SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Department of ECE

18ECE307J Applied Machine Learning

Project Title: Tweet Visualization and Sentiment Analysis

Team members:

V Andal Priyadharshini (RA1811004010087)

Vemu Lakshmi Ananya (RA1811004010089)

Lab Supervisor : Dr Malarvizhi G

TWEET VISUALIZATION AND SENTIMENT ANALYSIS

OBJECTIVE

The main objective of this task is to visualize tweets and perform sentiment analysis. We have to perform the following:

- To collect real-time data from Twitter using tweepy module for Tweet visualization.
- To clean the data to make processing easier.
- To perform sentiment Analysis based on the polarity of the tweets using TextBlob.
- To plot the positive, negative and neutral tweets on a Word cloud.
- To plot the count the number of tweets based on trending hashtags using NLTK and seaborn.

ABSTRACT

Social media have received more attention nowadays. Public and private opinion about a wide variety of subjects is expressed and spread continually via numerous social media. Twitter is one of the social media that is gaining popularity. Twitter offers organizations a fast and effective way to analyze customers' perspectives toward the critical to success in the market place. Developing a program for sentiment analysis is an approach used to extract sentiments out of the tweet classifying it into positive, negative or neutral. Here, we are using real-time data to create a dataset. Generally, the tweets are unstructured in format, so data cleaning has to be done to process the data. In this project, tweets are resolved using pre-processing phase and access of tweets has been accomplished via libraries using Twitter API developer account. The datasets are trained using algorithms in a way, such that, it becomes capable of classifying the tweets and it releases the required sentiments out of customers' perceptions based on the trending hashtags too.

ABOUT THE DATAFRAME CREATED

COLUMN NAME	DESCRIPTION
ID	Serial number of tweets
Tweet	Realtime data extracted from twitter
Clean_tweet	Further hashtags are removed to find the polarity.
Polarity	Getting the polarity score from TextBlob.
Analysis	Represent a positive, negative or neutral emotion based on polarity

Table 1: Dataframe Description

We are also finding out the top 10 hashtags for each type and plotting it based on the number of positive, negative, neutral tweets based on each hash-tags. A final data-frame is created for the graph plotted.

