Sentiment Analysis using Twitter API to predict election results based on tweets

Donald Trump vs. Joe Biden

This project includes the basics of social media mining by collecting Twitter data, pre-processing the data, and conducting exploratory analysis.

Task 1: Collecting Twitter Data

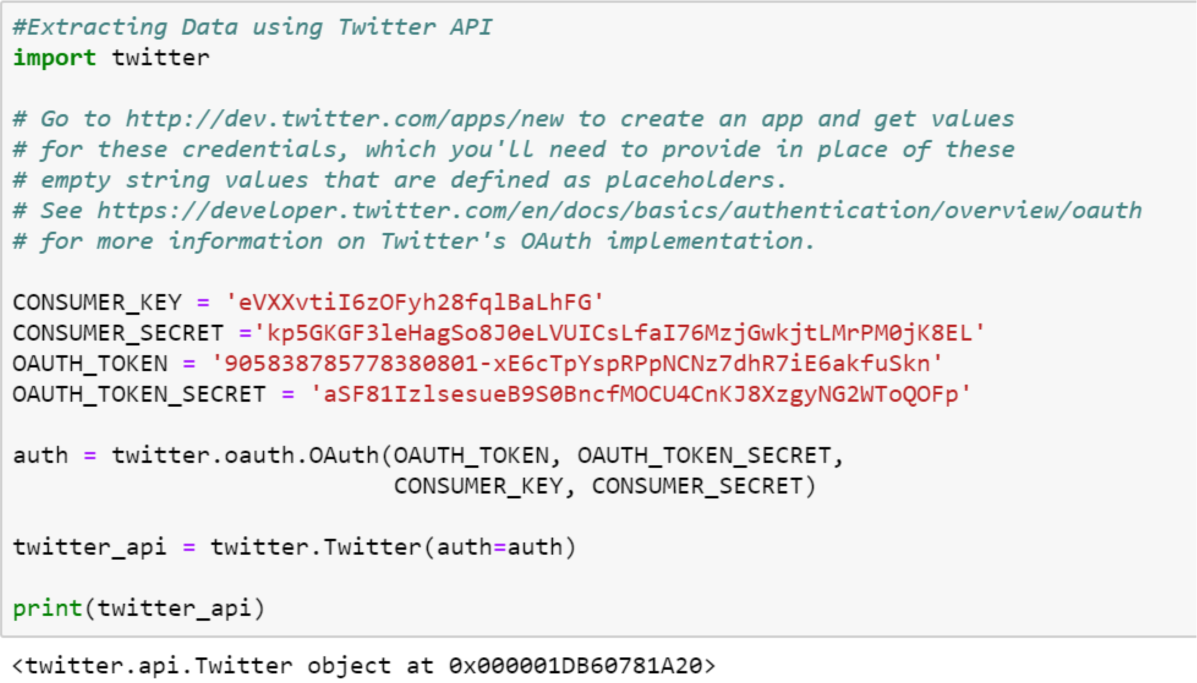
In order to collect tweets from Twitter API followed the below steps:

1) **Create a Twitter Developer Account**: Give application details/descriptions and click on create application.
2) **Key Generation**: Obtain Consumer API 'Keys and Access Tokens' used for authentication of twitter API.

Install the twitter package on the system using the command: **'pip install twitter'**

For fetching tweets using the twitter API we first need to complete authentication using the keys and access tokens obtained after account creation.

Twitter Authentication:



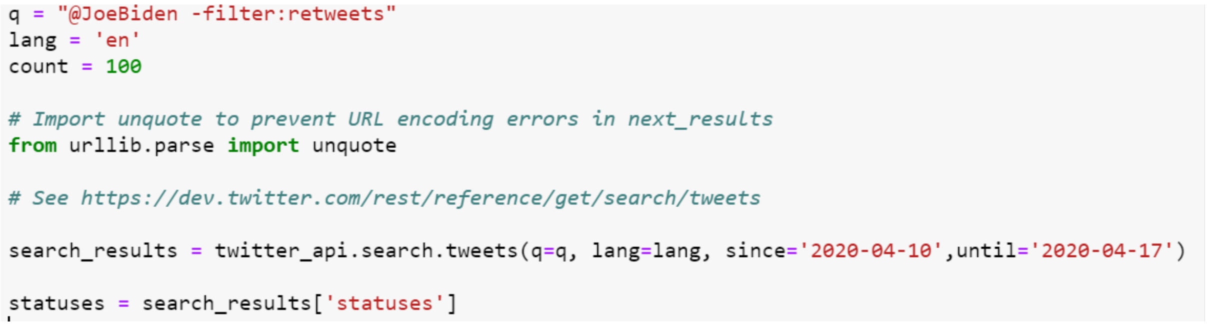
```
#Extracting Data using Twitter API
import twitter

# Go to http://dev.twitter.com/apps/new to create an app and get values
# for these credentials, which you'll need to provide in place of these
# empty string values that are defined as placeholders.
# See https://developer.twitter.com/en/docs/basics/authentication/overview/oauth
# for more information on Twitter's OAuth implementation.

CONSUMER_KEY = 'eVXXvtiI6zOFyh28fqlBaLhFG'
CONSUMER_SECRET ='kp5GKGF3leHagSo8J0eLVUICsLfaI76MzjGwkjtLMrPM0jK8EL'
OAUTH_TOKEN = '905838785778380801-xE6cTpYspRPpNCNz7dhR7iE6akfuSkn'
OAUTH_TOKEN_SECRET = 'aSF81IzlsesueB9S0BncfMOCU4CnKJ8XzgyNG2WToQOFp'

auth = twitter.oauth.OAuth(OAUTH_TOKEN, OAUTH_TOKEN_SECRET,
                           CONSUMER_KEY, CONSUMER_SECRET)

twitter_api = twitter.Twitter(auth=auth)

print(twitter_api)
```

```
<twitter.api.Twitter object at 0x000001DB60781A20>
```

Next, we need to find tweets related to Joe Biden and Donald trump using the Twitter API. Hence, we can mention their twitter user ids in our search query and save the tweets in a JSON file.

