

**A PROJECT REPORT ON**  
**SENTIMENT ANALYSIS ON POLITICAL TWEETS BY**  
**PRESIDENTIAL CANDIDATES ON US ELECTION**

SUBMITTED TO  
THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE  
OF  
**THIRD YEAR OF ENGINEERING (COMPUTER ENGINEERING)**

**SUBMITTED BY**

KRUNAL PATIL	Roll No : 3327
SHAIK AFTAB	Roll No : 3345
SURAJ KANYAL	Roll No : 3351
SWAPNIL SINGH	Roll No : 3352



**DEPARTMENT OF COMPUTER ENGINEERING**

**ARMY INSTITUTE OF TECHNOLOGY**  
**DIGHI HILLS, PUNE 411 015**

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**2020 -2021**



## **CERTIFICATE**

This is to certify that the project report entitled

### **“SENTIMENT ANALYSIS ON POLITICAL TWEETS BY PRESIDENTIAL CANDIDATES ON US ELECTION”**

#### **SUBMITTED BY**

KRUNAL PATIL	Roll No : 3327
SHAIK AFTAB	Roll No : 3345
SURAJ KANYAL	Roll No : 3351
SWAPNIL SINGH	Roll No : 3352

bonafide students of this institute and the work has been carried out under the supervision of **Prof. P. R. SONAWANE** and it is approved for the partial fulfillment of the requirement of SKILL DEVELOPMENT LABORATORY, TE (Computer Engineering) of Savitribai Phule Pune University.

**(Prof. P R Sonawane)**  
Guide  
Department of Computer  
Engineering

**(Prof.(Dr) S.R.Dhore)**  
Head,  
Department of Computer  
Engineering

**(Dr. B.P.Patil)**  
Principal,  
Army Institute of Technology

Place : Pune  
Date : 08 May 2021

## **ACKNOWLEDGEMENT**

*For the project entitled “Sentiment Analysis on Political Tweets by Presidential Candidates on US Election”, we required a lot of guidance from many friends and seniors and we are extremely thankful to all of them for the completion of our project work. The work we have done is only because of such assistance and guidance.*

*We respectfully thank our Teacher, **Prof. Prahlad R Sonawane** sir, for his guidance and giving us the opportunity to do the project work in Skill Development Laboratory and giving us all the support and which made us complete the task on time. We are extremely grateful to him for providing such a nice and comfortable environment even though he has to do this during his online sessions.*

*We are also grateful to **Prof. Dr. S. R. Dhore**, Head of Computer Engineering Department, Army institute of Technology for his indispensable support, suggestions.*

*I also want to thank all my batchmates for providing great support and guidance till the completion of our project. Lastly, our family members also give us enough support and strength till the end of the whole project. Without them our project cannot be completed on time.*

*Krunal Patil (3327)*

*Shaik Aftab (3345)*

*Suraj Singh Kanyal (3351)*

*Swapnil Singh (3352)*

## **ABSTRACT**

Sentiment analysis is an evaluation of the opinion of the speaker, writer or other subject with regard to some topic. Be it articles, facebook groups, books, songs, videos, etc. Too many opinions have made it difficult in finding the content of one's liking.

So, to solve this problem, we used concepts of machine learning using python and some other important libraries which will help us in achieving the required goal. Our objective is to create a sentiment analysis based on the tweets of some of the participating leaders and test our algorithms to test which one has a higher chance of winning.

A basic sentiment analysis works as one of the following factors for downloading and cleaning the tweets and arranging them in a csv file. Some of the libraries are in-built and some are customized. Accuracy is the key for selecting the algorithm and lastly data visualization which give the perfect explanation for the result.

## **TABLE OF CONTENTS**

<b>CHAPTER</b>	<b>TITLE</b>	<b>PAGE NO</b>
01	Introduction	06
02	Literature Survey	07
03	Project Design	09
04	Implementation	10
05	Coding Related	17
06	Deployment	18
07	Testing	20
08	Conclusion	22
09	Future Scope	
10	References	

# **1. INTRODUCTION**

## **1.1. OVERVIEW**

Sentiment Analysis is a Natural Language Processing task that aims to obtain anyone's feelings expressed in positive or negative comments, questions and requests, by analysing a large series of documents. Generally, sentiment analysis aims to determine the attitude of a speaker or a person with respect to some topic or the overall tonality of a document.

In past years, the exponential increase in the Internet usage and exchange of public opinion is the driving force behind Sentiment Analysis today. The World Web is a huge repository of structured and unstructured data. The analysis of this data to extract latent public opinion and sentiment is a challenging task.