BLOCK DIAGRAM

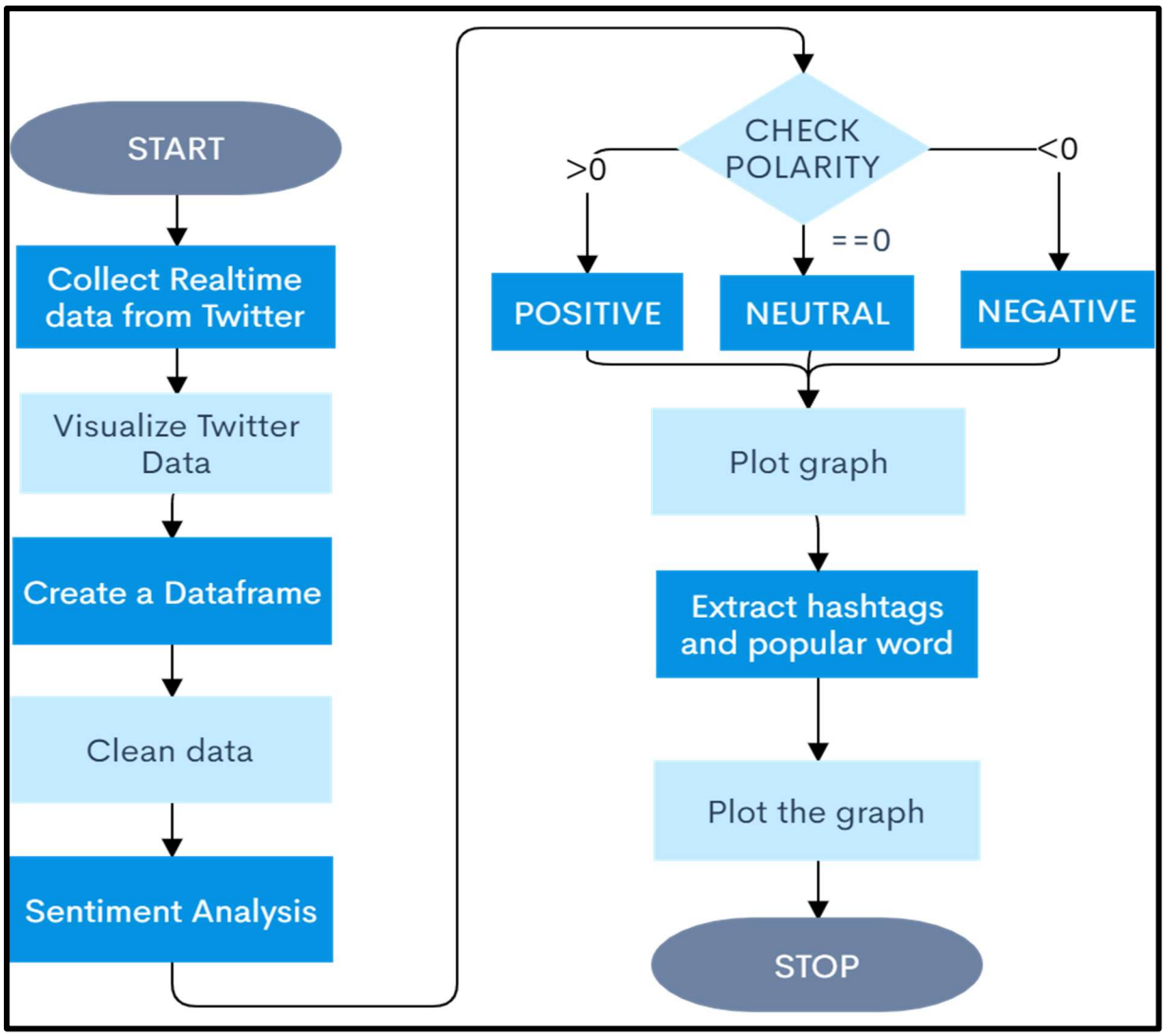

Figure 1: Flowchart for the program

PROGRAM

```
#import the libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import re
import warnings
warnings.filterwarnings('ignore')
```

```
import nltk
from tweepy import *
from textblob import *
from wordcloud import *
import credentials
accessToken = credentials.accessToken
accessTokenSecret = credentials.accessTokenSecret
consumerKey = credentials.consumerKey
consumerSecret = credentials.consumerSecret
authenticate = OAuthHandler(consumerKey, consumerSecret)
authenticate.set_access_token(accessToken, accessTokenSecret)
api=API(authenticate, wait_on_rate_limit=True)
#tweet Visualization
posts = api.user_timeline(screen_name="timesofindia", count=100, lang="en",
tweet_mode="extended")
print("Show the 5 recent tweets: \n")
i=1
for tweet in posts[0:5]:
    print(str(i)+')'+tweet.full_text+'\n')
    i=i+1
#creating a dataframe
df=pd.DataFrame([i+1 for i in range(100)],columns=['id'])
df['tweet']=[tweet.full_text.lower() for tweet in posts]
#show the first 5 rows of data
df.head()
# removes pattern in the input text
def remove_pattern(input_txt, pattern):
    r = re.findall(pattern, input_txt)
    for word in r:
        input_txt = re.sub(word, "", input_txt)
    return input_txt
# remove twitter handles (@user)
df['hash_tweet'] = np.vectorize(remove_pattern)(df['tweet'], "@[\w]*")
# remove links (@user)
df['hash_tweet'] = np.vectorize(remove_pattern)(df['tweet'], "https[\w]*")
# remove all characters except # and alphabets
df['hash_tweet'] = df['hash_tweet'].str.replace("[^a-zA-Z#]", " ")
# remove short words that do not affect the polarity
df['hash_tweet'] = df['hash_tweet'].apply(lambda x: " ".join([w for w in x.split() if len(w)>3]))
# individual words considered as tokens
tokenized_tweet = df['clean_tweet'].apply(nltk.word_tokenize)
stop = stopwords.words('english')
tokenized_tweet=tokenized_tweet.apply(lambda x: [item for item in x if item not in stop])
for i in range(len(tokenized_tweet)):
    tokenized_tweet[i] = " ".join(tokenized_tweet[i])
df['clean_tweet'] = tokenized_tweet
df.head()
df.head()
# remove hashtags
```

```
df['clean_tweet'] = np.vectorize(remove_pattern)(df['hash_tweet'], "#[\w]*")
df.head()
# plot the graph
def graph(word):
    wordcloud = WordCloud(width=800, height=500, random_state=42,
max_font_size=100).generate(word)
    plt.figure(figsize=(15,8))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.show()
# visualize the frequent words
all_words = " ".join([sentence for sentence in df['clean_tweet']])
graph(all_words)
#function to get polarity
def getPolarity(text):
    return TextBlob(text).sentiment.polarity