Tweets should be written in English:

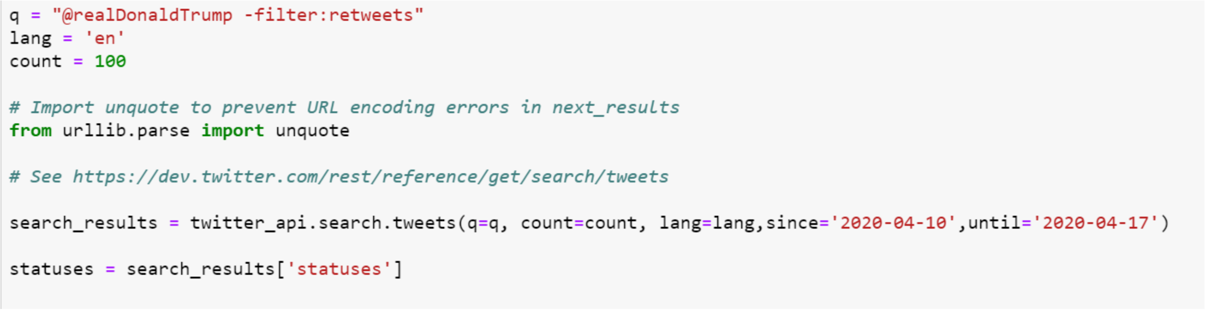
In order to get all the tweets in English we need to set the "**lang = en**" in the search query as shown below in the code snippet.



```python
q = "@JoeBiden -filter:retweets"
lang = 'en'
count = 100

# Import unquote to prevent URL encoding errors in next_results
from urllib.parse import unquote

# See https://dev.twitter.com/rest/reference/get/search/tweets

search_results = twitter_api.search.tweets(q=q, lang=lang, since='2020-04-10',until='2020-04-17')

statuses = search_results['statuses']
```

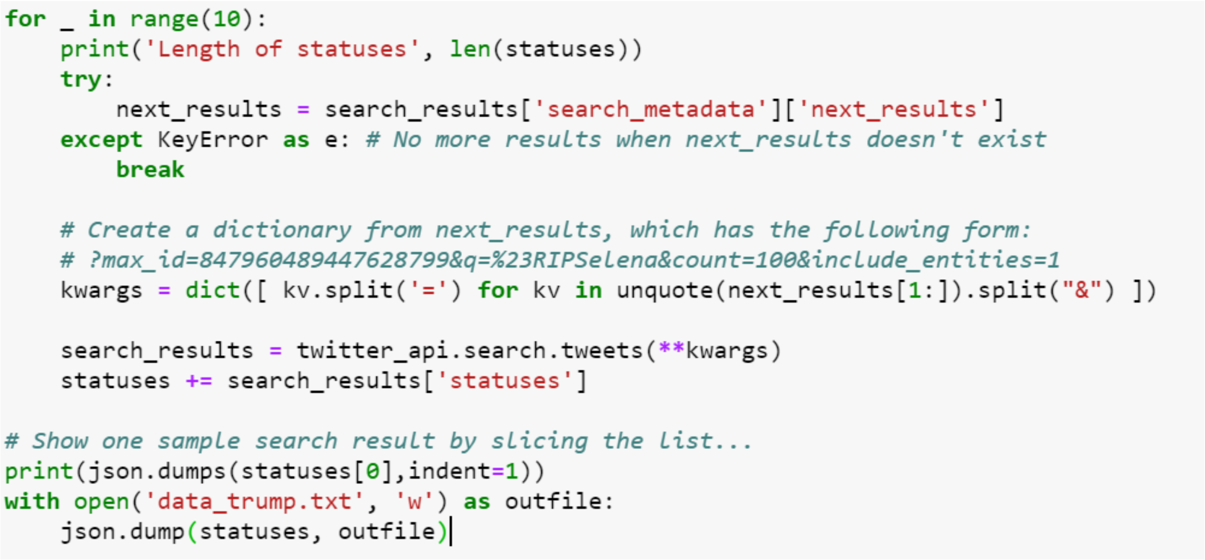
Remove/filter retweets to get more meaningful information if you can get at least 1,000 tweets.

In order to remove/filter retweets we need to set a filter in the search query **"-filter: retweets"**



```python
q = "@realDonaldTrump -filter:retweets"
lang = 'en'
count = 100

# Import unquote to prevent URL encoding errors in next_results
from urllib.parse import unquote

# See https://dev.twitter.com/rest/reference/get/search/tweets

search_results = twitter_api.search.tweets(q=q, count=count, lang=lang,since='2020-04-10',until='2020-04-17')

statuses = search_results['statuses']
```

To get at least 1000 tweets, we need to access the '**next_results'** field from the '**search_metadata**'.The API returns maximum of **100 tweets per page** and to access the remaining tweets we need to use the next_results field.



```python
for _ in range(10):
    print('Length of statuses', len(statuses))
    try:
        next_results = search_results['search_metadata']['next_results']
    except KeyError as e: # No more results when next_results doesn't exist
        break

    # Create a dictionary from next_results, which has the following form:
    # ?max_id=847960489447628799&q=%23RIPSelena&count=100&include_entities=1
    kwargs = dict([ kv.split('=') for kv in unquote(next_results[1:]).split("&") ])

    search_results = twitter_api.search.tweets(**kwargs)
    statuses += search_results['statuses']

# Show one sample search result by slicing the list...
print(json.dumps(statuses[0],indent=1))
with open('data_trump.txt', 'w') as outfile:
    json.dump(statuses, outfile)
```

Sample tweets for Joe Biden in JSON:

{
 "created_at": "Thu Apr 16 23:59:59 +0000 2020",
 "id": 1250937009528549378,
 "id_str": "1250937009528549378",

```
 "text": "@KamalaHarris Or mail in ? I sure hope that @JoeBiden endorses you for VP @KamalaHarris. I would be the happiest pe\u2026 https://t.co/j4LV8T5ZcF",
 "truncated": true,
 "entities": {
  "hashtags": [],
  "symbols": [],
  "user_mentions": [
   {
    "screen_name": "KamalaHarris",
    "name": "Kamala Harris",
    "id": 30354991,
    "id_str": "30354991",
    "indices": [
     0,
     13
    ]
   },
   {
    "screen_name": "JoeBiden",
    "name": "Joe Biden",
    "id": 939091,
    "id_str": "939091",
    "indices": [
     44,
     53
    ]
   },
   {
    "screen_name": "KamalaHarris",
    "name": "Kamala Harris",
    "id": 30354991,
    "id_str": "30354991",
    "indices": [
     74,
     87
    ]
   }
  ],
  "urls": [
   {
    "url": "https://t.co/j4LV8T5ZcF",
    "expanded_url": "https://twitter.com/i/web/status/1250937009528549378",
    "display_url": "twitter.com/i/web/status/1\u2026",
    "indices": [
     117,
     140
    ]
   }
  ]
 },
 "metadata": {
  "iso_language_code": "en",
  "result_type": "recent"
 },
 "source": "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for Android</a>",
 "in_reply_to_status_id": 1250935507242176512,
 "in_reply_to_status_id_str": "1250935507242176512",
 "in_reply_to_user_id": 30354991,
 "in_reply_to_user_id_str": "30354991",
 "in_reply_to_screen_name": "KamalaHarris",
 "user": {
  "id": 1249054457427566593,
  "id_str": "1249054457427566593",
  "name": "Lori",
  "screen_name": "Lori90920955",
  "location": "",
  "description": "live your  life, love everyone. Love my fur babies. grammy of 5 ! We got to take back our Country. Pray that WE VOTE HIM OUT #Bluewave2020",
  "url": null,
  "entities": {
   "description": {
    "urls": []
   }
  },
  "protected": false,
  "followers_count": 5,
  "friends_count": 244,
  "listed_count": 0,
  "created_at": "Sat Apr 11 19:19:46 +0000 2020",
  "favourites_count": 205,
  "utc_offset": null,
  "time_zone": null,
  "geo_enabled": false,
  "verified": false,
  "statuses_count": 210,
  "lang": null,
  "contributors_enabled": false,
  "is_translator": false,
  "is_translation_enabled": false,
  "profile_background_color": "F5F8FA",
  "profile_background_image_url": null,
  "profile_background_image_url_https": null,
  "profile_background_tile": false,
  "profile_image_url": "http://pbs.twimg.com/profile_images/1249054747212025856/MRMyohig_normal.jpg",
  "profile_image_url_https": "https://pbs.twimg.com/profile_images/1249054747212025856/MRMyohig_normal.jpg",
  "profile_banner_url": "https://pbs.twimg.com/profile_banners/1249054457427566593/1586640713",
  "profile_link_color": "1DA1F2",
  "profile_sidebar_border_color": "C0DEED",
  "profile_sidebar_fill_color": "DDEEF6",
  "profile_text_color": "333333",
  "profile_use_background_image": true,
  "has_extended_profile": true,
  "default_profile": true,
  "default_profile_image": false,
  "following": false,
  "follow_request_sent": false,
  "notifications": false,
  "translator_type": "none"
 },
 "geo": null,
 "coordinates": null,
 "place": null,
 "contributors": null,
 "is_quote_status": false,
 "retweet_count": 2,
 "favorite_count": 2,
 "favorited": false,
 "retweeted": false,
 "lang": "en"
}
```