The analysis of sentiments may be document based where the sentiment in the entire document is summarized as negative, positive or objective. Sentiment Analysis can be phrase based where the phrases in a sentence are classified according to polarity and subjectivity. The person may speak about some objective facts or subjective opinions. It is necessary to distinguish between the two.

A text may contain many fields, but it is necessary to find the direction towards which the sentiment is more. It identifies the polarity and subjectivity of the sentiment. They are classified as objective (facts), positive (a state of happiness, satisfaction writer from writer's end) or negative (a state of sorrow, disappointment from writer's end). The sentiments can further be given a score based on their degree of positivity, negativity or objectivity. Here we will compare the Sentiments of both Biden's and Trump's and based on them we will conduct the result.

## **1.2. Motivation**

Motivation for Sentiment Analysis is two-fold. Both seekers and givers highly value opinions.

### **1.2.1. The Seeker's Perspective**

While taking a decision it is very important for us to know the opinion of the people around us. With the advent of the Internet we see people expressing their opinions on social media platforms. These are now actively read by people who seek an opinion about a particular entity. From a seeker's point of view extracting opinions about a particular entity is very important. Trying to go through such a vast amount of information to understand the general opinion is impossible for users just by the sheer volume of this data. Hence, the need of a system that differentiates between good reviews and bad reviews. Further, labeling these documents with their sentiment would provide a succinct summary to the readers about the general opinion regarding an entity.

### **1.2.2. The Giver's Perspective**

With the explosion of Web 2.0 platforms such as blogs, discussion forums, etc., consumers have at their disposal, a platform to share their brand experiences and opinions, positive or negative regarding any product or service. Since the givers have started using the power of the Internet to expand their horizons, there has been a surge of review sites and blogs, where users can perceive an entity's advantages and faults. These opinions thus shape the future of the entity. The givers need a system that can identify trends in seeker's reviews and use them to improve their entity and also identify the requirements of the future.

### **1.3. Applications of Sentiment Analysis**

- a. Review-related Websites
- b. Automatic summarization of user reviews
- c. Sentiment Analysis can also be used in trend prediction.
- d. Applications in smart homes

Studies in sociology and other fields have been aided by Sentiment Analysis systems that show trends in human emotions especially on social networks.

### **1.4. Challenges**

Sentiment Analysis is a very challenging task. It requires a deep understanding of the problem. We discuss some of the challenges faced in Sentiment Analysis.

#### **1.4.1. Identifying subjective portions of text.**

The same word can be treated as subjective in one context, while it might be objective in another. For example:

- The language of the author was very crude.
- Crude oil is extracted from the sea beds.

The same word “crude” is used as an opinion in the first sentence, while it is completely objective in the second sentence.

#### **1.4.2. Associating sentiment with specific keywords:**

Many sentences indicate an extremely strong opinion, but it is difficult to pinpoint the source of these sentiments. Hence an association to a keyword or phrase is extremely difficult. For example:

- Every time I read ‘Pride and Prejudice’ I want to dig her up and beat her over the skull with her own shin-bone.

In this example, “her” refers to the character in the book “Pride and Prejudice”, which is not explicitly mentioned. In such cases the negative sentiment must be associated with the character in the book

#### **1.4.3. Domain dependence:**

The same sentence or phrase can have different meanings in different domains. The word unpredictable is positive in the domain of movies, but if

the same word is used in the context of a vehicle's steering, then it has a negative connotation.

#### **1.4.4. Sarcasm Detection:**

Sarcastic sentences express negative opinions about a target using positive words.

For example: – Nice perfume. You must marinate in it.

#### **1.4.5. Thwarted expressions:**

There are some sentences wherein a minority of the text determines the overall polarity of the document. For example:

– This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up. Most of the words used here are positive, but the ultimate sentiment is negative.

#### **1.4.6. Indirect negation of sentiment:**

Sentiment can be negated in subtle ways as opposed to a simple no, not, etc. For example:

– It avoids all cliches and predictability found in Hollywood movies. While the words cliche and predictable bear a negative sentiment, the usage of “avoids” negates their respective sentiments.

#### **1.4.7. Order dependence:**

While in traditional text classification, the discourse structure does not play any role in classification, since the words are considered independent of each other, discourse analysis is essential for Sentiment Analysis/Opinion Mining. For example: – A is better than B, conveys the exact opposite opinion from, B is better than A.

#### **1.4.8. Entity Recognition:**

Not everything in a text talks about the same entity. We need to separate out the text about a particular entity and then analyze its sentiment. For eg:

– I hate Nokia, but I like Samsung.

A simple bag-of-words approach will mark this as neutral, however, it carries a specific sentiment for both the entities present in the statement.

