1. INTRODUCTION

For my project, I coducted sentiment analysis on Tweets regarding a covid19 pandemic by multinomialNB classifier along with Random Forest classifier (as a pipeline).Through this project I examined the relationship between severity of virus and the public reaction to it particularly social media.In doing so,I hoped to develop an understanding how the public response through out the world.

1.1 OVERVIEW

This report is based upon some basic concepts of Natural Lang. Processing (NLP) that is used to acquire the sentiment analysis of tweet. NLP recently gained pretty much attention in order to analyse human lang. computationally. NLP covers a wide variety of feild like machine translation, spam ham detection, summarization, information extraction, sentiment analysis, and auto-predict, speech recognition etc.
In this report we explore how sentiment mining in social media can be exploited to determine the public reaction on global pandemic and how we encounter this pandemic with our created data.
such situation can help the needed people during an emergency sitution.

1.2 PURPOSE

The objective and main aim of this project is to understand the overview of the wider public opinion behind certain topics.Here we monitored opinion about the covid19 global pandemic.However we can not fully rely on what we get on monitoring since human language are complex.As a human we can understand the actual tone and actual meaning behind the words which depicts another meaning semantically.we can understand the sarcasm tone and anger by a peice of text but for a computer it is difficult to depict and understandand.Therefore We trained and test our model .

2. LITERATURE SURVEY

2.1 Existing problem

PROBLEM STATEMENT :The sentiment analysis of Indians after the extension of lockdown announcements to be analyzed with the relevant #tags on twitter and build a predictive analytics model to understand the behavior of people if the lockdown is further extended.
Also develop a dashboard with visualization of people reaction to the govt announcements on lockdown extension

