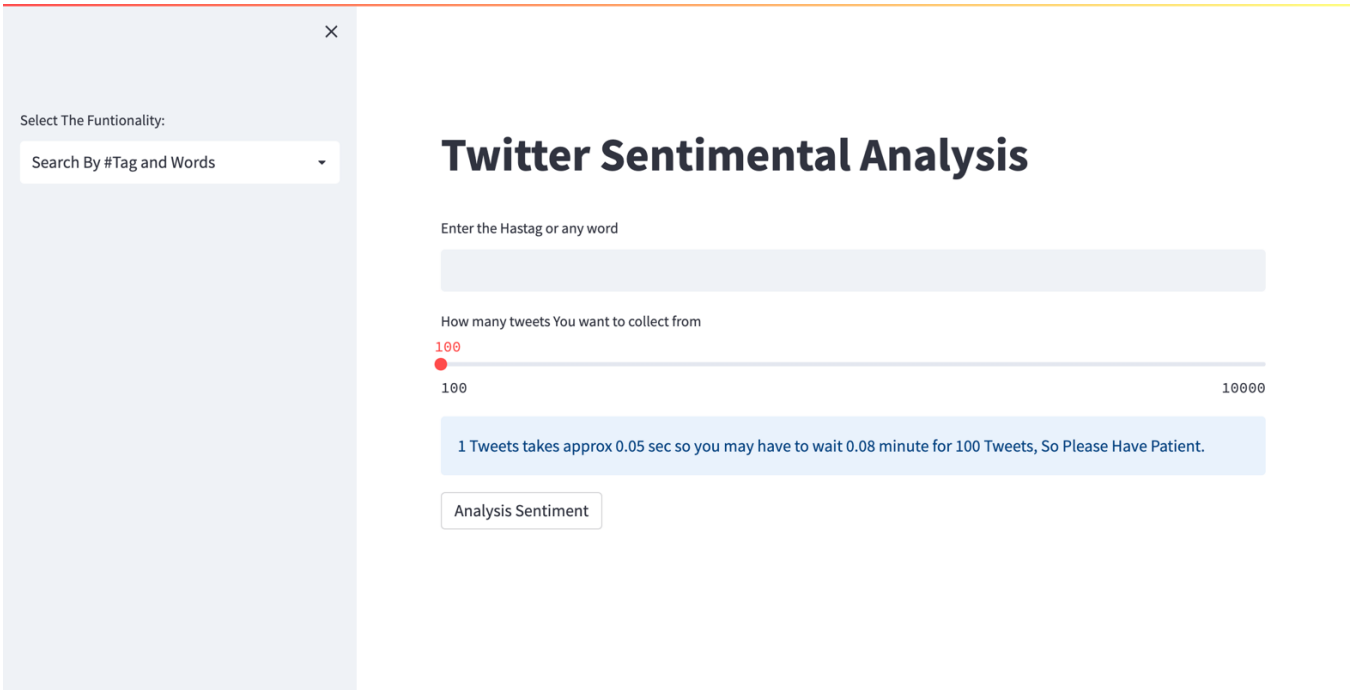
5. Deleted or Deferred Requirements

Identify any requirements that have been deleted after approval or that may be delayed until future versions of the system. For example:

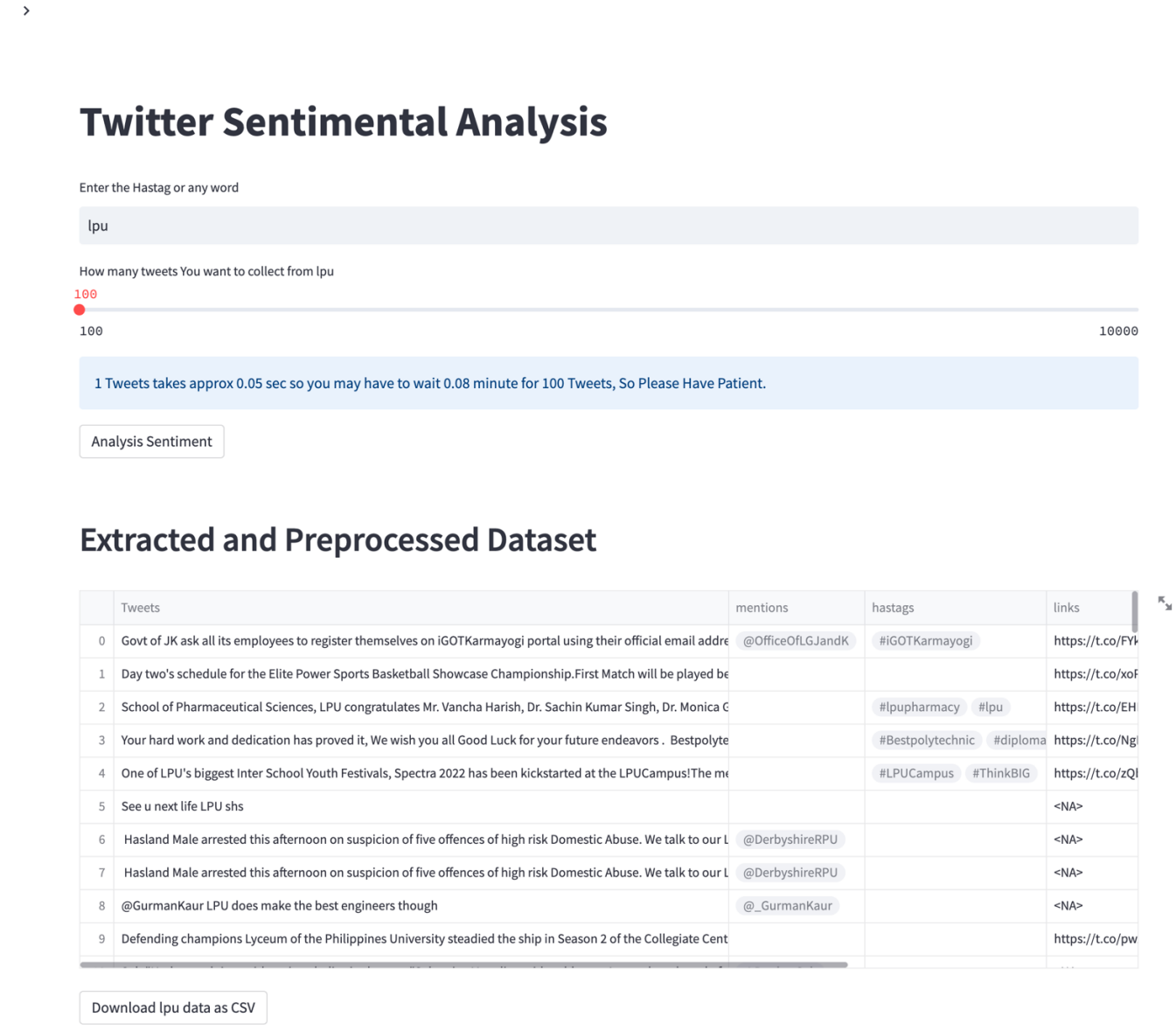
Req#	Business Requirement	Status	Comments	Pri	SME Reviewed /Approved
#storing_data/downloading data	For future reference	Completed	Maybe it will be implemented in the future	NA	Approved

