VISVESWARAYA TECHNOLOGICAL UNIVERSITY

BELGAUM, KARNATAKA

MINOR-PROJECT-II REPORT ON

"TWITTER SENTIMENT ANALYSIS"

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF ENGINEERING IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

USN: 2SD17CS052 NAME: NIKHIL ELIGAR USN: 2SD17CS027 NAME: GOUTAM TERDAL

Under the Guidance of

Prof. / Dr. A.A.QAZI
Dept. of CSE, SDMCET, Dharwad

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
S.D.M. COLLEGE OF ENGINEERING & TECHNOLOGY,
DHARWAD-580002

2021

S.D.M COLLEGE OF ENGINEERING & TECHNOLOGY,

DHARWAD –580002

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Minor-Project-2 work and presentation entitled **"TWITTER SENTIMENT ANALYSIS"** *is a bonafide work carried out by* **NIKHIL.ELIGAR (2SD17CS052)** and **GOUTAM.TERDAL (2SD17CS027),** *students of S. D. M. College of Engineering & Technology, Dharwad, in partial fulfillment for the award of* **Bachelor of Engineering in Computer Science and Engineering of** ***Vishveshwarya Technological University, Belgaum****, during the year 2020-2021. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The Minor-Project-2 has been approved, as it satisfies the academic requirements in respect of project report prescribed for the said degree.*

Dr. / Prof. A.A.QAZI	**Dr. U P Kulkarni**
Project Guide	HOD-CSE

ABSTRACT

Sentiment analysis or opinion mining is the computational study of people's opinions, sentiments, attitudes, and emotions expressed in written language. It is one of the most active research areas in natural language processing and text mining in recent years. Its popularity is mainly due to two reasons. First, it has a wide range of applications because opinions are central to almost all human activities and are key influencers of our behaviour. Whenever we need to make a decision, we want to hear others' opinions. Second, it presents many challenging research problems, which had never been attempted before the year 2000. Part of the reason for the lack of study before was that there was little opinionated text in digital forms. The research has also spread outside of computer science to management sciences and social sciences due to its importance to business and society as a whole.

PROBLEM STATEMENT

The problem in sentiment analysis is classifying the subjectivity and polarity of a given text at the document, sentence, or feature/aspect level and showing it through graphs. Whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral.

INTRODUCTION

We will use Twitter to perform sentiment analysis of the written text. We will use Twitter in this example but this can be also used in a business context to analyse different social media accounts, reviews of your company, reviews of your products and services, analysing support tickets, emails or free text from surveys to get an idea of the mood that is coming from people engaging with you and your business online.

What is sentiment analysis?

Sentiment Analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. It's also known as opinion mining, deriving the opinion or attitude of a speaker.

Why sentiment analysis?

Business: In marketing field companies use it to develop their strategies, to understand customers' feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don't buy some products.

Politics: In political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well!

Public Actions: Sentiment analysis also is used to monitor and analyse social phenomena, for the spotting of potentially dangerous situations and determining the general mood of the blogosphere.

Advertising on social media is one of the most important strategies a company or an organization opts to promote its product. It has always been so important for these organizations to know how their products are doing or how people are reacting to it. The idea we had in our mind when we started was to be able to predict the polarity of these products or a personality so we can tell if the strategy that they had opted for was successful. Twitter happens to be a great platform for people to go and post about anything that is how they feel or what they think about a certain product or personality. It's indeed a great tool to check what people think about a topic. Hence Sentiment Analysis was the tool fit for this job. It has emotions, attitudes or assessment which considers us that is as we think as humans. Now this is not necessarily that simple always. Contents can be written in different contexts many times.

Twitter proves to be a great source of data for analysing because:

- The API is clean and comes with rich developer tools.
- The data is rich in information and has a data format fit for analysis.
- Twitter data is accessible to anyone with fair usage rights.

METHOD USED:

We have used baseline method and in-built classifiers from NLTK (Natural Language Toolkit)

NLTK: - The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language NLTK supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities.