Sample tweets for Donald Trump in JSON:

```
{
 "created_at": "Thu Apr 16 23:59:59 +0000 2020",
 "id": 1250937012816801792,
 "id_str": "1250937012816801792",
 "text": "@Bunny_Slick @carl_cnp @JayMercer20 @itsJeffTiedrich @realDonaldTrump Jan 
31 Trump Administration declared the coro\u2026 https://t.co/6YwtfetL0n",
 "truncated": true,
 "entities": {
  "hashtags": [],
  "symbols": [],
  "user_mentions": [
   {
    "screen_name": "Bunny_Slick",
    "name": "Bunny Slick",
    "id": 793949726483693569,
    "id_str": "793949726483693569",
    "indices": [
     0,
     12
    ]
   },
   {
    "screen_name": "carl_cnp",
  "name": "Carl Purseglove",
  "id": 147672777,
  "id_str": "147672777",
  "indices": [
   13,
   22
  ]
 },
 {
  "screen_name": "JayMercer20",
  "name": "Jay Mercer",
  "id": 1219007709757886464,
  "id_str": "1219007709757886464",
  "indices": [
   23,
   35
  ]
 },
 {
  "screen_name": "itsJeffTiedrich",
  "name": "Jeff Tiedrich",
  "id": 1009577803304656896,
  "id_str": "1009577803304656896",
  "indices": [
   36,
   52
  ]
 },
 {
  "screen_name": "realDonaldTrump",
  "name": "Donald J. Trump",
  "id": 25073877,
  "id_str": "25073877",
  "indices": [
   53,
   69
  ]
 }
],
"urls": [
 {
  "url": "https://t.co/6YwtfetL0n",
  "expanded_url": "https://twitter.com/i/web/status/1250937012816801792",
  "display_url": "twitter.com/i/web/status/1\u2026",
  "indices": [
   117,
   140
  ]
 }
]
```

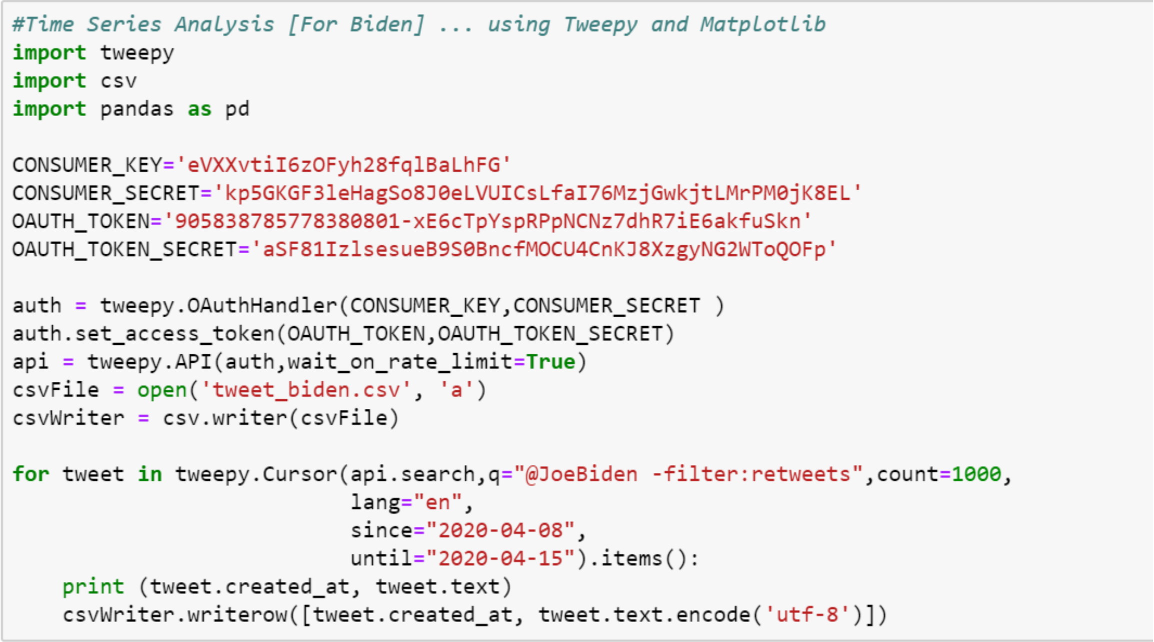
```
 },
 "metadata": {
  "iso_language_code": "en",
  "result_type": "recent"
 },
 "source": "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for An
droid</a>",
 "in_reply_to_status_id": 1250935646396362752,
 "in_reply_to_status_id_str": "1250935646396362752",
 "in_reply_to_user_id": 1096464427,
 "in_reply_to_user_id_str": "1096464427",
 "in_reply_to_screen_name": "Hammeredge1",
 "user": {
  "id": 1096464427,
  "id_str": "1096464427",
  "name": "Hammeredge",
  "screen_name": "Hammeredge1",
  "location": "Earth #Qanon #WeAreTheNewsNow",
  "description": "Jer 23:29 Is not my word like as a fire? saith the LORD; and like a hammer t
hat breaketh the rock in pieces? #Qanon #WeAreTheNewsNow #FactsMatter #WWG1WGA
",
  "url": null,
  "entities": {
   "description": {
    "urls": []
   }
  },
  "protected": false,
  "followers_count": 1419,
  "friends_count": 1431,
  "listed_count": 2,
  "created_at": "Wed Jan 16 21:52:18 +0000 2013",
  "favourites_count": 14339,
  "utc_offset": null,
  "time_zone": null,
  "geo_enabled": false,
  "verified": false,
  "statuses_count": 19239,
  "lang": null,
  "contributors_enabled": false,
  "is_translator": false,
  "is_translation_enabled": false,
  "profile_background_color": "000000",
  "profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png",
  "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.p
ng",
  "profile_background_tile": false,
  "profile_image_url": "http://pbs.twimg.com/profile_images/1248117452598161408/zDv0H
R3l_normal.jpg",
  "profile_image_url_https": "https://pbs.twimg.com/profile_images/1248117452598161408/zDv0HR3l_normal.jpg",
  "profile_banner_url": "https://pbs.twimg.com/profile_banners/1096464427/1551010712",
  "profile_link_color": "1B95E0",
  "profile_sidebar_border_color": "000000",
  "profile_sidebar_fill_color": "000000",
  "profile_text_color": "000000",
  "profile_use_background_image": false,
  "has_extended_profile": true,
  "default_profile": false,
  "default_profile_image": false,
  "following": false,
  "follow_request_sent": false,
  "notifications": false,
  "translator_type": "none"
 },
 "geo": null,
 "coordinates": null,
 "place": null,
 "contributors": null,
 "is_quote_status": false,
 "retweet_count": 0,
 "favorite_count": 0,
 "favorited": false,
 "retweeted": false,
 "lang": "en"
}
```