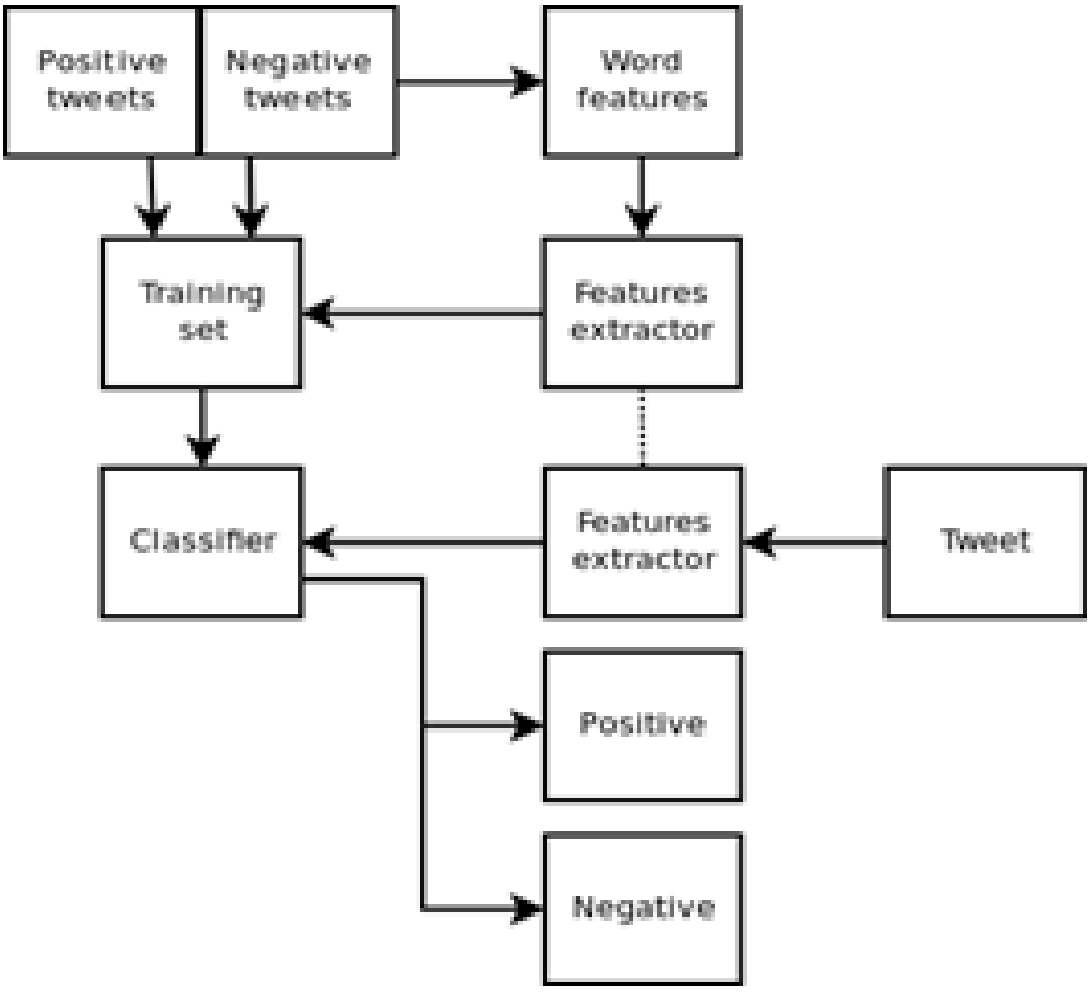
2.2 PROPOSED SOLUTION

The main idealogy is to understand the public opinion on the global pandemic. What kind of Behaviour the public is showing on social media. What kinds of tweet the public is tweeting.Our first step is to fetch live tweets regarding covid19 ,here we are fetching almost 10,000 tweets across all over the world which must be tweeted in english language particularly. Since we are fetching live tweets therfore in a single second we have loads of tweets therefore all the tweets that we are fetching they are almost upto 8 hours earlier from the time of run. Now all these tweets are served as our dataset and all the mining process done on them later we observe visually where the majority lies what are mind set of people regarding Covid19

3. THEORITICAL ANALYSIS

Sentiment analysis is a text analysis method that detects polarity (e.g. a positive or negative opinion) within text, whether a whole document, paragraph, sentence, or clause.Understanding people's emotions is essential for businesses since customers are able to express their thoughts and feelings more openly than ever before. from survey responses to social media conversations, brands are able to listen attentively to their customers, and tailor products and services to meet their needs.

3.1 BLOCK DIAGRAM



Sentiment analysis architecture

3.2 SOFTWARE DESIGN

Project goal: The objective behind this project is to monitor or to understand the public reaction on global pandemic.How the people react on this sitution and what are their behaviour in social media.Monitoring the people's reaction can give outcome of government steps such as to prevent the spread of coronovirus upto what extent people are satisfied with the government steps and what are the changes that they want to introduce or halt some changes.

Dataset: first we need a proper dataset where we divide the dataset into the ratio 1:3 as we want to test and train the same data set but of different values.However In our project we create our own dataset because we have to fetch live tweets. By using the pandas we convert the fetched tweet to a well organized dataset.

Data Preprocessing: Dealing with social media handle we have to handle many things such as emoticons used by user along with the abbreviations,punctuations,special symbols like $,# .In addition to this we have to remove mentioned people an sometimes links when a user provides in his tweet.The purpose of doing so is to deal with proper piece of text in order to get desired outcome.

Adding Subjectivity and Polarity: The next step is to introducing subjectivity and polarity column to our data set. To do so we are using textblob which gives the polarity and subjectivity of the piece of text. we convert every tweet to textblob text then calculate the value of polarity and subjejctivity.

For every text the value of polarity lies in range -1 to +1.

Sentiment calculation:

This is the most important step in which we are calculating sentiment ,we determine the sentiment on the basis of polarity value.If a value

1. lies between 0 and -1 ,then the sentiment is Negative
2. lies between 0 and 1 ,then the sentiment is Positive
3. is equal to 0 ,then the sentiment is Neutral

4. EXPERIMENTAL INVESTIGATION

The wide area of this project is to create a best model that can predict whether the peice of text is positive or negative or may be neutral.The model is trained with the dataframe created by us which consist of tweets along with sentiment we trained our model in this set of data. The whole data is divided into two categories one is for training the model while rest is for testing the model.

To make our data efficiently we have to preprocess them.On preprocessing we just simply remove extra information that a text contain along with removing of words that dont have true meaning like him,her,such as,a , an etc. we remained only with meaningful words.

From our model we are expecting that it shows good result when we predict the sentiment even if we add some piece of text and it can predict the sentiment well.we predict on the basis of keyword for ex. if a piece of code is of negative sentiment ,then the words that is used in that piece of code shows negativity and if that words also come in other piece of text then its value of negativity increases and if we uses that word in a new piece of text the polarity of this text lies between 0 and -1 that is the peice of text is negative.This is what we are expecting from our model.First our test and train data are similar we want to check whether we get 100% efficiency or not and then we introduce model with new test set and calculate the efficiency.