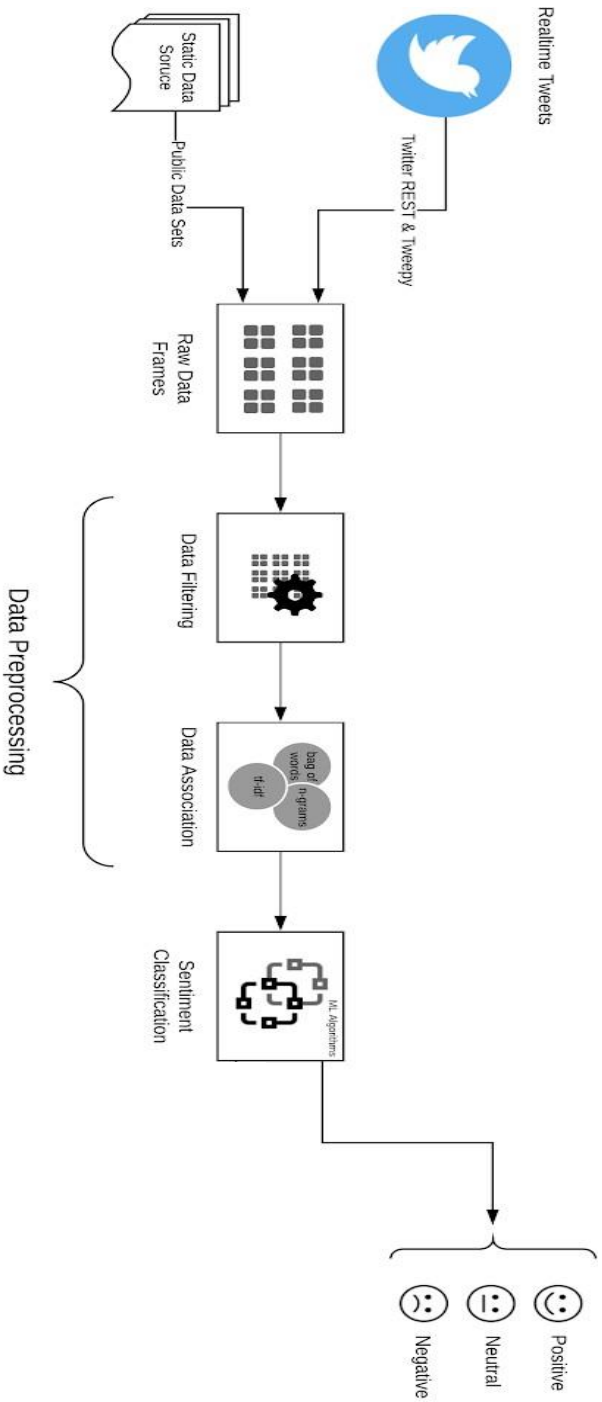
LITERATURE SURVEY

Sentiment analysis of in the domain of micro-blogging is a relatively new research topic so there is still a lot of room for further research in this area. Decent amount of related prior work has been done on sentiment analysis of user reviews, documents, web blogs/articles and general phrase level sentiment analysis. These differ from twitter mainly because of the limit of 280 characters per tweet which forces the user to express opinion compressed in very short text. The best results reached in sentiment classification use supervised learning techniques such as Naive Bayes and Support Vector Machines, but the manual labelling required for the supervised approach is very expensive. Some work has been done on unsupervised and semi-supervised approaches, and there is a lot of room of improvement. Various researchers testing new features and classification techniques often just compare their results to base-line performance. There is a need of proper and formal comparisons between these results arrived through different features and classification techniques in order to select the best features and most efficient classification techniques for particular applications.

We follow these 3 major steps in our program:

- Authorize twitter API client.
- Make a GET request to Twitter API to fetch tweets for a particular query.
- Parse the tweets. Classify each tweet as positive, negative or neutral.

DETAILED DESIGN



PROJECT SPECIFIC REQUIREMENTS

→ Linux Operating System / Windows
→ Python Platform (Google-Colab, Anaconda, Spyder, Juypter)
→ NLTK Packages
→ Modern Web Browser
→ Twitter API, Google API

IMPLEMENTATION and RESULT

1. **Load Twitter API**:

The first step is to register in the twitter application developer's portal and get the authorization. We need:

- consumerKey = 'xxxxxxxxxx'
- consumerSecret ='xxxxxxxxxx'
- accessToken = 'xxxxxxxxxx'
- accessTokenSecret = 'xxxxxxxxx'

In order to fetch tweets through Twitter API, one needs to register an App through their twitter account. Follow these steps for the same:

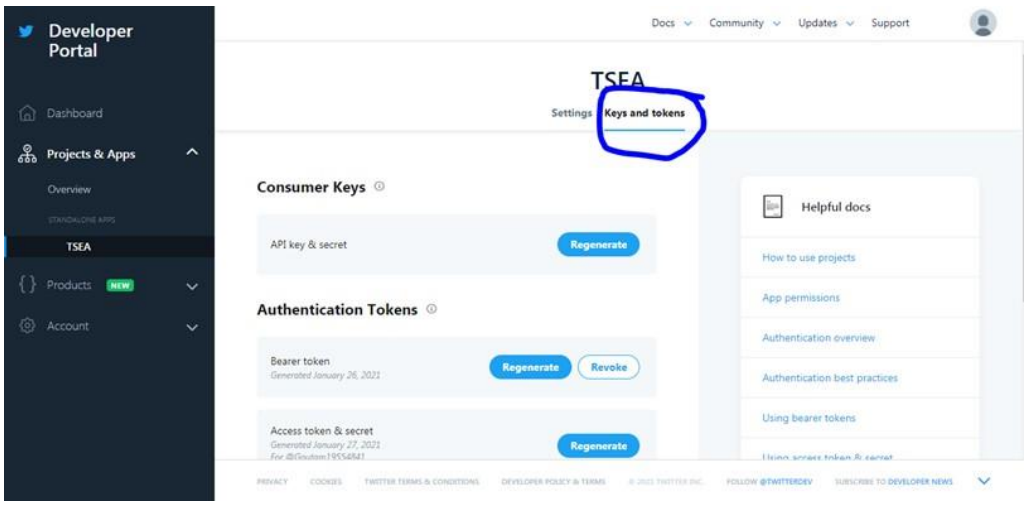
1. Open this LINK and click the button: 'Create New App'

2. Fill the application details.

3. Once the app is created, you will be redirected to the app page.

4. Open the 'Keys and Access Tokens' tab.

5. Copy 'Consumer Key', 'Consumer Secret', 'Access token' and 'Access Token Secret'.



2. **Load packages**:

Next import packages from google colab which are pre-installed in it We need:

- **Tweepy**: It is an open source Python package that gives you a very convenient way to access the Twitter API with Python. Tweepy includes a set of classes and methods that represent Twitter's models and API endpoints.

- **TextBlob**: It is a Python library for processing textual data. It provides a simple API for diving into common natural language processing(NLP) tasks such as part-of-speech tagging, noun phrases extraction, sentiment analysis, classification, translation and more.

- **WordCloud**: A word cloud (also called tag cloud or weighted list) is a visual representation of text data. Words are usually single words, and the importance of each is shown with font size or color. The more a specific word appears in the text, the bigger and bolder it appears in the word cloud.

- **Pandas**: It is a Python package providing fast, flexible and expressive data structures designed to make working with "relational" or "labelled" data both easy and intutuive. Pandas is mainly used for data analysis. It allows importing data from various file formats such as comma-seperated values, JASON, SQL, excel.

- **NumPy**: NumPy is the fundamental package for scientific computing in Python. ... NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.

- **re**: A regular expression (or re) specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing).

- **Matplotlib.pyplot**: Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc. In matplotlib.

```
#Description : This is a sentiment analysis program that parses the tweets fetched from Twitter using Python

#Import the libraries
import tweepy
from textblob import TextBlob
from wordcloud import WordCloud
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
```

Next, we will write Twitter API Credentials in our code

```
#Twitter Api Credentials
consumerKey = 'gAt1Ll9WyL1HFX7hSlMdTwlkQ'
consumerSecret = 'e2dA2yB5VGx7g9ps2B2YpZLdHJfXYNrcYhv5pAAyLGfdmzOdLh'
accessToken = '1352923696902463489-ED7CyO9mbYOLbqP05gILSn4zUhSEec'
accessTokenSecret = 'yb0aJxeKtADcWjPHF7KzBM7H0A6hWRsRUDNfhsEd9nCKq'
```

3. **<u>Creating Authentication object</u>**:

Next, we will use our keys and the OAuth function from the library 'tweepy' to create an authentication object. This object will include both Consumer Keys. We will then also include our Token Keys, which are AccessToken and AccessTokenSecret.

```
import tweepy

#Create the authentication object
authenticate = tweepy.OAuthHandler(consumerKey, consumerSecret)

#Set the access token and access token secret
authenticate.set_access_token(accessToken, accessTokenSecret)

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

The wait_on_rate_limit determines whether or not to automatically wait for rate limits to replenish. In our case, we have set this to True, so it will automatically wait.

4. **<u>Getting Tweets and Data Framing Tweets</u>**:

We will retrieve recent 100 Tweets from a twitter account that a user desires and print only first 5 Tweets. Here I have retrieved “Bill Gates” Tweets.