#Create two columns subjectivity and polarity
df['Polarity']=df['clean_tweet'].apply(getPolarity)
#show the new dataframe
df.head()
#create a function to compute negative, neutral and positve analysis
def getAnalysis(score):
    if score<0:
        return 'Negative'
    elif score==0:
        return 'Neutral'
    else:
        return 'Positive'
df['Analysis']=df['Polarity'].apply(getAnalysis)
#show the dataframe
df.head()
pos_words = " ".join([sentence for sentence in df['clean_tweet'][df['Analysis']=='Positive']])
graph(pos_words)
# frequent words visualization for -ve
neg_words = " ".join([sentence for sentence in df['clean_tweet'][df['Analysis']=='Negative']])
graph(neg_words)
# frequent words visualization for neutral
neu_words = " ".join([sentence for sentence in df['clean_tweet'][df['Analysis']=='Neutral']])
graph(neu_words)
def percentage(vibe):
    pertweets= df[df.Analysis ==vibe]
    pertweets = pertweets['tweet']
    vibeval=round(pertweets.shape[0]/df.shape[0]*100,1)
    print(vibe,"Tweets:",end=" ")
    return vibeval
positive=percentage('Positive')
print(positive)
negative=percentage('Negative')
print(negative)
```

```
neutral=percentage('Neutral')
print(neutral)
#show the value counts
df['Analysis'].value_counts()
plt.title('Sentiment Analysis')
plt.xlabel('Sentiment')
plt.ylabel('Counts')
df['Analysis'].value_counts().plot(kind='bar',color='red')
plt.show()
# extract the hashtag
def hashtag_extract(tweets):
    hashtags = []
    for tweet in tweets:
        ht = re.findall(r"#(\w+)", tweet)
        hashtags.append(ht)
    return hashtags
# extract hashtags from positive tweets
ht_positive = hashtag_extract(df['hash_tweet'][df['Analysis']=='Positive'])
# extract hashtags from negative tweets
ht_negative = hashtag_extract(df['hash_tweet'][df['Analysis']=='Negative'])
# extract hashtags from negative tweets
ht_neutral = hashtag_extract(df['hash_tweet'][df['Analysis']=='Neutral'])
#Remove the empty list items within the list
ht_positive = sum(ht_positive, [])
ht_negative = sum(ht_negative, [])
ht_neutral = sum(ht_neutral, [])
ht_positive[:5]
def pdframe(vibe):
    freq = nltk.FreqDist(vibe)
    d = pd.DataFrame({'Hashtag': list(freq.keys()),'Count': list(freq.values())})
    # select top 10 hashtags
    d = d.nlargest(columns='Count', n=10)
    plt.figure(figsize=(20,9))
    sns.barplot(data=d, x='Hashtag', y='Count')
    plt.show()
    return d
print('Positive tweets based on hashtags')
freq_pos = pdframe(ht_positive)
freq_pos
print('Negative tweets based on hashtags')
freq_neg = pdframe(ht_negative)
freq_neg
print('Neutral tweets based on hashtags')
freq_neu = pdframe(ht_neutral)
freq_neu
```