#### **1.4.9. Identifying opinion holders:**

It is non-trivial to identify the opinion holders in any given piece of text. All that is written in a piece of text is not always the opinion of the author. For example, when the author quotes someone else, it becomes difficult to identify the source of that particular opinion. Such sentences are usually observed in news articles.

## **2. LITERATURE SURVEY**

## **2.1. Project Strategy**

Sentiment analysis deals with identifying and classifying opinions or sentiments which are present in source text. Social media is generating a huge amount of sentiment rich data in the form of tweets, status updates, reviews and blog posts etc. Sentiment analysis of this user generated data is very useful in knowing the opinion of the crowd.

Twitter sentiment analysis is an activity as Literature Survey on Sentiment Analysis of Twitter Data using Machine Learning Approaches compared to basic sentiment analysis due to the presence of slang words and misspellings. The maximum limit of characters that are allowed in Twitter is 140. Machine learning approach can be used for analyzing sentiments from the text. Some sentiment analysis is performed by analyzing twitter posts about electronic products like cell phones, computers etc. using Machine Learning approach. By performing sentiment analysis in a specific domain, it is possible to identify the effect of domain information in sentiment classification. They presented a new feature vector for classifying the tweets as positive, negative or neutral and extract people's opinion about products.



## **2.2. Tweets Mining**

Some researchers had an approach where posted tweets from the Twitter micro-blogging site are subjected to preprocessing and classified based on their emotional content as positive, negative and neutral or irrelevant; and compares the performance of various classifying algorithms based on their precision and recall in such cases. Further, the paper also discusses the applications of this research and its limitations.

A large number of machine learning like Naïve Bayes and Random Forest models performed sentiment analysis on product review data . Some work in this field included experiments with mood classification on blog posts. One of the research also deals with review of aspect-based opinion polling from unlabeled free-form textual

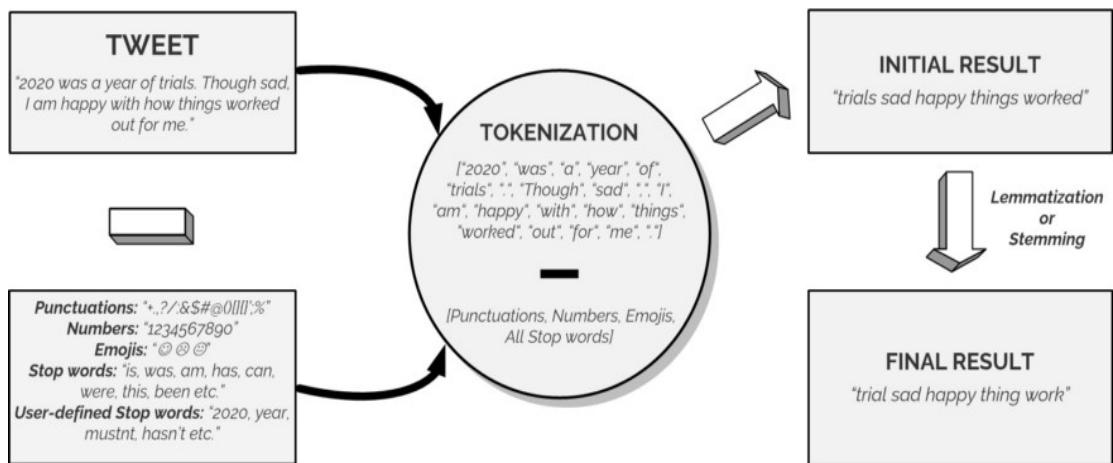
customer reviews without requiring customers to answer any questions. The tweet retrieval process needs access tokens from the twitter developer site and a piece of code which performs the operation of retrieving those tweets. As the base language used will be python, we choose to implement the Python library called Tweepy. This library is developed for the twitter API. With Tweepy library, you can easily integrate your Python application with the Twitter service

### 2.3. Technology

100% Pure Python - works on any Python Platform v3 or later, and Google App Engine ready, Zero dependency: Additional jars are not required, Built-in OAuth support, already built gzip support. There are some system requirements that need to be followed for the Tweepy python library to successfully operate. The library supports Windows and Unix Operating systems with Python 2.7 or higher versions installed on it. A python document is also provided in case a user needs to find the method name, syntax, or the root package while implementing the code. To use the python library, the user just needs to add the .py file to the python application class path.

### 2.4. Tweets Processing

The existing system works only on the dataset which is constrained to a particular topic. The existing systems also do not determine the measure of impact the results determined can have on the particular field taken into consideration and it does not allow retrieval of data based on the query entered by the user i.e. it has constrained scope.