```
# Extract 100 tweets from the twitter user
posts = api.user_timeline(screen_name = "Billgates", count = 100, lang = "en", tweet_mode = "extended")

#Print the last 5 tweets from the account
print("Show the 5 recent tweets: \n")
for tweet in posts[0:5]:
   print(tweet.full_text + '\n')
```

Here is the result of first 5 Tweets printed.

```
Show the 5 recent tweets:
I am truly grateful for his wisdom and leadership, and most of all for his enduring friendship. Warren will continue to inspire our foundation as we work to fight
I will always have a deep sense of accountability to Warren, paying close attention to the data to track our progress and identify areas where we can do better. E
Should you pick your nose?
You can read an excerpt from Matt Richtel's fascinating book about the immune system on my blog to find the answer: https://t.co/cgfc1aUzPA https://t.co/Uc7vqT2BJ
The journalist @ElizKolbert has created a fascinating beat for herself covering humanity's impact on nature and our attempts to control it. Her latest book is a g
The persistence of countless Rotarians gives me hope that we can achieve a polio-free world. As they close the #Rotary21 Convention, I'd like to thank @Rotary for
```

Now, we will create data frame for the printed tweets using pandas library

```
# Create a dataframe with a column called Tweets
import pandas as pd
df = pd.DataFrame([tweet.full_text for tweet in posts], columns = ['Tweets'])

# Show the first 5 rows of data
df.head()
```

	Tweets
0	I am truly grateful for his wisdom and leaders...
1	I will always have a deep sense of accountabil...
2	Should you pick your nose?\n\nYou can read an ...
3	The journalist @ElizKolbert has created a fasc...
4	The persistence of countless Rotarians gives m...

5. **Cleaning tweets**:

Before we carry out this analysis, we must preprocess our tweets as you can see from the screenshot, many tweets include URL links, # (hashtags), @mentions, RT mentions. If we include this in our analysis, we will receive inaccurate results. Therefore, we must clean our text by removing these leaving us with only the raw text/tweet. We will be using the regular expression operation (re) to clean our text.

```
#Create a function to clean the tweets

def cleanTxt(text):
  text = re.sub(r'@[A-Za-z0-9]+', '', text)#Removing @mentions
  text = re.sub(r'#', '', text)#Removing '#' has tag
  text = re.sub(r'RT[\s]+', '', text)#Removing RT
  text = re.sub(r'https?:\/\/\S+', '', text)#Removing hyperlink
  return text

#clean the tweets
df['Tweets']= df['Tweets'].apply(cleanTxt)

#show the cleaned tweets
df
```

When we run this, a table should be formed that contains our cleaned tweets.

	Tweets
0	I am truly grateful for his wisdom and leaders...
1	I will always have a deep sense of accountabil...
2	Should you pick your nose?\n\nYou can read an ...
3	The journalist has created a fascinating beat...
4	The persistence of countless Rotarians gives m...
...	...
95	I enjoyed spending time with recently in Seat...
96	I'm thrilled to join , _RDG, , , , , , and Mo...
97	: In 2020, global health went local. \n\nCOVID...
98	: Only 3% of Black students learn computer sci...
99	: After being sworn in this morning, I'm honor...

100 rows × 1 columns

6. **Getting Subjectivity and Polarity**:

Now we have cleaned the text, we will be using the library TextBlob and the sentiment function to calculate polarity and subjectivity. We will create two functions (one for subjectivity and polarity). These will calculate the values for each tweet and then apply this function to all the tweets, representing each value as additional columns.

Subjectivity (how subjective or opinionated the text is — a score of 0 is fact, and a score of +1 is very much an opinion) and the other to get the tweets called Polarity (how positive or negative the text is, — score of -1 is the highest negative score, and a score of +1 is the highest positive score).