RESULT AND DISCUSSION

```
In [5]: posts = api.user_timeline(screen_name="timesofindia", count=100, lang="en", tweet_mode="extended")
        print("Show the 5 recent tweets: \n")
        i=1
        for tweet in posts[0:5]:
            print(str(i)+')'+tweet.full_text+'\n')
            i=i+1
```

```
Show the 5 recent tweets:

1)Over 2.15 lakh beneficiaries of age group 18-44 get vaccine

#COVID19 Live updates: https://t.co/HA3gZ90Rwe https://t.co/TydNK2ch7I

2)Chhattisgarh reports 15,274 new #COVD19 cases, 1,088 recoveries and 266 deaths in the last 24 hours Active cases: 1,20,977 De
ath toll: 9,275 https://t.co/755a47HJoq

3)#Update | Election Commission of India lifts model code of conduct from states where elections were conducted

4)RT @toisports: #IPL2021

It is in players' hands to make a choice: @GraemeSmith49 on South Africans willing to exit #IPL 🏏

As many as 11…

5)You will have to wait till 2022 for Facebook Messenger, Instagram chats to get E2E encryption

https://t.co/e8QdOJzDQa via @gadgetsnow https://t.co/fRGh9WmefY
```

Figure 2: Visualization of the top 5 tweets from the real-time data collected from Twitter

```
In [9]: # remove hashtags
        df['clean_tweet'] = np.vectorize(remove_pattern)(df['hash_tweet'], "#[\w]*")
        df.head()
```

Out[9]:

	id	tweet	hash_tweet	clean_tweet
0	1	over 2.15 lakh beneficiaries of age group 18-4...	over lakh beneficiaries group vaccine #covid l...	over lakh beneficiaries group vaccine live up...
1	2	chhattisgarh reports 15,274 new #covd19 cases,...	chhattisgarh reports #covd cases recoveries de...	chhattisgarh reports cases recoveries deaths ...
2	3	#update | election commission of india lifts m...	#update election commission india lifts model ...	election commission india lifts model code co...
3	4	rt @toisports: #ipl2021 \n\nit is in players' ...	toisports #ipl players hands make choice graem...	toisports players hands make choice graemesmi...
4	5	you will have to wait till 2022 for facebook m...	will have wait till facebook messenger instagr...	will have wait till facebook messenger instagr...

Dataframe 1: Dataframe after cleaning

```
In [11]: # visualize the frequent words
         all_words = " ".join([sentence for sentence in df['clean_tweet']])
         graph(all_words)
```

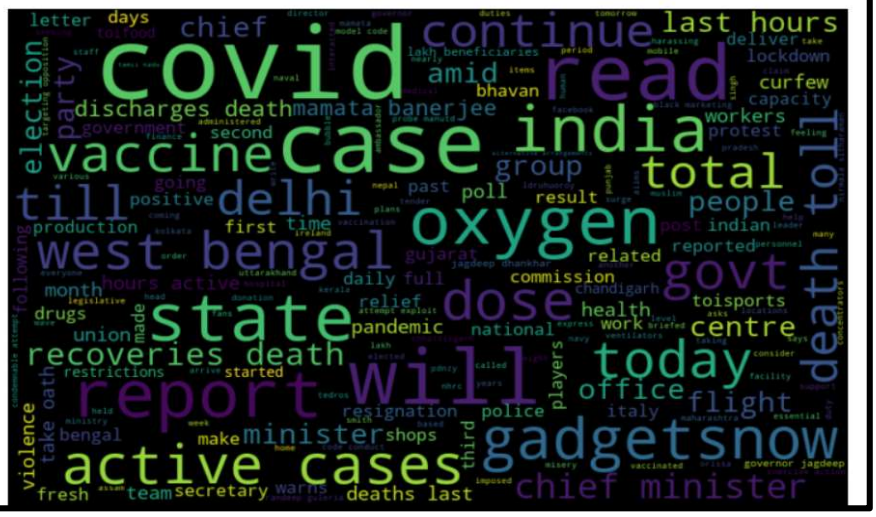

Figure 3: WordCloud to picturize all words in tweets

The data from clean_tweet column is used to obtain this word cloud. The words appeared more frequent are bigger in size than less frequent words.