Task 2: Exploratory Analysis

- **A time series figure with the number of tweets per day over time for both candidates**

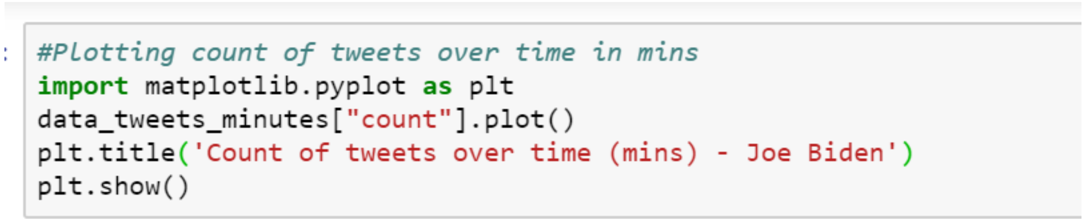
  NOTE: Collecting tweets for each day was taking a lot of time. Hence, I have used tweepy to collect the tweets for 14 April, 2020 from 4:00 pm to 23:59 pm for Joe Biden and 14 April, 2020 from 23:00 pm to 23:59 pm for Donald Trump.

  Tweet Collection:

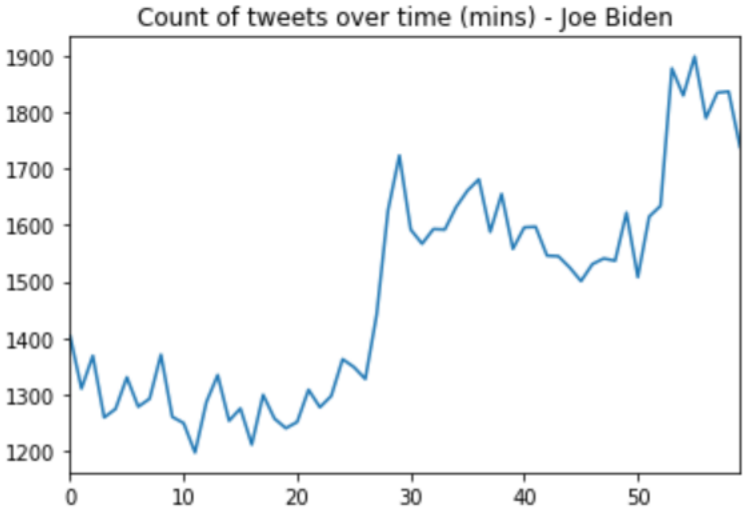


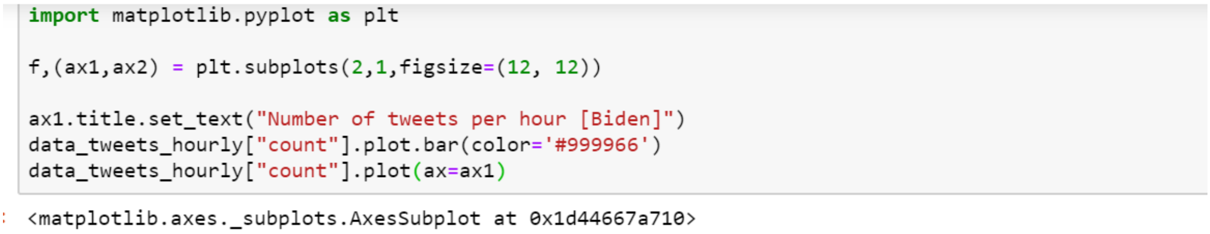
```
#Time Series Analysis [For Biden] ... using Tweepy and Matplotlib
import tweepy
import csv
import pandas as pd

CONSUMER_KEY='eVXXvtiI6zOFyh28fqlBaLhFG'
CONSUMER_SECRET='kp5GKGF3leHagSo8J0eLVUICsLfaI76MzjGwkjtLMrPM0jK8EL'
OAUTH_TOKEN='905838785778380801-xE6cTpYspRPpNCNz7dhR7iE6akfuSkn'
OAUTH_TOKEN_SECRET='aSF81IzlsesueB9S0BncfMOCU4CnKJ8XzgyNG2WToQOFp'

auth = tweepy.OAuthHandler(CONSUMER_KEY,CONSUMER_SECRET )
auth.set_access_token(OAUTH_TOKEN,OAUTH_TOKEN_SECRET)
api = tweepy.API(auth,wait_on_rate_limit=True)
csvFile = open('tweet_biden.csv', 'a')
csvWriter = csv.writer(csvFile)

for tweet in tweepy.Cursor(api.search,q="@JoeBiden -filter:retweets",count=1000,
                           lang="en",
                           since="2020-04-08",
                           until="2020-04-15").items():
    print (tweet.created_at, tweet.text)
    csvWriter.writerow([tweet.created_at, tweet.text.encode('utf-8')])
```

Time Series [Joe Biden]:



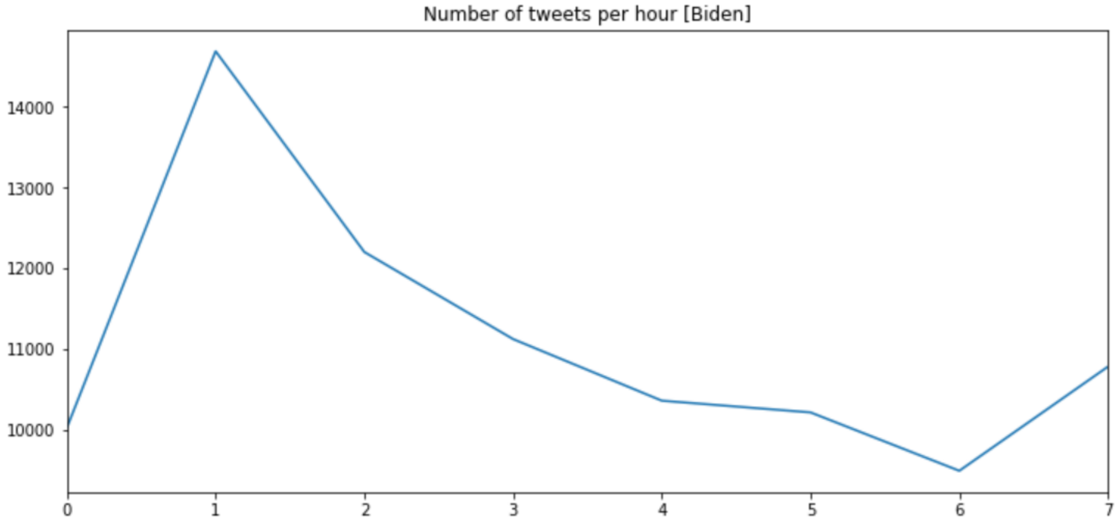
```
#Plotting count of tweets over time in mins
import matplotlib.pyplot as plt
data_tweets_minutes["count"].plot()
plt.title('Count of tweets over time (mins) - Joe Biden')
plt.show()
```



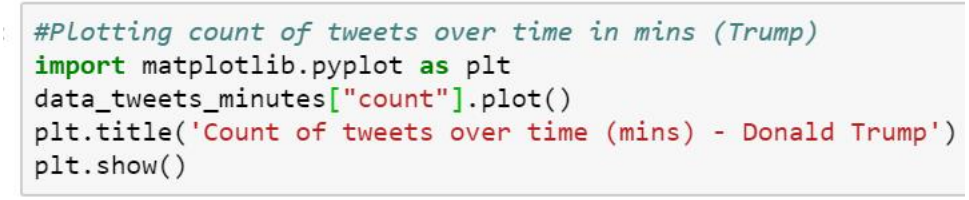


```
import matplotlib.pyplot as plt

f,(ax1,ax2) = plt.subplots(2,1,figsize=(12, 12))

ax1.title.set_text("Number of tweets per hour [Biden]")
data_tweets_hourly["count"].plot.bar(color='#999966')
data_tweets_hourly["count"].plot(ax=ax1)
```

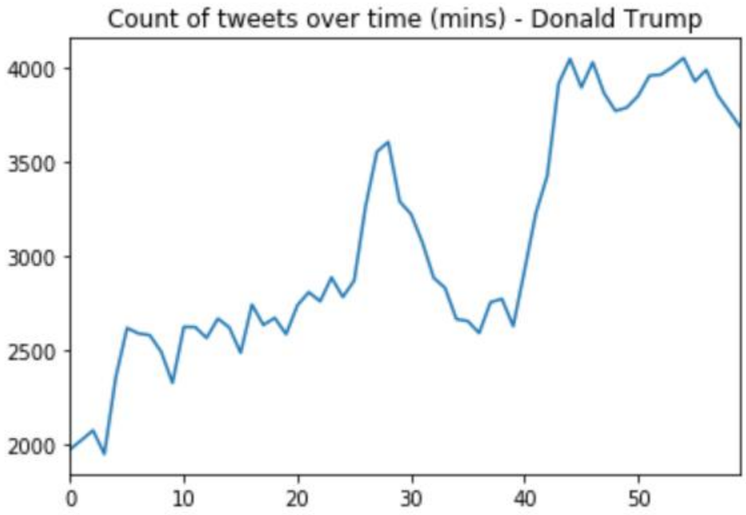
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d44667a710>
```

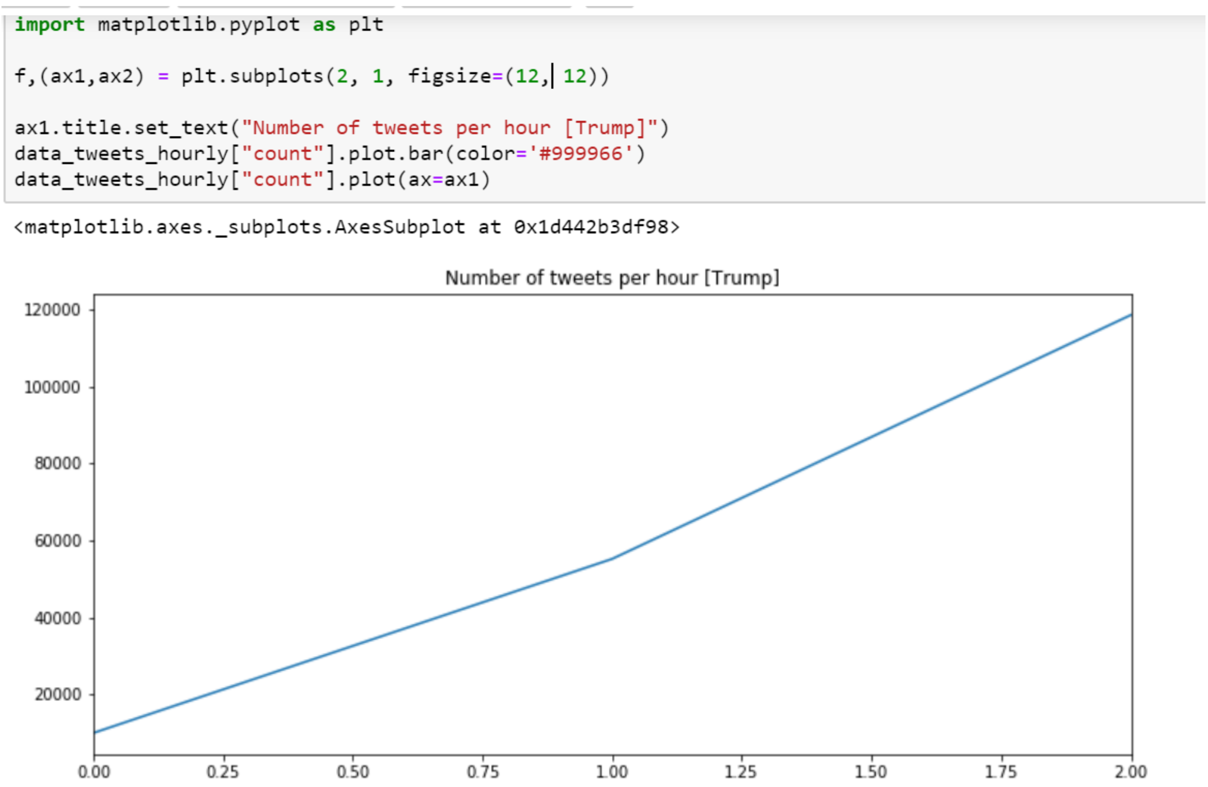


Time Series [Donald Trump]



