Write-up	Correctness of Program	Documentation of Program	Viva	Timely Completion	Total	Dated Sign of Subject Teacher
4	4	4	4	4	20	

Expected Date of Completion:..... Actual Date of Completion:.....

Group C

Assignment No: 2

Title of the Assignment:

Use the following dataset and classify tweets into positive and negative tweets.
https://www.kaggle.com/ruchi798/data-science-tweets

Objective of the Assignment: Students should be able to classify the tweets into positive and negative tweets.

Prerequisite:

1. **Basic of Python Programming**
2. **Basic of Text Processing.**

Contents for Theory:

1. **Step 1: Data Collection**
2. **Step 2: Sentiment Analysis**
3. **Step 3: Visualization.**

We will begin by scraping and storing Twitter data. We will then classify the Tweets into positive, negative, or neutral sentiment with a simple algorithm. Then, we will build charts using Plotly and Matplotlib to identify trends in sentiment.

Step 1: Data collection

Command -

```
import pandas as pd
df = pd.read_csv('/content/data_visualization.csv')
```

Output -

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshe
ll.py:2882: DtypeWarning: Columns (22,24) have mixed types.Specify
dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Let's now take a look at some of the variables present in the data frame:

Command -

```
df.info()
```

Output -

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 33590 entries, 0 to 33589
Data columns (total 36 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               33590 non-null  int64
 1   conversation_id  33590 non-null  int64
 2   created_at       33590 non-null  object
 3   date             33590 non-null  object
 4   time             33590 non-null  object
 5   timezone         33590 non-null  int64
 6   user_id          33590 non-null  int64
 7   username         33590 non-null  object
 8   name             33590 non-null  object
 9   place            85 non-null     object
 10  tweet            33590 non-null  object
 11  language         33590 non-null  object
 12  mentions         33590 non-null  object
 13  urls             33590 non-null  object
 14  photos           33590 non-null  object
 15  replies_count    33590 non-null  int64
 16  retweets_count   33590 non-null  int64
 17  likes_count      33590 non-null  int64
 18  hashtags         33590 non-null  object
 19  cashtags         33590 non-null  object
 20  link             33590 non-null  object
 21  retweet          33590 non-null  bool
 22  quote_url        1241 non-null   object
 23  video            33590 non-null  int64
 24  thumbnail        9473 non-null   object
 25  near             0 non-null      float64
```

```
 26  geo              0 non-null      float64
 27  source           0 non-null      float64
 28  user_rt_id       0 non-null      float64
 29  user_rt          0 non-null      float64
 30  retweet_id       0 non-null      float64
 31  reply_to         33590 non-null  object
 32  retweet_date     0 non-null      float64
 33  translate        0 non-null      float64
 34  trans_src        0 non-null      float64
 35  trans_dest       0 non-null      float64
d    types: bool(1), float64(10), int64(8), object(17)
memory usage: 9.0+ MB
```

The data frame has 35 columns. The most main variables we will be using in this analysis are date and tweet. Let's take a look at a sample Tweet in this dataset, and see if we can predict whether it is positive or negative:

Command -

```
df['tweet'][10]
```

Output -

```
We are pleased to invite you to the EDHEC DataViz Challenge grand
final for a virtual exchange with all Top 10 finalists to see how
data visualization creates impact and can bring out compelling
stories in support of @UNICEF's mission. https://t.co/Vbj9B48VjV
```

Step 2: Sentiment Analysis

The Tweet above is clearly positive. Let's see if the model is able to pick up on this, and return a positive prediction. Run the following lines of code to import the NLTK library, along with the SentimentIntensityAnalyzer (SID) module.

Command -

```
import nltk
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()
import re
import pandas as pd
import nltk
nltk.download('words')
words = set(nltk.corpus.words.words())
```

The SID module takes in a string and returns a score in each of these four categories - positive, negative, neutral, and compound. The compound score is calculated by normalizing

the positive, negative, and neutral scores. If the compound score is closer to 1, then the Tweet can be classified as positive. If it is closer to -1, then the Tweet can be classified as negative. Let's now analyze the above sentence with the sentiment intensity analyzer.

Command -

```
sentence = df['tweet'][0] sid.polarity_scores(sentence)
['compound']
```

The output of the code above is 0.7089, indicating that the sentence is of positive sentiment. Let's now create a function that predicts the sentiment of every Tweet in the dataframe, and stores it as a separate column called 'sentiment.' First, run the following lines of code to clean the Tweets in the data frame:

Command –

```
def cleaner(tweet):
    tweet = re.sub("@[A-Za-z0-9]+","",tweet) #Remove @ sign
    tweet = re.sub(r"(?:\@|http?\://|https?\://|www)\S+", "",
tweet) #Remove http links
    tweet = " ".join(tweet.split())
    tweet = tweet.replace("#", "").replace("_", " ") #Remove
hashtag sign but keep the text
    tweet = " ".join(w for w in nltk.wordpunct_tokenize(tweet)
         if w.lower() in words or not w.isalpha())
    return tweet
df['tweet_clean'] = df['tweet'].apply(cleaner)
```

Now that the Tweets are cleaned, run the following lines of code to perform the sentiment analysis:

Command -

```
word_dict =
{'manipulate':-1,'manipulative':-1,'jamescharlesiscancelled':-1,'j
amescharlesisoverparty':-1,

'pedophile':-1,'pedo':-1,'cancel':-1,'cancelled':-1,'cancel
culture':0.4,'teamtati':-1,'teamjames':1,
            'teamjamescharles':1,'liar':-1}
import nltk
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()
sid.lexicon.update(word_dict)
list1 = []
for i in df['tweet_clean']:
```