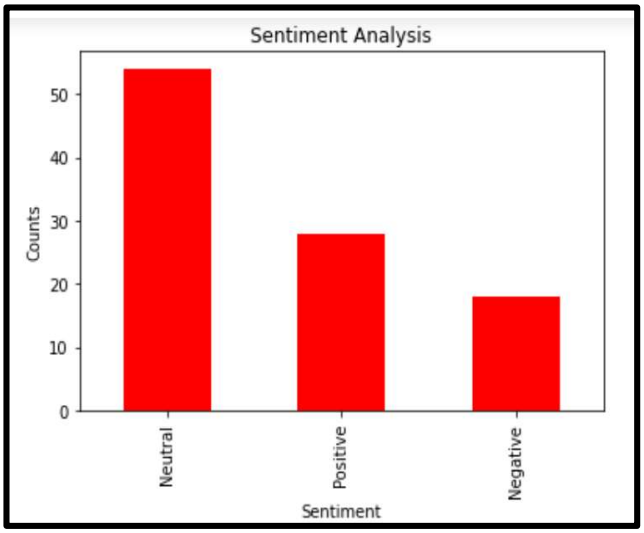

Figure 4: Count of tweets based on sentiment

clean_tweet	Polarity	Analysis
over lakh beneficiaries group vaccine live up...	0.136364	Positive
chhattisgarh reports cases recoveries deaths ...	-0.066667	Negative
election commission india lifts model code co...	0.000000	Neutral
toisports players hands make choice graemesmi...	0.375000	Positive
will have wait till facebook messenger instagr...	0.000000	Neutral