```
#create a function to get subjectivity
def getSubjectivity(text):
  return TextBlob(text).sentiment.subjectivity

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

#Create two columns 'Subjectivity' and 'Polarity'
df['Subjectivity'] = df['Tweets'].apply(getSubjectivity)
df['Polarity'] = df['Tweets'].apply(getPolarity)

#Show the new dataframe with columns 'Subjectivity' and 'Polarity'
df
```

	Tweets	Subjectivity	Polarity
0	I am truly grateful for his wisdom and leaders...	0.500000	0.318182
1	I will always have a deep sense of accountabil...	0.450000	0.250000
2	Should you pick your nose?\n\nYou can read an ...	0.850000	0.700000
3	The journalist has created a fascinating beat...	0.783333	0.633333
4	The persistence of countless Rotarians gives m...	0.500000	0.000000
...	...	...	...
95	I enjoyed spending time with recently in Seat...	0.475000	0.250000
96	I'm thrilled to join , _RDG, , , , , , and Mo...	0.700000	0.600000
97	: In 2020, global health went local. \n\nCOVID...	0.000000	0.000000
98	: Only 3% of Black students learn computer sci...	0.657778	-0.002222
99	: After being sworn in this morning, I'm honor...	0.000000	0.000000

100 rows × 3 columns

7. **Word Cloud**:

Let's see how well the sentiments are distributed. A good way to accomplish this task is by understanding the common words by plotting word clouds.

A word cloud (also known as text clouds or tag clouds) is a visualization, the more a specific word appears in the text, the bigger and bolder it appears in the word cloud.

Let's visualize all the words in the data using the word-cloud plot.

```
[ ]  #plot the Word Cloud
     allWords = ' '.join( [twts for twts in df['Tweets']] )
     wordCloud = WordCloud(width = 400, height = 200, random_state = 21, max_font_size = 100).generate(allWords)

     plt.imshow(wordCloud, interpolation = "bilinear")
     plt.axis('off')
     plt.show()
```

8. Analysing Tweets to positive, negative and neutral:

In our table with all our tweets and values for subjectivity and polarity, we will create another column to show other users which tweets are positive, negative, or neutral.

```
#Create a function to compute negative (-1), neutral(0) and positive(+1) 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
```

	Tweets	Subjectivity	Polarity	Analysis
0	I am truly grateful for his wisdom and leaders...	0.500000	0.318182	Positive
1	I will always have a deep sense of accountabil...	0.450000	0.250000	Positive
2	Should you pick your nose?\n\nYou can read an ...	0.850000	0.700000	Positive
3	The journalist has created a fascinating beat...	0.783333	0.633333	Positive
4	The persistence of countless Rotarians gives m...	0.500000	0.000000	Neutral
...	...	...	...	...
95	I enjoyed spending time with recently in Seat...	0.475000	0.250000	Positive
96	I'm thrilled to join , _RDG, , , , , , and Mo...	0.700000	0.600000	Positive
97	: In 2020, global health went local. \n\nCOVID...	0.000000	0.000000	Neutral
98	: Only 3% of Black students learn computer sci...	0.657778	-0.002222	Negative
99	: After being sworn in this morning, I'm honor...	0.000000	0.000000	Neutral

100 rows × 4 columns

9. **<u>Printing positive and negative Tweets</u>**:

Print the positive tweets in ascending order. The most positive tweet is the #1 tweet.

```
# printing positive tweets
print('Printing positive tweets:\n')
j=1
sortedDF = df.sort_values(by=['Polarity'])  #Sort the tweets
for i in range(0, sortedDF.shape[0] ):
  if( sortedDF['Analysis'][i] == 'Positive'):
    print(str(j) + ') '+ sortedDF['Tweets'][i])
    print()
    j = j+1
```

```
Printing positive tweets:

1) I am truly grateful for his wisdom and leadership, and most of all for his enduring friendship. Warren will continue to inspire our foundation as we work to

2) I will always have a deep sense of accountability to Warren, paying close attention to the data to track our progress and identify areas where we can do bet

3) Should you pick your nose?

You can read an excerpt from Matt Richtel's fascinating book about the immune system on my blog to find the answer:

4) The journalist  has created a fascinating beat for herself covering humanity's impact on nature and our attempts to control it. Her latest book is a good re

5) I'm almost always interested in books about American presidents, and I especially loved A Promised Land. It's a fascinating look at what it's like to steer