6. Output

Index Page

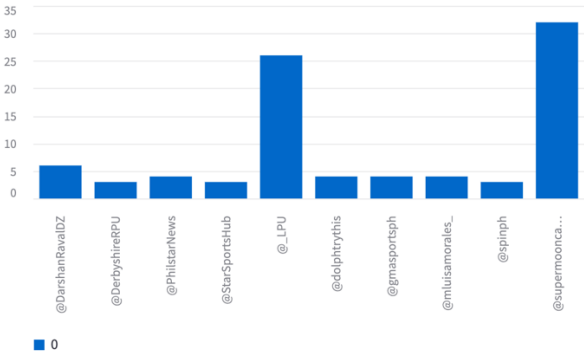


Analysis Page

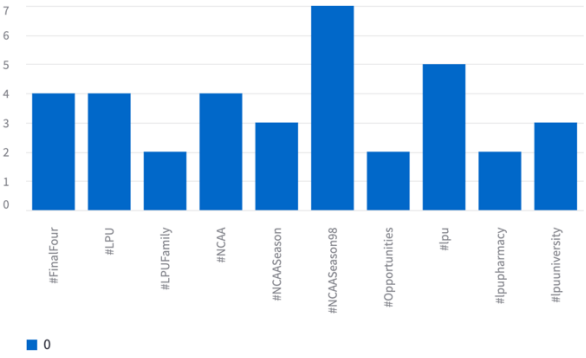


EDA On the Data

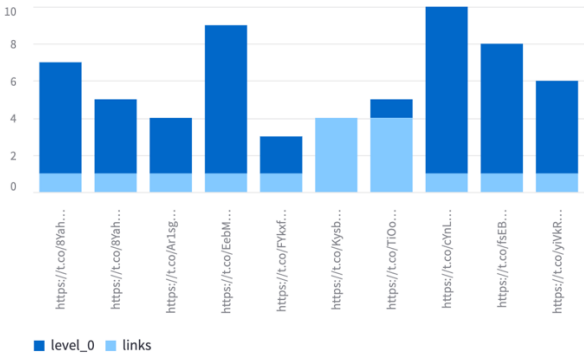
Top 10 @Mentions in 100 tweets



Top 10 Hastags used in 100 tweets



Top 10 Used Links for 100 tweets



All the Tweets that containes top 10 links used

	Tweets
0	Govt of JK ask all its employees to register themselves on iGOTKarmayogi portal using t
34	In-form Pirates zero in on Top Two spot NCAA FinalFour By
36	School of Pharmaceutical Sciences, LPU congratulates Dr. Sachin Kumar Singh and tear
44	Congratulations to the students of BTech. ECE from School of Electronics and Electrical
46	Enoch Valdez picked up where he left off as he bucketed 21 points and made 89% of Lyc
48	LPU gains lead as Letran stops Arellano in CCE MLBB Season 2 |
50	In-form Pirates zero in on Top Two spot NCAA FinalFour By
51	In-form Pirates zero in on Top Two spot NCAA FinalFour By
52	LPU gains lead as Letran stops Arellano in CCE MLBB Season 2 |
53	LPU gains lead as Letran stops Arellano in CCE MLBB Season 2 |