Dataframe 2: Polarity Analysis of Tweets

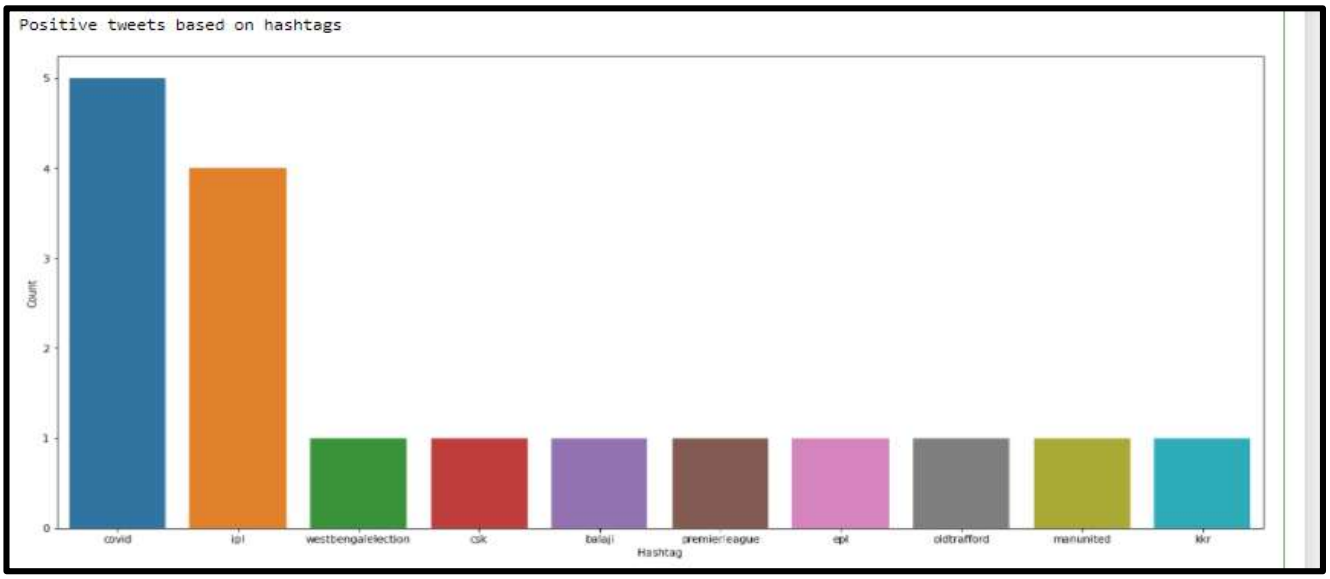

Figure 5: Number of positive tweets based on trending hashtags

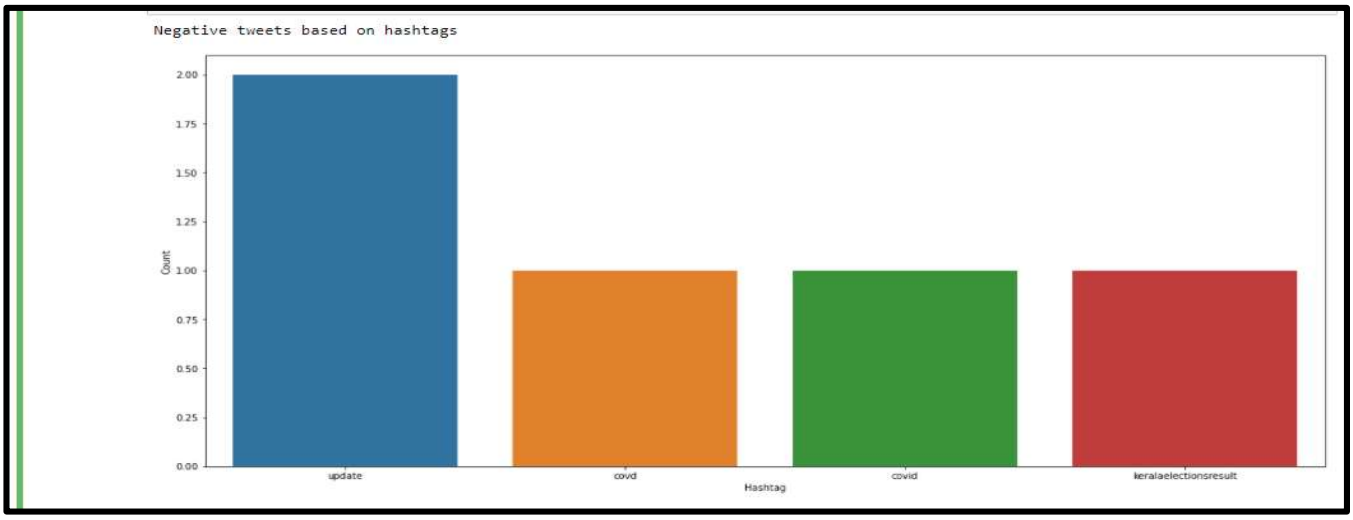

Figure 6: Number of negative tweets based on trending hashtags

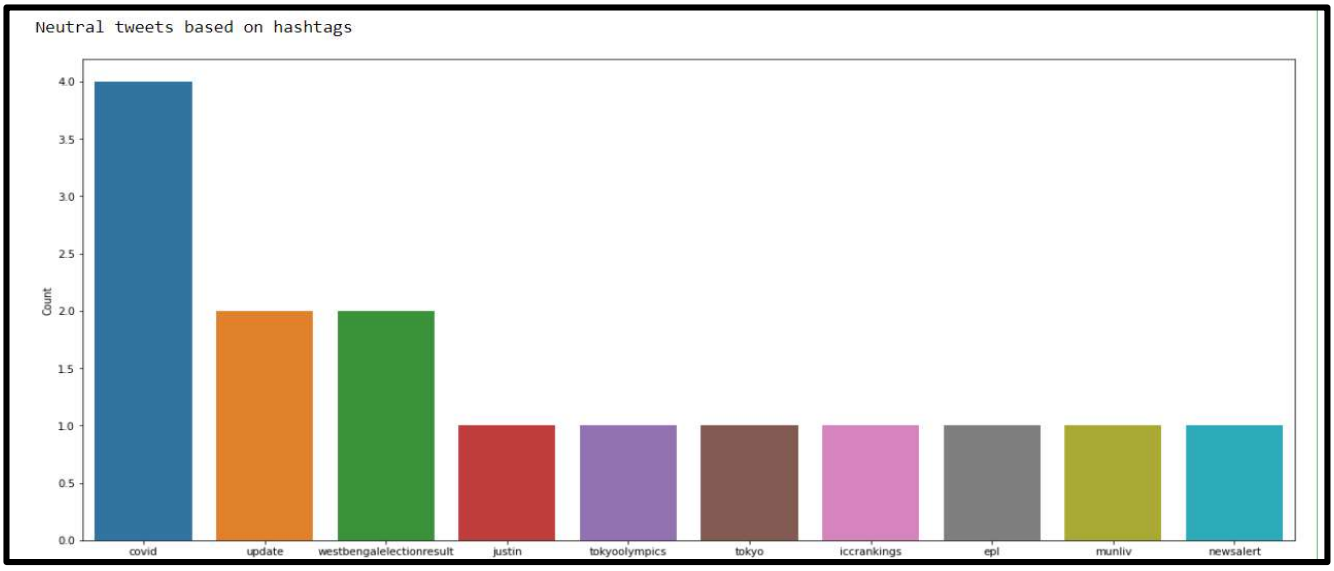

Figure 7: Number of neutral tweets based on trending hashtags

POSITIVE

	Hashtag	Count
0	covid	5
1	ipl	4
2	westbengalelection	1
3	csk	1
4	balaji	1
5	premierleague	1
6	epl	1
7	oldtrafford	1
8	manunited	1
9	kkr	1

NEGATIVE

	Hashtag	Count
4	covid	4
0	update	2
5	westbengalelectionresult	2
1	justin	1
2	tokyoolympics	1
3	tokyo	1
6	iccrankings	1
7	epl	1
8	munliv	1
9	newsalert	1

NEUTRAL

	Hashtag	Count
1	update	2
0	covd	1
2	covid	1
3	keralaelectionsresult	1

Dataframe 3: Top 10 Positive, Negative and Neutral Hashtag Counts

DISCUSSIONS:

We have analyzed the tweets taken from the timeline of timesofindia twitter page. However, by changing the screen_name attribute in api_usertimeline, we can obtrain the tweets of other pages in twitter as well. Tweets can be obtained only if we have a twitter developer account. Getting a developer account was very hard as our application had to undergo seven reviews before we got it approved. The twitter developer policies are very strict, so we are not allowed to share the credentials in public. We have used credentials module that was created by us to access the secret consumer and access keys.

CONCLUSION