6) This book gave me a deeper, more nuanced appreciation for the system that is at the core of humanity's fight against COVID-19 and everything our foundation's
```

Print the negative tweets in descending order. The most negative tweet is the #1 tweet.

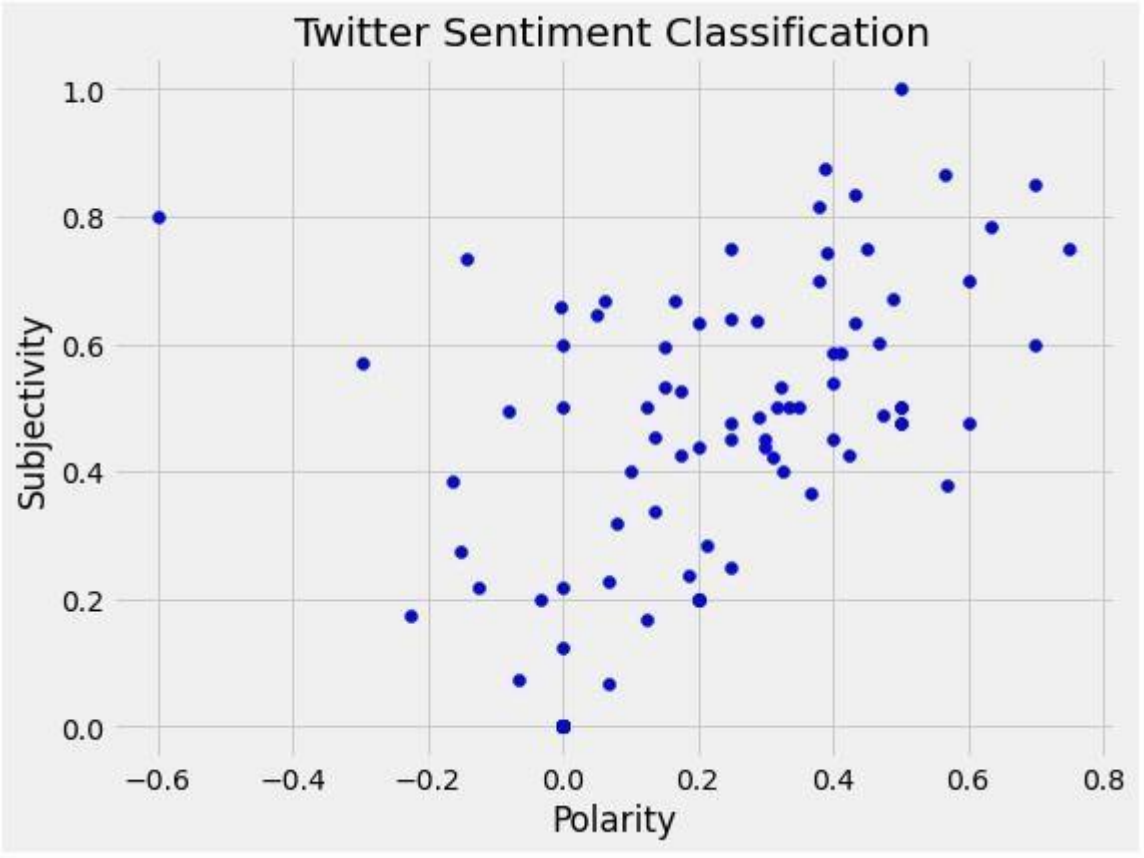
```
# Printing negative tweets
print('print negative tweets:\n')
j=1
sortedDF = df.sort_values(by=['Polarity'],ascending=False) #Sort the tweets
for i in range(0, sortedDF.shape[0]):
  if (sortedDF['Analysis'][i] == 'Negative'):
    print(str(j) + ') '+sortedDF['Tweets'][i])
    print()
    j=j+1
```

```
print negative tweets:

1) Communities of color have been hit hard by COVID-19. One of the reasons why parts of the medical system often fail Black and brown people is because it's not

2) Dr. Stephaun Wallace () has spent the last year helping make COVID-19 vaccines work for everybody. Stephaun and his colleagues at  are working to reach the pe

3) It's deeply unfair that the people who contribute the least to climate change will suffer the worst from its effects:

4) : Over the past few weeks health workers in Ethiopia ET, Nigeria NG, Sudan SD and the Philippines PH were vaccinated against COVI…

5) For decades, Australian researcher Ruth Bishop led global efforts to identify and combat rotavirus. Her life is a reminder of the importance of scientific rese

6) : Black folks have questions about the COVID-19 vaccine. I sat down w/ Black healthcare workers &amp;amp; they answered my questions.…
```

10. **<u>Plotting Subjectivity and Polarity</u>**:

Plot the polarity and subjectivity as a scatter plot. It looks like the majority of the tweets are positive, as many of the points are on the right side of the polarity at value 0.00.