Twitter Sentment Analysis Bar Chart

9. Source Code

app.py

```
#=========================================
# Title:  Twitter Sentiment Analysis
# Author: Manish Das & Shahid Sadiq
# Date:   12 Nov 2022
#Status: TextCloud Module is in progress
#=========================================

from attr import has
import streamlit as st
from helper import preprocessing_data, graph_sentiment, analyse_mention, analyse_hastag,
download_data


st.set_page_config(
     page_title="TSA By Manish Das",
     page_icon="📊",
     layout="wide",
     initial_sidebar_state="expanded",
)

title = "";

st.title("Twitter Sentimental Analysis ")

function_option = st.sidebar.selectbox("Select The Funtionality: ", ["Search By #Tag and
Words", "Search By Username"])

if function_option == "Search By #Tag and Words":
    word_query = st.text_input("Enter the Hastag or any word")
    title = word_query

if function_option == "Search By Username":
    word_query = st.text_input("Enter the Username ( Don't include @ )")
    title = word_query

number_of_tweets = st.slider("How many tweets You want to collect from {}".format(word_query),
min_value=100, max_value=10000)
st.info("1 Tweets takes approx 0.05 sec so you may have to wait {} minute for {} Tweets, So
Please Have Patient.".format(round((number_of_tweets*0.05/60),2), number_of_tweets))

if st.button("Analysis Sentiment"):
```

```
    data = preprocessing_data(word_query, number_of_tweets, function_option)
    analyse = graph_sentiment(data)
    mention = analyse_mention(data)
    hastag = analyse_hastag(data)

    st.write(" ")
    st.write(" ")
    st.header("Extracted and Preprocessed Dataset")
    st.write(data)
    download_data(data, label=title)
    st.write(" ")


    col1, col2, col3 = st.columns(3)
    with col2:
        st.markdown("### EDA On the Data")


    col1, col2 = st.columns(2)

    with col1:
        st.text("Top 10 @Mentions in {} tweets".format(number_of_tweets))
        st.bar_chart(mention)
    with col2:
        st.text("Top 10 Hastags used in {} tweets".format(number_of_tweets))
        st.bar_chart(hastag)

    col3, col4 = st.columns(2)
    with col3:
        st.text("Top 10 Used Links for {} tweets".format(number_of_tweets))
        st.bar_chart(data["links"].value_counts().head(10).reset_index())

    with col4:
        st.text("All the Tweets that containes top 10 links used")
        filtered_data =
data[data["links"].isin(data["links"].value_counts().head(10).reset_index()["index"].values)]
        st.write(filtered_data)


    #Printing Bar Chart
    st.subheader("Twitter Sentment Analysis Bar Chart")
    st.bar_chart(analyse)


    #Display Wordcloud //Work in Progress
    #st.text(hastag)
    # stopwords = set(STOPWORDS)
    # stopwords.update(["the", "a", "an", "in"])
    # wordcloud = WordCloud(width=1600,
stopwords=stopwords,height=800,max_font_size=200,max_words=50,collocations=False,
background_color='black').generate(hastag)
```

```
    # plt.figure(figsize=(40,30))
    # plt.imshow(wordcloud, interpolation="bilinear")
    # plt.axis("off")
    # plt.show()


#Just to hide the "Made with Streamlit"
hide_streamlit_style = """
            <style>
            #MainMenu {visibility: hidden;}
            footer {visibility: hidden;}
            </style>
            """
st.markdown(hide_streamlit_style, unsafe_allow_html=True)
```