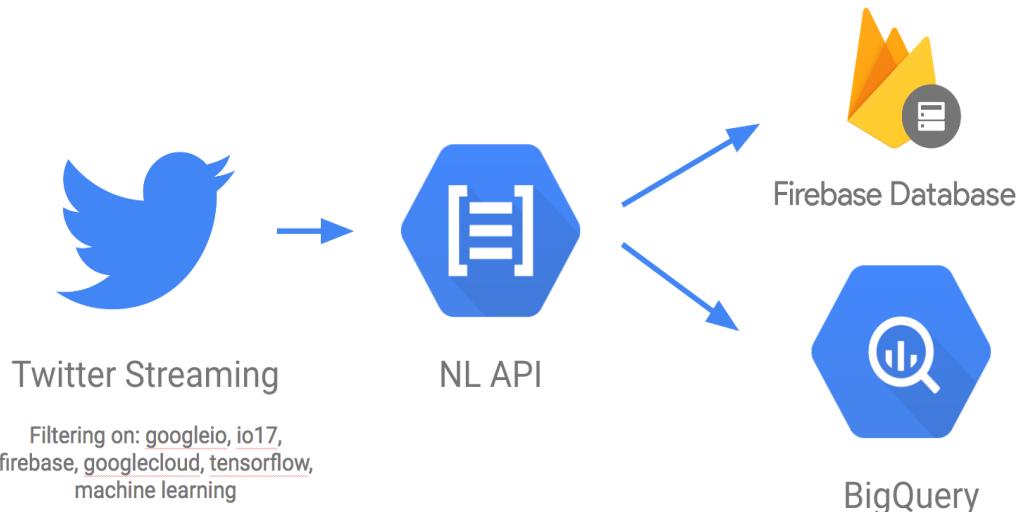


In simple words, it works on static data rather than dynamic data. Unsupervised algorithms like Vector Quantization, are used for data compression, pattern recognition, facial and speech recognition, etc. and therefore cannot be used in determining sentiment in twitter data. Apriorist algorithm fails to handle large datasets and as a result can generate faulty results. In the proposed system, we will retrieve tweets from twitter using twitter API based on the query. The Collected tweets will be

subjected to preprocessing. We will then apply the supervised algorithm on the stored data. The supervised algorithm used in our system is TextBlob using TensorFlow.

## 2.5. Machine Learning

The object of this TensorFlow is to show some of the top NLP solutions specific in deep learning and some in classical machine learning methods. It's important to be awarded that for getting competition results all the models proposed in this post should be training on a bigger scale (GPU, more data, more epochs, etc.).



The results of the algorithms i.e., the sentiment will be represented in graphical manner (bar charts). The proposed system is more effective than the existing one. This is because we will be able to know how the statistics determined from the representation of the result can have an impact in a particular field.

### **3. PROJECT DESIGN**

### **3.1. Project task set**

Aim is to collect all the tweets and then run our algorithm to judge which person's thoughts are more effective in public. So there are mainly three (4) steps we will be working on,

- 1. TOOLS**
- 2. DATASET CREATION**
- 3. DATA ANALYSIS**
- 4. DATA VISUALIZATION**

#### **3.1.1. TOOLS**

1. **Python** — a programming language
2. **Tweepy** — a type of RESTful API specifically for Twitter
3. **Textblob** — processed textual data library tool (already trained on numerous textual data.)
4. **Pandas** — data manipulation and analysis library
5. **NumPy** — scientific computing library
6. **Matplotlib** — plotting library
7. **Plotly** — plotting library
8. **Seaborn** — Data visualization library based on Matplotlib
9. **Word Cloud** — library for a visual representation of textual data

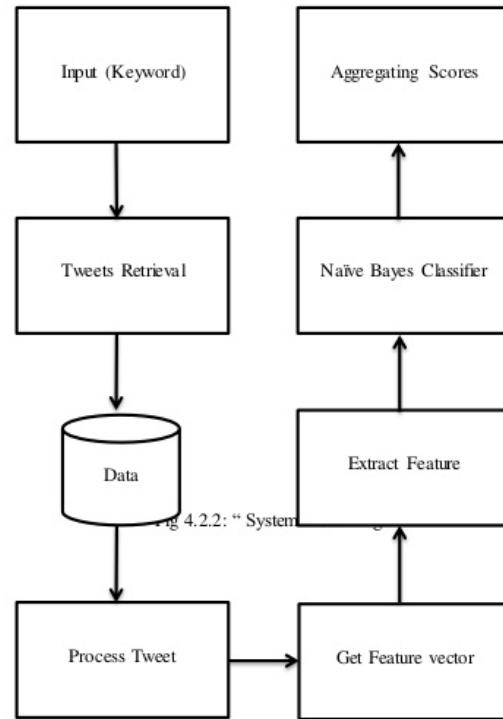
#### **3.1.2. DATA CREATION**

First, we will collect all the tweets. We will clean the data so that unwanted fields can be removed. Feature Extraction is the main key to prepare the data and return all tweets into a CSV file.