```
    list1.append((sid.polarity_scores(str(i)))['compound'])
```

The word_dict created above is a dictionary of custom words I wanted to add into the model. Words like 'teamjames' mean that people's sentiment around James Charles is positive, and that they support him. The dictionary used to train the sentiment intensity analyzer wouldn't already have these words in them, so we can update it ourselves with custom words.
Now, we need to convert the compound scores into categories - 'positive', 'negative', and 'neutral.'

Command -

```
df['sentiment'] = pd.Series(list1)
def  sentiment_category(sentiment):
    label = ''
    if(sentiment>0):
        label = 'positive'
    elif(sentiment == 0):
        label = 'neutral'
    else:
        label = 'negative'
    return(label)
df['sentiment_category'] =
df['sentiment'].apply(sentiment_category)
```

Let's take a look at the head of the data frame to ensure everything is working properly:

Command -

```
df  =  df[['tweet','date','id','sentiment','sentiment_category']]
df.head()
```

Output -

	tweet	date	id	sentiment	sentiment_category
0	Take your storytelling to the next level using...	2021-06-20	1406335989484822531	0.7089	positive
1	Choosing Fonts for Your Data Visualization | b...	2021-06-19	1406292636789526537	0.0000	neutral
2	This data visualization shows where our greate...	2021-06-19	1406082288035811330	0.0000	neutral
3	Looking for examples of stellar charts made so...	2021-06-18	1405948260796100610	0.4019	positive
4	With #WISQARS Data Visualization, you can disp...	2021-06-18	1405942146960613376	-0.4215	negative

Notice that the first few Tweets are the combination of positive, negative and neutral sentiment. For this analysis, we will only be using Tweets with positive and negative sentiment, since we want to visualize how stronger sentiments have changed over time.

Step 3: Visualization

Now that we have Tweets classified as positive and negative, let's take a look at changes in sentiment over time. We first need to group positive and negative sentiment and count them by date:

Command -

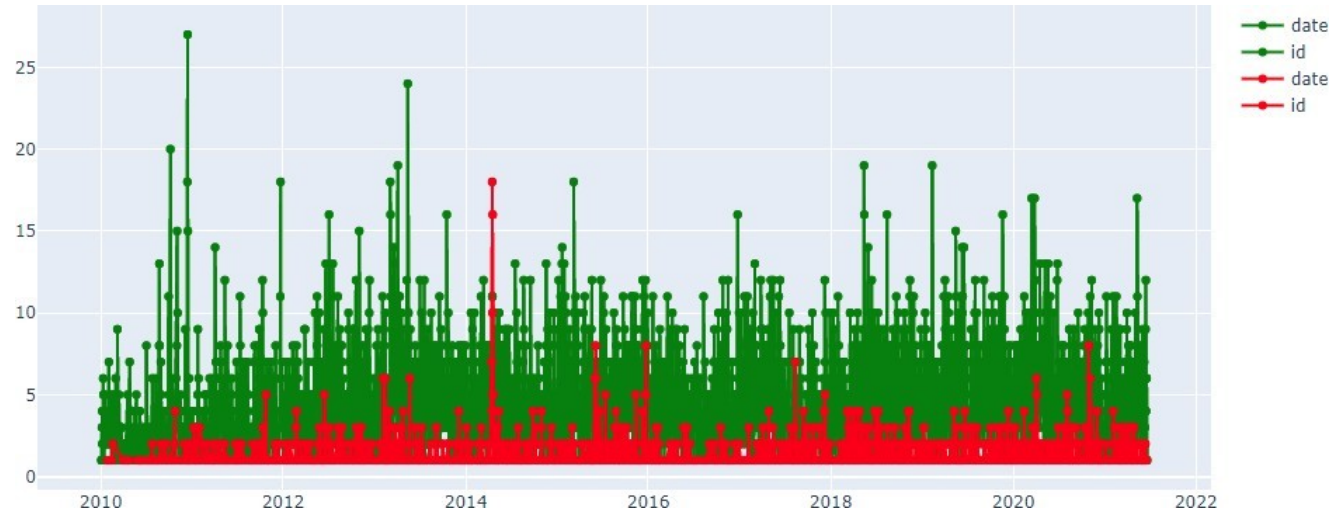
```
neg = df[df['sentiment_category']=='negative']
neg = neg.groupby(['date'],as_index=False).count()
pos = df[df['sentiment_category']=='positive']
pos = pos.groupby(['date'],as_index=False).count()
pos = pos[['date','id']]
neg = neg[['date','id']]
```

Now, we can visualize sentiment by date using Plotly, by running the following lines of code:

Command -

```
import plotly.graph_objs as go
fig = go.Figure()
for col in pos.columns:
    fig.add_trace(go.Scatter(x=pos['date'],  y=pos['id'],
                             name = col,
                             mode = 'markers+lines',
                             line=dict(shape='linear'),
                             connectgaps=True,
                             line_color='green'
                             )
                  )
for col in neg.columns:
    fig.add_trace(go.Scatter(x=neg['date'],  y=neg['id'],
                             name = col,
                             mode = 'markers+lines',
                             line=dict(shape='linear'),
                             connectgaps=True,
                             line_color='red'
                             )
                  )
fig.show()
```

Final Output - You should see a chart that looks like this:



The red line represents negative sentiment, and the green line represents positive sentiment.

Assignment Questions:

1. **What is Twitter sentiment analysis?**
2. **What is Natural Language Processing (NLP) and What are the stages in the life cycle of NLP?**
3. **What is NLTK? How to tokenize a sentence using the NLTK package?**
4. **Explain any two real-life applications of Natural Language Processing.**