helper.py

```
#==========================================
# Title:  Twitter Sentiment Analysis
# Author: Manish Das & Shahid Sadiq
# Date:   12 Nov 2022
#==========================================

import tweepy
import pandas as pd
import configparser
import re
from textblob import TextBlob
import streamlit as st
import datetime, pytz


emoji_pattern = re.compile("["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
        u"\U00002500-\U00002BEF"  # chinese char
        u"\U00002702-\U000027B0"
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        u"\U0001f926-\U0001f937"
        u"\U00010000-\U0010ffff"
        u"\u2640-\u2642"
        u"\u2600-\u2B55"
```

```
        u"\u200d"
        u"\u23cf"
        u"\u23e9"
        u"\u231a"
        u"\ufe0f"  # dingbats
        u"\u3030"  # flags (iOS)
        "]+", flags=re.UNICODE)



def twitter_connection():

    config = configparser.ConfigParser()
    config.read("config.ini")

    api_key = config["twitter"]["api_key"]
    api_key_secret = config["twitter"]["api_key_secret"]
    access_token = config["twitter"]["access_token"]

    auth = tweepy.OAuthHandler(api_key, api_key_secret)
    api = tweepy.API(auth)

    return api


api = twitter_connection()


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)
    text = re.sub("\n","",text) # Removing hyperlink
    text = re.sub(":","",text) # Removing hyperlink
    text = re.sub("_","",text) # Removing hyperlink
    text = emoji_pattern.sub(r'', text) #or re.sub(emoji_pattern, '', text) is same thing
    return text

def extract_mentions(text):
    text = re.findall("(@[A-Za-z0-9\d\w]+)", text)
    return text

def extract_hastag(text):
    text = re.findall("(#[A-Za-z0-9\d\w]+)", text)
    return text

def getSubjectivity(text):
   return TextBlob(text).sentiment.subjectivity
```

```
# Create a function to get the polarity
def getPolarity(text):
   return  TextBlob(text).sentiment.polarity

def getAnalysis(score):
  if score < 0:
    return 'Negative'
  elif score == 0:
    return 'Neutral'
  else:
    return 'Positive'



@st.cache(allow_output_mutation=True)
def preprocessing_data(word_query, number_of_tweets, function_option):

  if function_option == "Search By #Tag and Words":
    posts = tweepy.Cursor(api.search_tweets, q=word_query, count = 200, lang ="en",
tweet_mode="extended").items((number_of_tweets))

  if function_option == "Search By Username":
    posts = tweepy.Cursor(api.user_timeline, screen_name=word_query, count = 200,
tweet_mode="extended").items((number_of_tweets))

  data  = pd.DataFrame([tweet.full_text for tweet in posts], columns=['Tweets']) #Fetching Raw
Data from Twitter

  data["mentions"] = data["Tweets"].apply(extract_mentions)
  data["hastags"] = data["Tweets"].apply(extract_hastag)
  data['links'] = data['Tweets'].str.extract('(https?:\/\/\S+)', expand=False).str.strip()
  data['retweets'] = data['Tweets'].str.extract('(RT[\s@[A-Za-z0-9\d\w]+)',
expand=False).str.strip()

  data['Tweets'] = data['Tweets'].apply(cleanTxt)
  discard = ["CNFTGiveaway", "GIVEAWAYPrizes", "Giveaway", "Airdrop", "GIVEAWAY",
"makemoneyonline", "affiliatemarketing", "GiveawayWinner"]
  data = data[~data["Tweets"].str.contains('|'.join(discard))] #remove all the keywords which
having discard keywords

  data['Subjectivity'] = data['Tweets'].apply(getSubjectivity)
  data['Polarity'] = data['Tweets'].apply(getPolarity)

  data['Analysis'] = data['Polarity'].apply(getAnalysis)

  return data #Returing clean and formatted Data


def download_data(data, label):
    current_time = datetime.datetime.now(pytz.timezone('Asia/Kolkata'))
```

```
    current_time = "_{}_{}-{}-{}".format(current_time.date(), current_time.hour,
current_time.minute, current_time.second)
    export_data = st.download_button(
                        label="Download {} data as CSV".format(label),
                        data=data.to_csv(),
                        file_name='{}{}.csv'.format(label, current_time),
                        mime='text/csv',
                    )
    return export_data




def analyse_mention(data):
  mention = pd.DataFrame(data["mentions"].to_list()).add_prefix("mention_")

  try:
    mention = pd.concat([mention["mention_0"], mention["mention_1"], mention["mention_2"]],
ignore_index=True)
  except:
    mention = pd.concat([mention["mention_0"]], ignore_index=True)

  mention = mention.value_counts().head(10)
  return mention



def analyse_hastag(data):

  hastag = pd.DataFrame(data["hastags"].to_list()).add_prefix("hastag_")

  try:
    hastag = pd.concat([hastag["hastag_0"], hastag["hastag_1"], hastag["hastag_2"]],
ignore_index=True)
  except:
    hastag = pd.concat([hastag["hastag_0"]], ignore_index=True)

  hastag = hastag.value_counts().head(10)
  return hastag




def graph_sentiment(data):

  analys = data["Analysis"].value_counts().reset_index().sort_values(by="index",
ascending=False)
  return analys
```

Further all the necessary code is uploaded in this GitHub repo

https://github.com/ManishDass/Twitter-Sentiment-Analysis

10. Role of team members

1. Manish Das

    - Worked on the module (Data Pre-processing)

2. Shahid Sadiq

    - Worked on the module (user interface)

11. References

Streamlit.io : https://streamlit.io/

Pandas : https://pandas.pydata.org/

Tweepy : https://www.tweepy.org/

TextBlob : https://textblob.readthedocs.io/en/dev/

Twitter Developer : https://developer.twitter.com/en