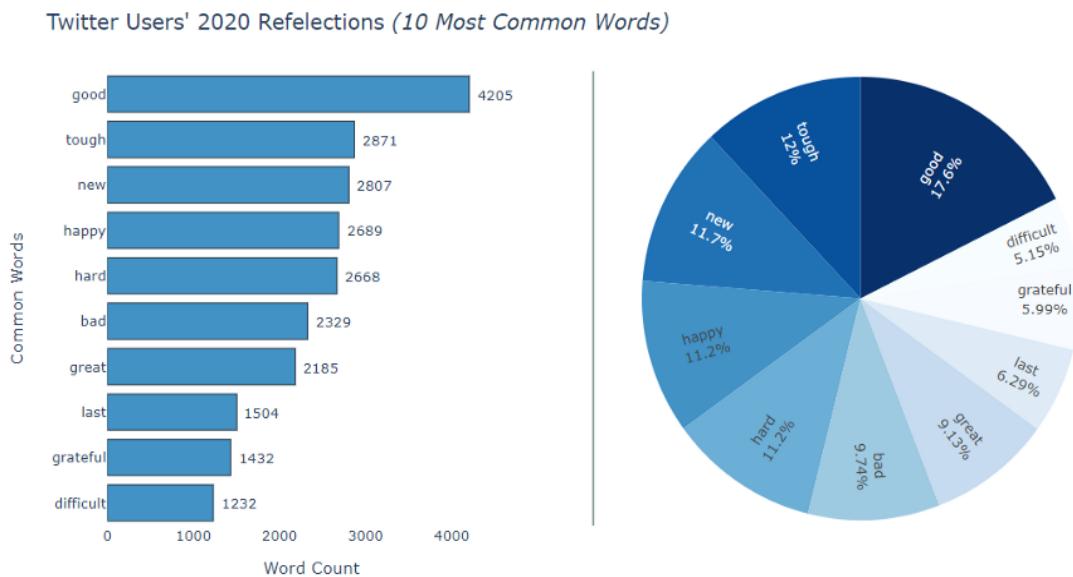
### 3.1.3. ALGORITHM SELECTION AND TRAIN\_TEST\_SPLIT

Creation of a Model with the best possible algorithm is the next step. There are various algorithms we can run to test on what conditions our problem can be solved in less time. We are going to use TextBlob, Vader, Flair and a customized algorithm using tensorflow to check which algorithm has higher accuracy. After selection on the algorithm, we split the dataset into the Training and testing part.



### 3.1.4. DATA VISUALIZATION

The last step is data visualization. We deploy our application and return the result with the help of data visualization, there can be so many ways to return the output in pie graph or in histogram and so many others.



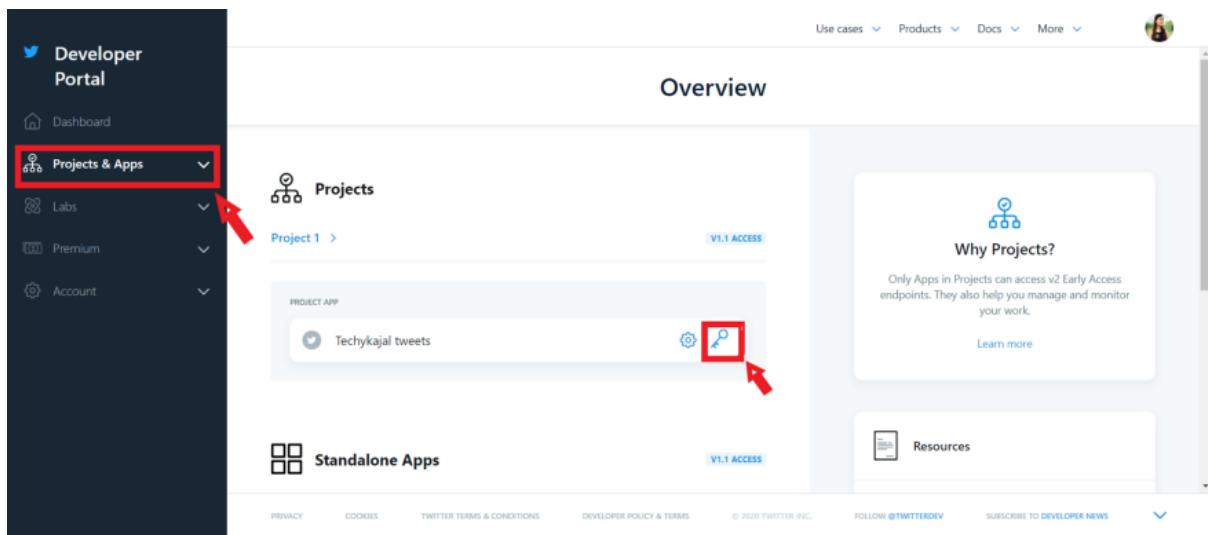
## **4. IMPLEMENTATION**

#### 4.1. DATA CREATION

In this part, we will scrap the replies of tweets done by both the leaders using Python Web Scraping. We can also scrap tweet replies using software if you don't want to use Twitter's API.