Twitter is a huge platform and source of improperly structured and sentiment datasets that can be analyzed to produce trending emotions and many more. In Twitter sentiment analysis we inspect or mine each and every element of the real-time tweets extracted from twitter. This project explains various steps involved in visualization and analysis of twitter sentiments using TextBlob, WordCloud, Tweepy packages in Python that are used to perform twitter sentiment analysis. Amongst the various algorithms available, we used the polarity factor for classifying the tweets as either positive, negative or neutral. Whenever a tweet is fed for sentiment analysis, it goes through various data cleaning steps before sentiment analysis. For analyzing a tweet, it is very necessary to know the morph and elements of the tweet. Then, we have to classify the number of tweets based on top ten hashtags on each sentiment. We have achieved that successfully and implemented it using our code.

DEMO VIDEO LINK:
https://drive.google.com/file/d/1LoRIaVjCN1P80MH-9waoXeBs-Se_ehSY/view?usp=sharing

REFERENCE:

[1] GitHub. 2021. *aswintechguy/Machine-Learning-Projects*. [online] Available at: https://github.com/aswintechguy/Machine-Learning-Projects/ . [Accessed 3 May 2021].

[2] A. Sarlan, C. Nadam and S. Basri, "(PDF) Twitter sentiment analysis", *ResearchGate*, 2014. [Online]. Available: https://www.researchgate.net/publication/301408174_Twitter_sentiment_analysis . [Accessed: 03- May- 2021].

[3] "Use Cases, Tutorials, & Documentation", *Developer.twitter.com*, 2021. [Online]. Available: https://developer.twitter.com/en . [Accessed: 03- May- 2021].

[4] V. Sahayak, V. Shete and A. Pathan, "Sentiment Analysis on Twitter Data", *https://www.ijirae.com/* , 2015. [Online]. Available: https://www.ijirae.com/volumes/Vol2/iss1/28.JACS10092.pdf . [Accessed: 04- May- 2021].