```
#Plotting count of tweets over time in mins (Trump)
import matplotlib.pyplot as plt
data_tweets_minutes["count"].plot()
plt.title('Count of tweets over time (mins) - Donald Trump')
plt.show()
```





```
import matplotlib.pyplot as plt

f,(ax1,ax2) = plt.subplots(2, 1, figsize=(12, 12))

ax1.title.set_text("Number of tweets per hour [Trump]")
data_tweets_hourly["count"].plot.bar(color='#999966')
data_tweets_hourly["count"].plot(ax=ax1)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d442b3df98>
```

Perform two more interesting analysis of your choice (e.g., sentiment analysis, clustering and so on).

1] Sentiment Analysis using VADER:

I have used python’s **VADER** package for doing sentiment analysis. VADER (**Valence Aware Dictionary and sentiment Reasoner**) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media.

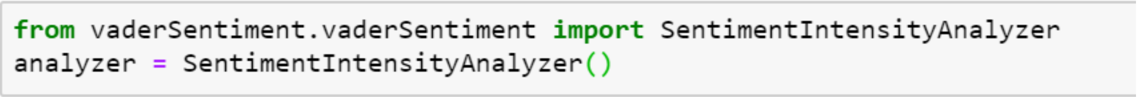
VADER provides us with how negative or positive a tweet is and it doesn’t require any training data but is constructed from a generalizable, valence-based, human-curated gold standard sentiment lexicon.

- We first need to install VADER package on the system:
  pip install vaderSentiment
- It uses polarity_scores() to get the polarity score of each sentence.
- This function returns four scores based on the input sentence i.e Positive Score, Negative Score, Compound Score and Neutral Score.
- The compound score is the normalized weighted composite score and is the single most useful metric to determine the sentiment of the sentence.

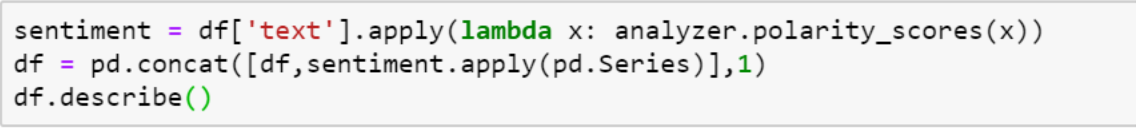
Below is the code snippet of VADER sentiment analyser.

VADER for Biden:

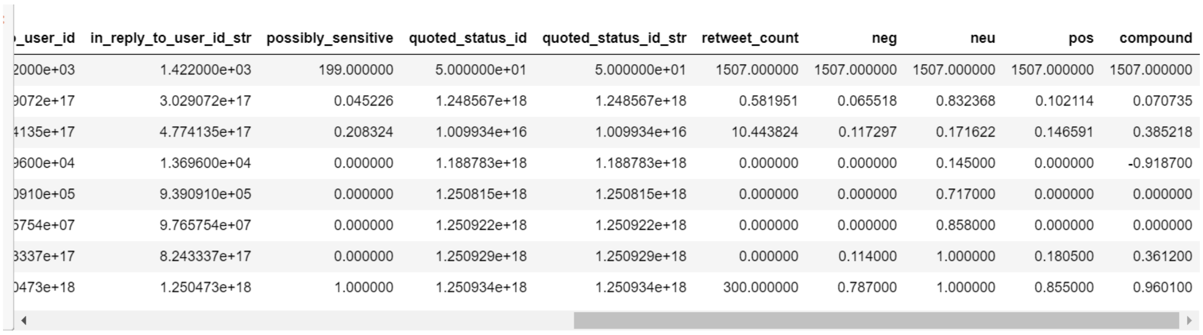
Compound Score: 0.8957



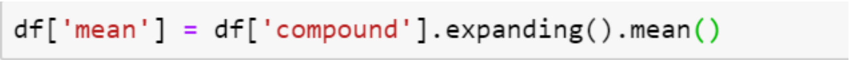
```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyzer = SentimentIntensityAnalyzer()
```



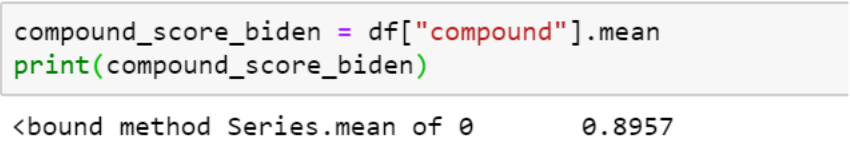
```
sentiment = df['text'].apply(lambda x: analyzer.polarity_scores(x))
df = pd.concat([df,sentiment.apply(pd.Series)],1)
df.describe()
```



_user_id	in_reply_to_user_id_str	possibly_sensitive	quoted_status_id	quoted_status_id_str	retweet_count	neg	neu	pos	compound
2000e+03	1.422000e+03	199.000000	5.000000e+01	5.000000e+01	1507.000000	1507.000000	1507.000000	1507.000000	1507.000000
9072e+17	3.029072e+17	0.045226	1.248567e+18	1.248567e+18	0.581951	0.065518	0.832368	0.102114	0.070735
4135e+17	4.774135e+17	0.208324	1.009934e+16	1.009934e+16	10.443824	0.117297	0.171622	0.146591	0.385218
9600e+04	1.369600e+04	0.000000	1.188783e+18	1.188783e+18	0.000000	0.000000	0.145000	0.000000	-0.918700
0910e+05	9.390910e+05	0.000000	1.250815e+18	1.250815e+18	0.000000	0.000000	0.717000	0.000000	0.000000
5754e+07	9.765754e+07	0.000000	1.250922e+18	1.250922e+18	0.000000	0.000000	0.858000	0.000000	0.000000
3337e+17	8.243337e+17	0.000000	1.250929e+18	1.250929e+18	0.000000	0.114000	1.000000	0.180500	0.361200
0473e+18	1.250473e+18	1.000000	1.250934e+18	1.250934e+18	300.000000	0.787000	1.000000	0.855000	0.960100

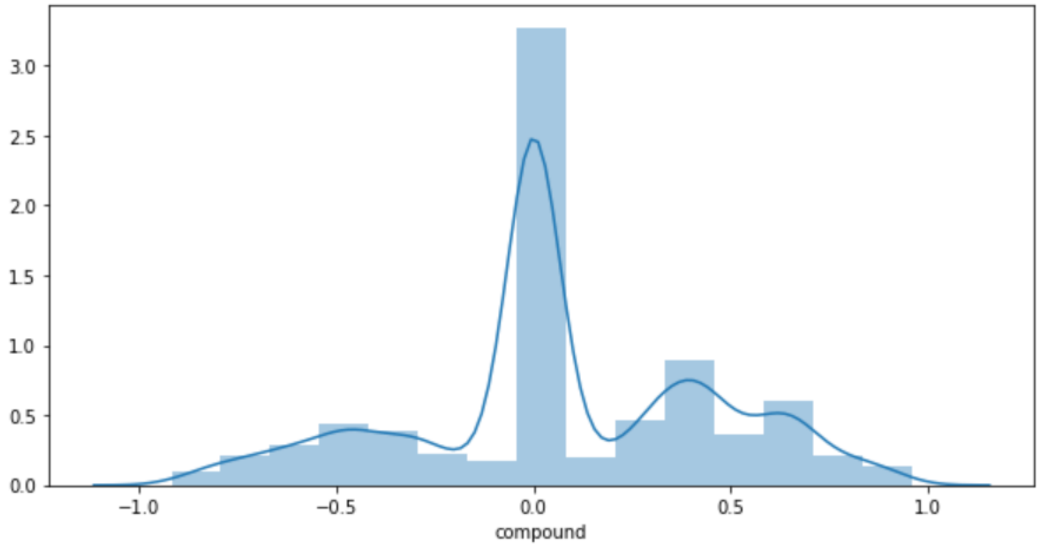