**Steps 1.** Register and create a Twitter Developer Account For API

**Steps 2.** Go to the developer dashboard and click Projects and apps and then click Keys icon.



**Steps 3.** Copy “API Key”, “API Secret”, “Access Token”, and “Access Token Secret”



#### Authentication Tokens (i)

Bearer Token  
Generated Jul 27, 2020

[Regenerate](#)

[Revoke](#)

Access Token & Secret  
Generated Jul 27, 2020

[Regenerate](#)

[Revoke](#)

*Created with Read, Write, and Direct Messages permissions*

```
CONSUMER_KEY      = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxx'
CONSUMER_SECRET   = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
ACCESS_TOKEN      = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxx-xxxxxxxxxxxxxxxxxxxx'
ACCESS_TOKEN_SECRET= 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
```

**Note** - Keep these keys safe and private, also Consumer Key is the same as API Key.

**Steps 4.** Download Tweepy using the pip package manager

```
pip install tweepy
```

**Steps 5.** Import Tweepy, CSV and Enter the Oauth keys. Tweepy is used for accessing Twitter API using python. CSV module is used here to write scraped tabular data in CSV(comma-separated values) format. SSL (Secure Sockets Layer) provides peer authentication facilities for network sockets, both client-side and server-side.

**Steps 6.** Setup Authentication with Twitter

Using OAuthHandler passes the credentials to allow access to Twitter's API features.

```
authenticate    = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
authenticate.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
api            = tweepy.API(authenticate, wait_on_rate_limit = True)
```

**Steps 7.** Enter twitter username which you want to analyse. Get all the tweets of that particular person and store them in posts variable.

```
USER1_TWITTER_USERNAME = "narendramodi"
USER2_TWITTER_USERNAME = "MamataOfficial"

USER1_POSTS = api.user_timeline(
                screen_name = USER1_TWITTER_USERNAME,
                count = 1000,
                lang = "en",
                tweet_mode = "extended"
            )

USER2_POSTS = api.user_timeline(
                screen_name = USER2_TWITTER_USERNAME,
                count = 1000,
                lang = "en",
                tweet_mode = "extended"
            )
```

**Steps 8.** Create dataframe from fetched tweets

```
USER1_DATAFRAME = pd.DataFrame(  
    [tweet.full_text for tweet in posts],  
    columns = ['Tweets'])  
  
USER2_DATAFRAME = pd.DataFrame(  
    [tweet.full_text for tweet in posts1],  
    columns = ['Tweets'])
```

**Steps 9.** Removing unwanted fields from data.

```
def cleantxt(text):  
    text = re.sub(r'@[A-Za-z0-9]+', '', text)  
    text = re.sub(r'#', '', text)  
    text = re.sub(r'https?:\/\/\S+', '', text)  
    text = re.sub(r'RT[\s]+', '', text)  
    return text  
  
def cleantxt1(text):  
    text = re.sub(r'https?:\/\/\S+', '', text)  
    text = re.sub(r'RT[\s]+', '', text)  
    return text  
  
USER1_DATAFRAME['Tweets'] = USER1_DATAFRAME['Tweets'].apply(cleantxt)  
USER2_DATAFRAME['Tweets'] = USER2_DATAFRAME['Tweets'].apply(cleantxt1)
```

Now your data is saved into dataframes and ready to be visualized.

## 4.2. DATA ANALYSIS

Since our first task is completed of getting the data in clean form. Now our next task is to select a suitable algorithm so that we can test the accuracy. So we start from Sentiment analysis using TextBlob.

### 4.2.1. TEXTBLOB

TextBlob is a python library and offers a simple API to access its methods and perform basic NLP tasks. Here, we are using this library to perform text classification in either positive or negative on the basis of sentiment analysis.

This library is just like a Python string with the functionality that you can easily use its functions. It provides a really cool functionality that can easily

summarize the text, provide you with sentiments of the text, spelling correction, translation, and language detection and so more...

There are few other NLP libraries in python such as Spacy, NLTK, gensim which also provides you with many functionalities to deal with textual data. Here we also used some other libraries like Vader and Flair and one self customised library, but when we run the algorithms, TextBlob has the highest accuracy, so we decided to choose this.