Below are two tables, In first we are predicting the sentiment with same test and train data while the second is one predicting the sentiment with new test set.

	Tweet	Sentiment	Prediction
0	Rightly condem the soulless AynRandian Robotro...	Positive	Positive
1	Going up stairs without using the banister to ...	Negative	Neutral
2	State Health Department reports 149 COVID19 ca...	Neutral	Neutral
3	But are they refundable if COVID19 is still a...	Positive	Positive
4	tiktok instagramreels that’s how who baby jus...	Neutral	Neutral
...	...	...	...
9995	Tested negative. For COVID. That’s all. covid19	Negative	Neutral
9996	interestingly I contracted HepC in prison wit...	Positive	Positive
9997	26 Mississippi lawmakers at capital have COVID...	Positive	Positive
9998	Yettrumpet al. wants to have rallies. Covid be...	Positive	Positive
9999	Today was all about smallpox in myDIseaseAndHi...	Negative	Negative

[10000 rows x 3 columns]

	Tweet	Sentiment	pred
9190	I signed the letter and hope YOU will toNext s...	Neutral	Neutral
1010	A very good tip if get infected by COVID19Dont...	Positive	Positive
5692	So far 96 daycares in Iowa have had confirmed ...	Positive	Positive
5335	Goodness It's like grilling Gestapostyle rathe...	Positive	Positive
8848	Using social and behavioural science to suppor...	Positive	Neutral
...	...	...	...
3484	The study was inconclusive about whether emplo...	Neutral	Neutral
6736	Rubbish very little to do with luck how is tra...	Negative	Negative
8730	"A growing number of Houston residents are dyi...	Neutral	Neutral
1008	People are sharing their postCOVID19 symptoms ...	Neutral	Positive
9963	Excellent time to remind folks that you need ...	Positive	Positive

[3000 rows x 3 columns]

5.FLOWCHART

4.6 Flow Chart

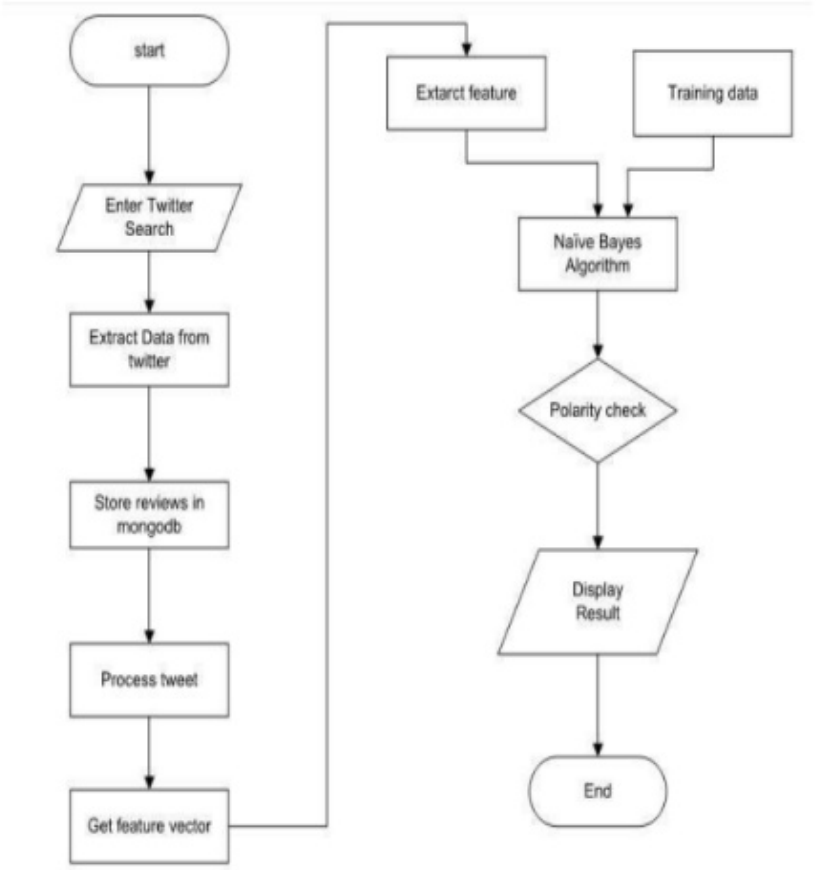
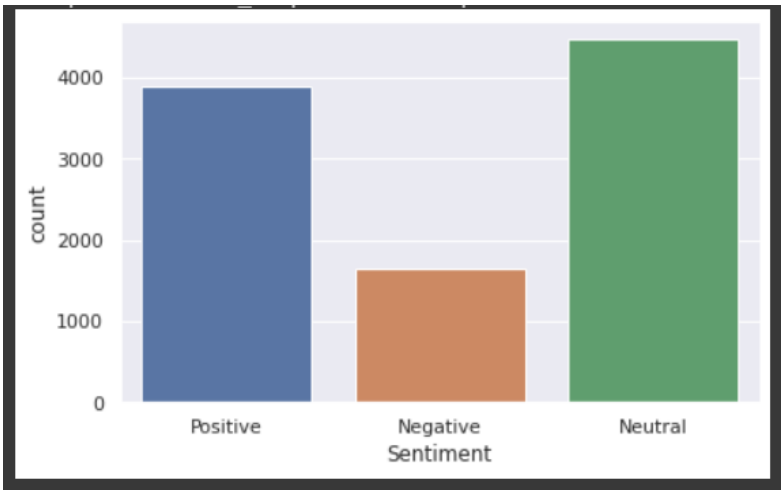