```
df['mean'] = df['compound'].expanding().mean()
```



```
compound_score_biden = df["compound"].mean
print(compound_score_biden)
```

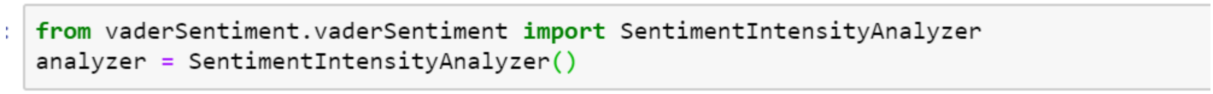
```
<bound method Series.mean of 0        0.8957
```



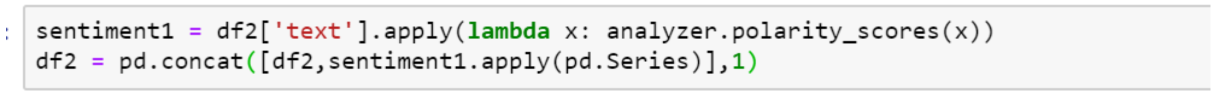
The mean compound score for Biden is 0.8957 which means that the overall sentiment of the public towards Biden is positive.

VADER for Trump:

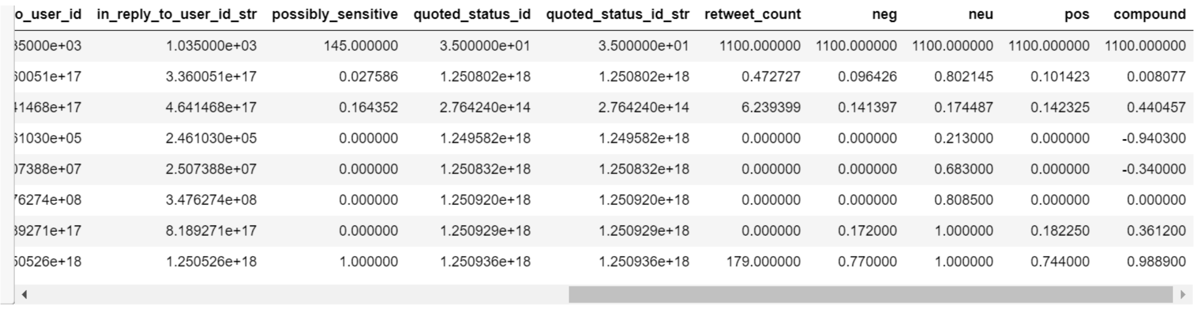
Compound Score: 0.00



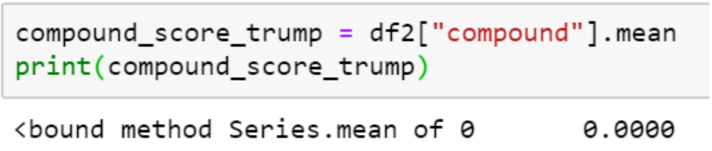
```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyzer = SentimentIntensityAnalyzer()
```



```
sentiment1 = df2['text'].apply(lambda x: analyzer.polarity_scores(x))
df2 = pd.concat([df2,sentiment1.apply(pd.Series)],1)
```

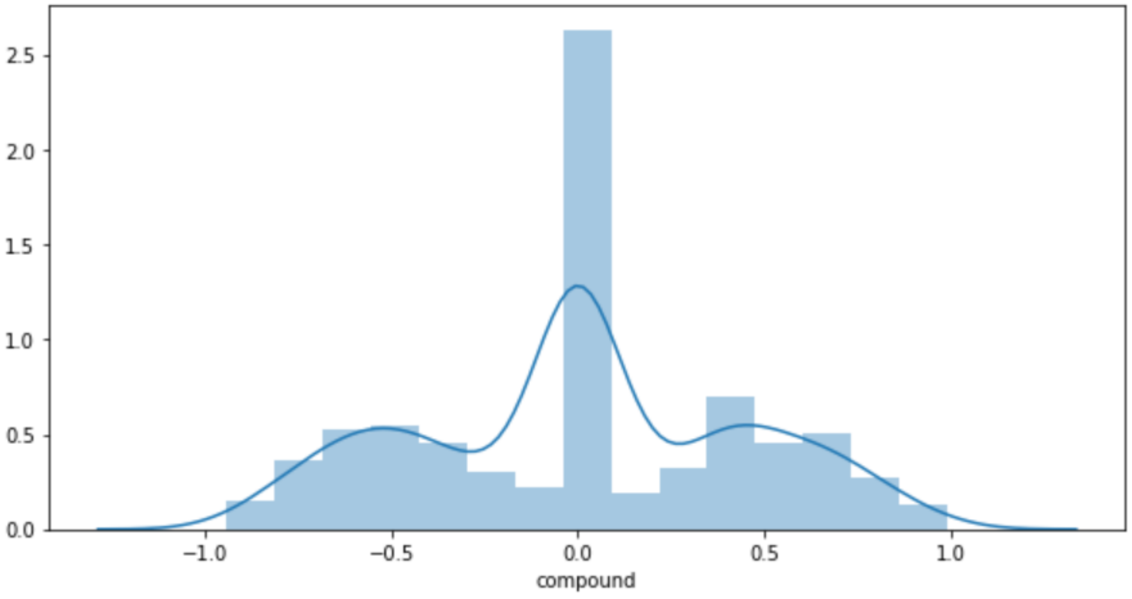


o_user_id	in_reply_to_user_id_str	possibly_sensitive	quoted_status_id	quoted_status_id_str	retweet_count	neg	neu	pos	compound
35000e+03	1.035000e+03	145.000000	3.500000e+01	3.500000e+01	1100.000000	1100.000000	1100.000000	1100.000000	1100.000000
60051e+17	3.360051e+17	0.027586	1.250802e+18	1.250802e+18	0.472727	0.096426	0.802145	0.101423	0.008077
41468e+17	4.641468e+17	0.164352	2.764240e+14	2.764240e+14	6.239399	0.141397	0.174487	0.142325	0.440457
61030e+05	2.461030e+05	0.000000	1.249582e+18	1.249582e+18	0.000000	0.000000	0.213000	0.000000	-0.940300
07388e+07	2.507388e+07	0.000000	1.250832e+18	1.250832e+18	0.000000	0.000000	0.683000	0.000000	-0.340000
76274e+08	3.476274e+08	0.000000	1.250920e+18	1.250920e+18	0.000000	0.000000	0.808500	0.000000	0.000000
89271e+17	8.189271e+17	0.000000	1.250929e+18	1.250929e+18	0.000000	0.172000	1.000000	0.182250	0.361200
50526e+18	1.250526e+18	1.000000	1.250936e+18	1.250936e+18	179.000000	0.770000	1.000000	0.744000	0.988900