#### 4.2.2. Polarity

Polarity ranges from -1 to +1(negative to positive) and tells whether the text has negative sentiments or positive sentiments. Polarity tells about factual information.

```
def getsubjectivity(text):
    return TextBlob(text).sentiment.subjectivity

def getpolarity(text):
    return TextBlob(text).sentiment.polarity

USER1_DATAFRAME['subjectivity'] = USER1_DATAFRAME['Tweets'].apply(getsubjectivity)
USER1_DATAFRAME['polarity'] = USER1_DATAFRAME['Tweets'].apply(getpolarity)

USER2_DATAFRAME['subjectivity'] = USER2_DATAFRAME['Tweets'].apply(getsubjectivity)
USER2_DATAFRAME['polarity'] = USER2_DATAFRAME['Tweets'].apply(getpolarity)
```

#### 4.2.3. Subjectivity

Subjectivity also ranges from -1 to +1( negative to positive) . So more +ve subjectivity means less factual data and mostly public opinion.

	Tweets	subjectivity	polarity
0	वरिष्ठ पत्रकार शेष नारायण सिंह जी का निधन अत्य...	0.000000	0.000000
1	'Jal', 'Thal' and 'Nabh'...our armed forces ha...	0.000000	0.000000
2	: Fighting the Invisible Enemy: MoD's Response...	0.400000	0.200000
3	ગુજરાતના પૂર્વ નાણામંત્રી શ્રી મૃતાપભાઈ શાહના...	0.000000	0.000000
4	Reviewed various aspects of the COVID-19 respo...	0.500000	0.000000
...	...	...	...
195	: Handing over of a Fast Patrol Vessel to Seyc...	0.500000	0.200000
196	: Joint e-inauguration of the new Magistrates'...	0.454545	0.136364
197	Speaking at the meeting with Chief Ministers.	0.000000	0.000000
198	Boosting friendship with Seychelles.	0.000000	0.000000
199	India will pay a fitting tribute to the great ...	0.625000	0.650000

So, In the above output 2nd statement, you can see that subjectivity is 0.9 which is indicating text contained at 500th row is barely a personal opinion.

#### 4.2.4. Neutral comments with ‘zero’ Polarity and subjectivity

There are many cases where polarity is zero because there is some data which either doesn’t contain any text or simply have links or hashtags only.

After the Train\_Test\_Split part takes place and we show the result through visualizations.

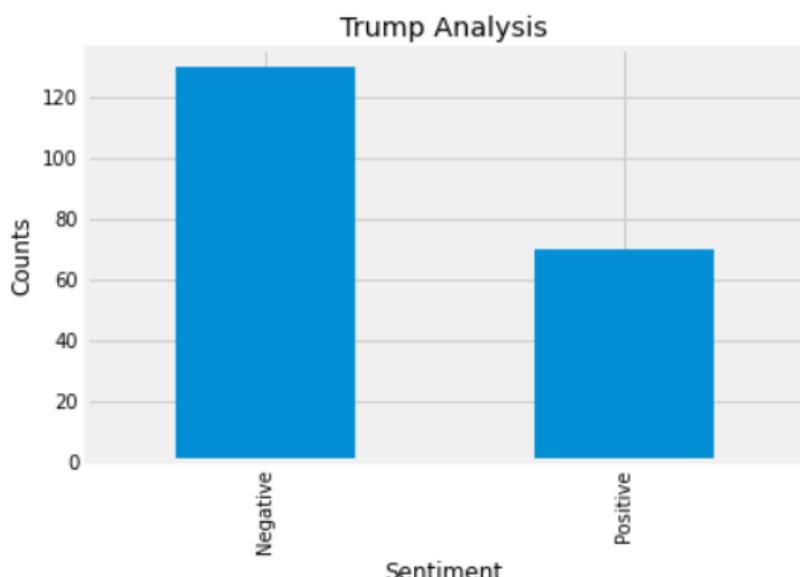
### 4.3. DATA VISUALIZATION

Visualizing data gives you a clearer picture of what we are actually doing. So it’s the most important step that we included into our project for making it understand better in the simplest and easy way possible in your presentations. And also, it is an important step before applying any analysis and modeling.

Here, I am comparing Positive/Negative tweets on Trump’s tweets with that of Joe Biden to get a better understanding through visualization of who will be going to win this election.