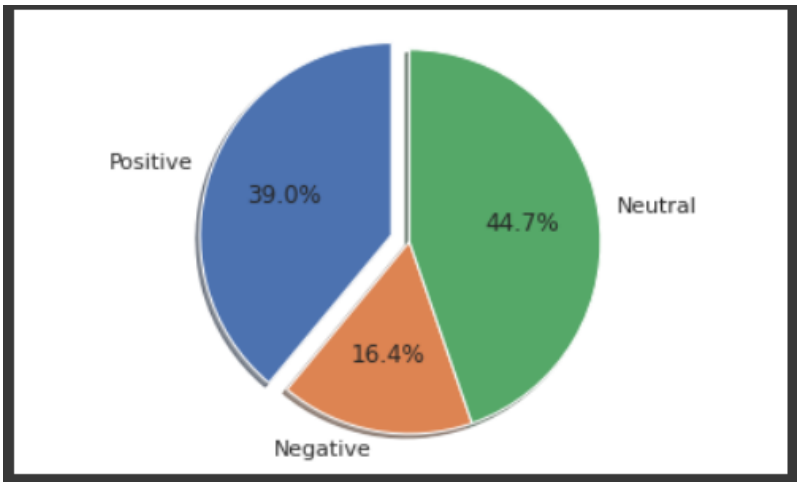
Fig 4.6: "Flow Chart Diagram"

6.RESULT

we successfully perform sentiment analysis and visualised it pretty well. below are the screenshots of visulisation from a code.

Hopefully we also design a model to predict the sentiment by using MultinomialNB and random forest classifier.the last screenshot consist of the approximation of machine learning model.





	precision	recall	f1-score	support
Negative	0.85	0.34	0.48	463
Neutral	0.70	0.97	0.82	1336
Positive	0.87	0.70	0.78	1201
accuracy			0.77	3000
macro avg	0.81	0.67	0.69	3000
weighted avg	0.79	0.77	0.75	3000

7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

By using sentiment analysis, you gauge how customers feel about different areas of your business without having to read thousands of customer comments at once.

If you have thousands or even tens of thousands of survey responses per month, it is impossible for one person to read all of these responses and have an unbiased and consistent measure of customer sentiment. By using sentiment analysis and automating this process, you can easily drill down into different customer segments of your business and get a better understanding of sentiment in these segments

DISADVANTAGES:

While sentiment analysis is useful, we do not believe it is a complete replacement for reading survey responses, as there are often useful nuances in the comments themselves. Where sentiment analysis can help you further is by identifying which of these comments you should read, for example allowing you to focus on the most negative comments.

8.APPLICATIONS

there are various application of sentiment analysis such as :

1.Machine Translation

2.Text Categorization

3.Spam ham Filtering

4.Information Extraction aka IE

5.Indexer ,etc

9.CONCLUSION

sentiment analysis of tweets regarding covid19 has provided a complete understanding of people's behaviour regarding pandemic.By doing this we are capable of monitoring the people's opinion how they are reacting to the sitution whether they support government measures or not what are the steps people are suggesting.we are fetching recent tweets upto 7-8 hour earlier regarding covid-19 and observe what majority of tweets depict whether they spread positivity to the environment or spreading negativity.By fetching recent tweets we actually monitoring recent behaviour of public.