```
#plot the subjectivity and polarity
plt.figure(figsize=(8,6))
for i in range(0, df.shape[0]):
  plt.scatter(df['Polarity'][i], df['Subjectivity'][i], color='blue')

plt.title('Twitter Sentiment Classification')
plt.xlabel('Polarity')
plt.ylabel('Subjectivity')
plt.show()
```



11. **Printing percentages of positive and negative Tweets**:

Positive tweets: 68%

```
[21] # Print the percentage of positive tweets
     ptweets = df[df.Analysis == 'Positive']
     ptweets = ptweets ['Tweets']
     ptweets

     round ( (ptweets.shape[0] / df.shape[0]) * 100 , 1)
```

```
68.0
```

Negative tweets: 11%

```
# Print the percentage of Negaive tweets
ntweets = df[df.Analysis == 'Negative']
ntweets = ntweets['Tweets']
ntweets

round( (ntweets.shape[0] / df.shape[0]) * 100, 1)
```

```
11.0
```

12. **Showing the value counts**:

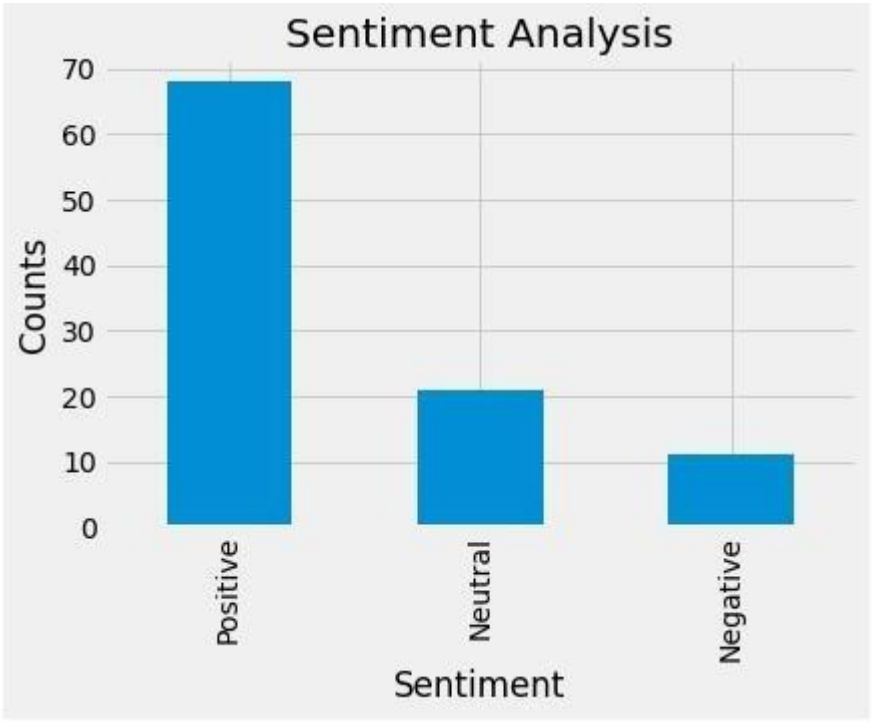
```
#show the value count
df['Analysis'].value_counts()
```

```
Positive    68
Neutral     21
Negative    11
Name: Analysis, dtype: int64
```

13. **<u>Visually showing the value counts</u>**:

Finally, now that we have the number of tweets that are +ve, -ve or neutral, we can create a bar chart to represent this. This is will be done in a very similar way to how we represented the subjectivity against polarity scatter graph.

```
#Plot and visualize the count
plt.title('Sentiment Analysis')
plt.xlabel('Sentiment')
plt.ylabel('Counts')
df['Analysis'].value_counts().plot(kind='bar')
plt.show()
```



CONCLUSION AND FUTURE SCOPE

Doing sentiment analysis of the tweets enabled us to calculate numerical values of subjectivity and polarity.

This could help us to understand better this Twitter account in terms of the language that is being used.

Combining this with additional information about likes and comments can be very useful from marketing point of view and can enable us to find some correlation between subjectivity, polarity and the engagement of the users for a specified Twitter account.

IMPROVEMENTS:

From the baseline, the goal is to improve the accuracy of the classifier in order to determine better which tweet is positive, negative or neutral. There are several ways of doing this and we present only few possible improvements.