```
compound_score_trump = df2["compound"].mean
print(compound_score_trump)
```

```
<bound method Series.mean of 0         0.0000
```



The mean compound score for Trump is 0.00 which means that the overall sentiment of the public towards Trump is neutral.

Based on the compound scores of VADER analysis, Joe Biden would win the 2020 Elections as he has a more positive compound score as compared to Trump.

2]Sentiment Analysis using Textblob (NLTK):

Textblob is the python library for processing textual data.

Install it using following command:
pip install textblob

Also, we need to install some NLTK corpora using following command:

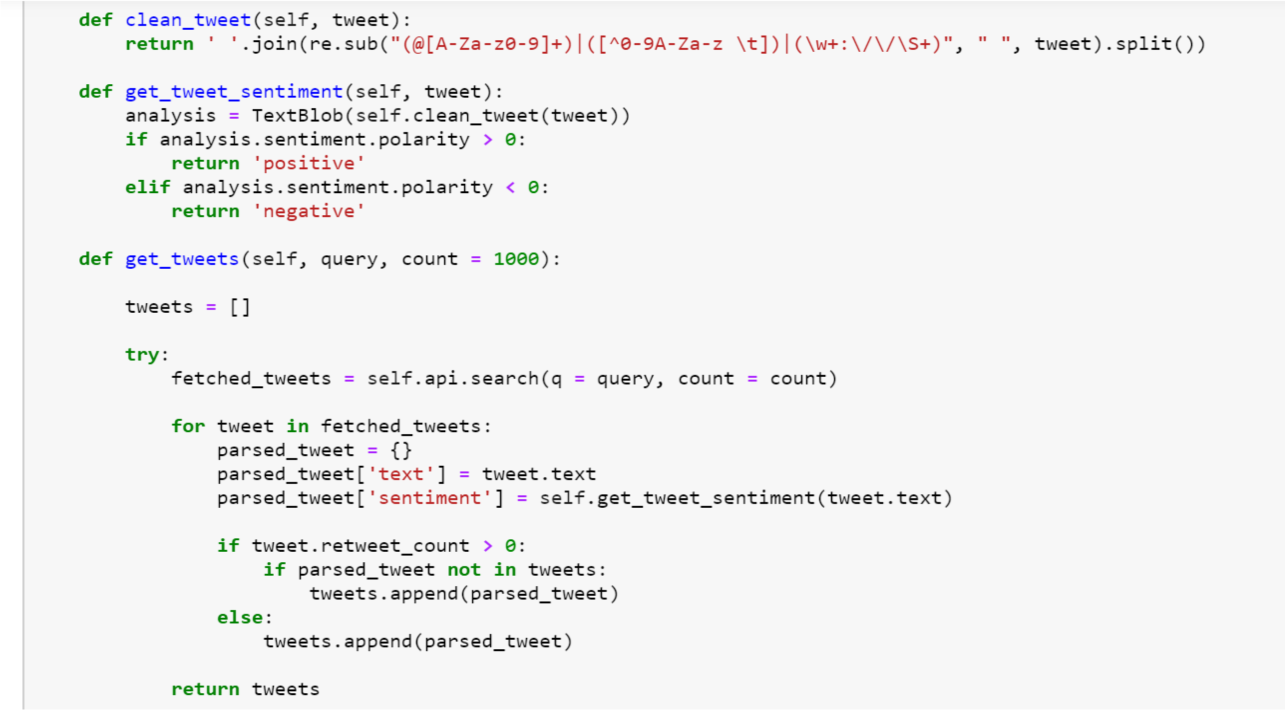
python -m textblob.download_corpora

The code includes 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.

First, the **clean_tweet** method is used to remove links, special characters, etc. from the tweet using some simple regex. Then, as we pass **tweet** to create a **TextBlob** object, following processing is done over text by textblob library:

- Tokenize the tweet, i.e. split words from body of text.
- Remove stop words from the tokens.
- Do POS (part of speech) tagging of the tokens and select only significant features/tokens like adjectives, adverbs, etc.
- Pass the tokens to a **sentiment classifier** which classifies the tweet sentiment as positive, negative or neutral by assigning it a polarity between -1.0 to 1.0.
- Positive and negative features are extracted from each positive and negative review respectively.
- Training data now consists of labelled positive and negative features. This data is trained on a Naive Bayes Classifier.
- Then, use **sentiment.polarity** method of **TextBlob** class to get the polarity of tweet between -1 to 1.
- Finally, parsed tweets are returned. Then, we can do various type of statistical analysis on the tweets. For example, in the code below I tried to find the percentage of positive, negative and neutral tweets about a query.



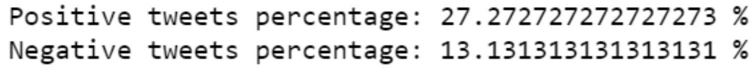
```
def clean_tweet(self, tweet):
    return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", tweet).split())

def get_tweet_sentiment(self, tweet):
    analysis = TextBlob(self.clean_tweet(tweet))
    if analysis.sentiment.polarity > 0:
        return 'positive'
    elif analysis.sentiment.polarity < 0:
        return 'negative'

def get_tweets(self, query, count = 1000):

    tweets = []

    try:
        fetched_tweets = self.api.search(q = query, count = count)

        for tweet in fetched_tweets:
            parsed_tweet = {}
            parsed_tweet['text'] = tweet.text
            parsed_tweet['sentiment'] = self.get_tweet_sentiment(tweet.text)

            if tweet.retweet_count > 0:
                if parsed_tweet not in tweets:
                    tweets.append(parsed_tweet)
            else:
                tweets.append(parsed_tweet)

        return tweets
```

Analysis for Biden:

Positive tweets percentage: 29.0 %
Negative tweets percentage: 17.0 %

Based on the above scores, it can be said that Biden has a more positive impact on the audience.

Analysis for Trump:



```
Positive tweets percentage: 27.272727272727273 %
Negative tweets percentage: 13.131313131313131 %
```

Based on the above scores, it can be said that Trump has a more positive impact on the audience. However, a lesser positive impact than Biden. Hence, it can be concluded that Joe Biden has higher chances of winning the 2020 elections as per the above analysis.