In the next section we test our model with the help of two classifier one is MultinomialNB and random forest classifier.MultinomialNB are tested with same cases which we used to train the model while in random forest classifier we have different set of test and train.By multinomialNB we want to confirm the 100% success which we get it while by using random forest classifier we are getting success the range of 60-80% every time we fetch recent tweets we get different set of success.

10. FUTURE SCOPE

Sentiment analysis is a uniquely powerful tool for businesses that are looking to measure attitudes, feelings and emotions regarding their brand. To date, the majority of sentiment analysis projects have been conducted almost exclusively by companies and brands through the use of social media data, survey responses and other hubs of user-generated content. By investigating and analyzing customer sentiments, these brands are able to get an inside look at consumer behaviors and, ultimately, better serve their audiences with the products, services and experiences they offer.

The future of sentiment analysis is going to continue to dig deeper, far past the surface of the number of likes, comments and shares, and aim to reach, and truly understand, the significance of social media interactions and what they tell us about the consumers behind the screens. This forecast also

predicts broader applications for sentiment analysis – brands will continue to leverage this tool, but so will individuals in the public eye, governments, nonprofits, education centers and many other organizations.

Deeper, Broader Insights from Sentiment Analysis

Sentiment analysis is getting better because social media is increasingly more emotive and expressive. A short while ago, Facebook introduced "Reactions," which allows its users to not just 'Like' content, but attach an emoticon, whether it be a heart, a shocked face, angry face, etc. To the average social media user, this is a fun, seemingly silly feature that gives him or her a little more freedom with their responses. But, to anyone looking to leverage social media data for sentiment analysis, this provides an entirely new layer of data that wasn't available before. Every time the major social media platforms update themselves and add more features, the data behind those interactions gets broader and deeper.

Greater Personalization for Audiences

As a result of deeper and better understanding of the feelings, emotions and sentiments of a brand or organization's key, high-value audiences, members of these audiences will increasingly receive experiences and messages that are personalized and directly related to their wants and needs. Rather than segment markets based on age, gender, income and other surface demographics, organizations can further segment based on how their audience members actually *feel* about the brand or how they use social media. While some people shudder at the thought of companies learning more about them, more exact targeting means that, in the near

future, we will no longer be scratching our head wondering why we see advertisements for products we'd never dream of purchasing. In other words, the spray-and-pray advertising tactics are almost put to rest and there will be a time when every marketing message we see will be relevant and useful to us. Sentiment analysis is going to be a large contributing factor towards achieving this vision.

11.BIBLIOGRAPHY

- https://www.linkedin.com/pulse/future-sentiment-analysis-shahbaz-anwar
- https://www.datacamp.com/community/tutorials/simplifying-sentiment-analysis-python
- https://www.nltk.org/howto/sentiment.html
- https://github.com/shivam019sarathe/covid-tweets-analysis
- https://towardsdatascience.com/introduction-to-data-visualization-in-python-89a54c97fbed
- https://arxiv.org/ftp/arxiv/papers/1601/1601.06971.pdf
- https://www.slideshare.net/anilsth91/tweet-sentiment-analysis-78718965

12. APPENDIX

A. SOURCE CODE

```
import tweepy
from textblob import TextBlob
import pandas as pd
import numpy as np
from wordcloud import WordCloud
import re
```

```
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')

#Twitter api credentials
ConsumerKey       =  'PJHKZkuz3S11A8TALtvN8oCo9'
ConsumerSecret    =
'9Bfnu5S8WdK71O08ZdHd7Wa2H5CG2QpMWSCVkLzrQ6EymKIzfy'
accessToken       =  '1223257823141777408-ChNcuKATurPQaJS0BPymBhdeA4thCi'
accessTokenSecret =  '0vl7PnwSRMqjwshmvrKPKipDqiRMN70JIsYLKTuaD2Q7Z'

#Create the Authentication object
authenticate  = tweepy.OAuthHandler(ConsumerKey,ConsumerSecret)

#set the access token
authenticate.set_access_token(accessToken,accessTokenSecret)

#create the api object while passing in the authenticate information
api = tweepy.API(authenticate, wait_on_rate_limit=True)

#multiple list carries different info. of fetched tweet
tweets_text = []
tweets_id= []
tweets_date = []
tweets_time = []
tweets_retweet_count = []
#fetching live tweets from twitter
try:
# Pulling individual tweets from query
   i=0
   for tweet in tweepy.Cursor(api.search , q='#covid19 -filter:retweets '  , count=100 ,
lang='en').items():
# Adding to list that contains all tweets
     tweets_text.append((tweet.text))
     tweets_id.append(tweet.id)
     tweets_date.append(tweet.created_at.date())
     tweets_time.append(tweet.created_at.time())
     tweets_retweet_count.append(tweet.retweet_count)
```

```
        i=i+1
        if i == 10000:
          break
        else:
          pass
    except BaseException as e:
       print('failed on_status,',str(e))




#creating dataframe of tweet_text
twitter_df_text = pd.DataFrame(tweets_text,columns=['Tweet'])

#df of id
twitter_df_id = pd.DataFrame(tweets_id,columns=['id'])

#df of date
twitter_df_date = pd.DataFrame(tweets_date,columns=['Date'])

#df of time
twitter_df_time = pd.DataFrame(tweets_time,columns=['Time'])

#df of retweet_count
twitter_df_retweet_count =
pd.DataFrame(tweets_retweet_count,columns=['Retweet_count'])



#concatenating all the sub dataframes
twitter_df =
pd.concat([twitter_df_id,twitter_df_date,twitter_df_time,twitter_df_text,twitter_df_retweet
_count],axis=1)
twitter_df.info()
twitter_df.head(5)

#removing mentioned people
```

```
import re
for i in range(len(twitter_df['Tweet'])):
  twitter_df['Tweet'][i] = re.sub(r'@[A-Za-z0-9_A-Za-z0-9]+', '', twitter_df['Tweet'][i])

#to remove special characters and punctuation
spcl_char = [ '#' , '@' , '$'  , '*' , ':' , ';' , '?' , '/' ,  '!' , '|' , '\n' , '¿' , '\n' , ',' ,  '!' , ' " ','(' , ')' ,'+','-','<', '=', '>' ,
'[', ' \ ',']','`',
            '{ ', '|' , '}','~ ']
def char_remove():
  for char in spcl_char:
    twitter_df['Tweet']  = twitter_df['Tweet'].str.replace(char,"")
  return twitter_df['Tweet']
char_remove()
twitter_df.head(5)

#removing link in Tweet if present
twitter_df['Tweet'] = twitter_df['Tweet'].str.replace('http\S+|www.\S+', '', case=False)
twitter_df.head(5)



#to remove ascii characters
twitter_df['Tweet']  = twitter_df['Tweet'].str.replace("["
        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"
                    "]+", ")
twitter_df.head()

#adding polarity and subjectivity
from textblob import sentiments
def getpolarity(text):
   blob = TextBlob(text)
   sent = blob.sentiment.polarity
   return sent

def getsubjectivity(text):
   blob = TextBlob(text)
   sent = blob.sentiment.subjectivity
   return sent
twitter_df['polarity']  = twitter_df['Tweet'].apply(getpolarity)
twitter_df['subjectivity']  = twitter_df['Tweet'].apply(getsubjectivity)
twitter_df.head()

#adding sentiment
def category(text):
  blob1 = TextBlob(text)
  sent = blob1.sentiment.polarity
  if sent>0:
    status = "Positive"
  elif sent==0:
    status = "Neutral"
  else:
    status = "Negative"
  return status
twitter_df['Sentiment']  = twitter_df['Tweet'].apply(category)
twitter_df.head(500)


#VISUALISATION part starts from here
```

```
#Plotting the wordcloud
allwords  = ' '.join([tweets for tweets in twitter_df['Tweet']])
wordcloud = WordCloud(max_font_size=160,width=1200 , height=600,max_words= 200
).generate(allwords)
plt.imshow(wordcloud, cmap = 'plasma',interpolation='bilinear')
plt.axis('off')
plt.show( )


#plotting COUNTER plot
import seaborn as sns
sns.set(style="darkgrid")
sns.countplot(x='Sentiment', data= twitter_df )

#plotting polarity and subjectivity
plt.figure(figsize=(8,6) )
#i is basically a total no. of rows
for j in range(0,i):
  plt.scatter(twitter_df['polarity'],twitter_df['subjectivity'],edgecolor='black',cmap='plasma'
)
plt.xlabel('Polarity')
plt.ylabel('subjectivity')
plt.title('SENTIMENT')
plt.plot()




#creating dataframes that consist of only positive,negative and neutral tweets only
twitter_df_pos = twitter_df.loc[twitter_df['Sentiment'] == 'Positive' ]
twitter_df_neg = twitter_df.loc[twitter_df['Sentiment'] == 'Negative' ]
twitter_df_neut = twitter_df.loc[twitter_df['Sentiment'] == 'Neutral' ]

#resetting index
twitter_df_pos.reset_index(drop=True , inplace=True)
twitter_df_neg.reset_index(drop=True , inplace=True)
```

```
twitter_df_neut.reset_index(drop=True , inplace=True)



#polarity vs subjectivity graph of positive tweets
plt.figure(figsize=(8,6) )


for j in range(0,len(twitter_df_pos)):

plt.scatter(twitter_df_pos['polarity'],twitter_df_pos['subjectivity'],edgecolor='black',cmap=
'plasma' )
plt.xlabel('Polarity')
plt.ylabel('subjectivity')
plt.title('SENTIMENT ANALYSIS OF POSITIVE TWEETS')
plt.plot()

#polarity vs subjectivity graph of negative tweets
plt.figure(figsize=(8,6) )


for j in range(0,len(twitter_df_neg)):

plt.scatter(twitter_df_neg['polarity'],twitter_df_neg['subjectivity'],edgecolor='black',cmap='
plasma' )
plt.xlabel('Polarity')
plt.ylabel('subjectivity')
plt.title('SENTIMENT ANALYSIS OF NEGATIVE TWEETS')
plt.plot()

#polarity vs subjectivity graph of Neutral tweets
plt.figure(figsize=(8,6) )


for j in range(0,len(twitter_df_pos)):
```

```
plt.scatter(twitter_df_neut['polarity'],twitter_df_neut['subjectivity'],edgecolor='black',cmap
='plasma' )
plt.xlabel('Polarity')
plt.ylabel('subjectivity')
plt.title('SENTIMENT ANALYSIS OF NEUTRAL TWEETS')
plt.plot()

#plotting pie chart

#calculating the percentage of pos,neg and neut among 10000 tweets
percent_pos  = (len(twitter_df_pos)/len(twitter_df))*100
percent_neg  = (len(twitter_df_neg)/len(twitter_df))*100
percent_neut  = (len(twitter_df_neut)/len(twitter_df))*100

import matplotlib.pyplot as plt

# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = 'Positive', 'Negative', 'Neutral'
sizes = [percent_pos,percent_neg,percent_neut]
explode = (0.1, 0, 0)  # only "explode" the 1st slice (i.e. 'Positive')

fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)

ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()

import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
def text_process(tweet):
  return [word for word in tweet.split() if word.lower() not in stopwords.words('english')]
twitter_df['Tweet'].apply(text_process)
twitter_df.head()
```

```
#fitting Tweet data
from sklearn.feature_extraction.text import CountVectorizer,TfidfTransformer
bow_transfromer = CountVectorizer(analyzer=text_process).fit(twitter_df['Tweet'])


Tweet_bow = bow_transfromer.transform(twitter_df['Tweet'])

tfidf_transformer = TfidfTransformer().fit(Tweet_bow)
tweet_tfidf = tfidf_transformer.transform(Tweet_bow)

from sklearn.naive_bayes import MultinomialNB
tweet_detect = MultinomialNB().fit(tweet_tfidf,twitter_df['Sentiment'])

tweet_pred = tweet_detect.predict(tweet_tfidf)
pred_df  = pd.DataFrame(tweet_pred,columns=['Prediction'])

net_df = pd.concat([twitter_df['Tweet'],twitter_df['Sentiment'],pred_df],axis = 1)
print(net_df )

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
t_train,t_test,s_train,s_test =
train_test_split(twitter_df['Tweet'],twitter_df['Sentiment'],test_size = 0.3)


from sklearn.pipeline import Pipeline
pipeline = Pipeline([
                ('bow',CountVectorizer(analyzer=text_process)),
                ('tfidf', TfidfTransformer()),
                ('classifier', RandomForestClassifier())
])


pipeline.fit(t_train,s_train)


cc = pd.concat([t_test,s_test],axis=1)
```

```
prediction = pipeline.predict(t_test)
prediction = list(prediction)
cc['pred'] = prediction


print(cc)


from sklearn.metrics import  classification_report
print(classification_report(s_test,prediction))

pip install nbconvert

$ jupyter nbconvert --to HTML Copy_of_2020_part_3point1.ipynb
```