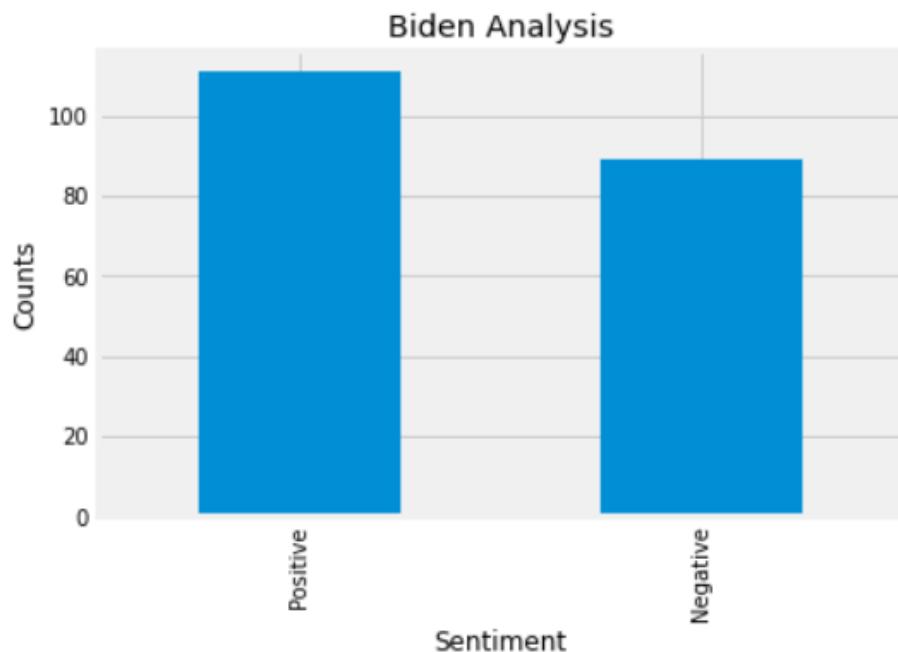
#### 4.3.1. DONALD TRUMP

```
▶ import matplotlib.pyplot  
  
matplotlib.pyplot.title('Trump Analysis')  
matplotlib.pyplot.xlabel('Sentiment')  
matplotlib.pyplot.ylabel('Counts')  
df['Analysis'].value_counts().plot(kind='bar')  
matplotlib.pyplot.show()
```



#### 4.3.2. JOE BIDEN

```
▶ import matplotlib.pyplot  
  
matplotlib.pyplot.title('Biden Analysis')  
matplotlib.pyplot.xlabel('Sentiment')  
matplotlib.pyplot.ylabel('Counts')  
df1['Analysis'].value_counts().plot(kind='bar')  
matplotlib.pyplot.show()
```



#### 4.3.3. Analyzing Most Positive and Most Negative replies on both the handles

One can have insights from the below-attached outputs that are the keywords in a sentence that contribute to either polarity 1 or -1. And also, you can analyse how good this TextBlob sentiment analysis is.

```

def getanalysis(score):
    if (score < 0.05):
        return 'Negative'
    else:
        return 'Positive'

USER1_DATAFRAME['Analysis'] = USER1_DATAFRAME['polarity'].apply(getanalysis)
USER2_DATAFRAME['Analysis'] = USER2_DATAFRAME['polarity'].apply(getanalysis)

USER1_DATAFRAME['Analysis'].value_counts()

Negative    132
Positive     68
Name: Analysis, dtype: int64

USER2_DATAFRAME['Analysis'].value_counts()

Positive    113
Negative     87
Name: Analysis, dtype: int64

```

#### 4.3.4. WORD CLOUDS

A ‘word cloud’ is a visual portrayal of word recurrence. The more generally the term shows up inside the content being dissected, the bigger the word shows up in the picture produced. Word clouds are progressively being utilized as a straightforward device to recognize the focal point of composite material.

Word clouds can be useful to find your customers’ pain points for business purposes, I am hereby using it to get insights of public opinion about their leader and most frequently used keywords by the citizens against their leaders.

##### 4.3.4.1. DONALD TRUMP WORD CLOUD

```

allwords=' '.join([twts for twts in df['Tweets']])
wordcloud=WordCloud(width=5000,height=5000,random_state=9).generate(allwords)

import matplotlib.pyplot
matplotlib.pyplot.imshow(wordcloud,interpolation='bilinear')
matplotlib.pyplot.axis('off')
matplotlib.pyplot.show()

```



## **5. CODING RELATED**

### **5.1. PYTHON:**

It is often used as a “scripting language” for web applications. This means that it can automate specific series of tasks, making it more efficient. Consequently, Python is often used in software applications, pages within a web browser, the shells of operating systems and some games.

### **5.2. TWEETPY:**

It is open-sourced, hosted on GitHub and enables Python to communicate with Twitter platform and use its API.

### **5.3. CSV module:**

It is used here to write scraped tabular data in CSV(comma-separated values) format. TEXTBLOB: It is a Python (2 and 3) 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 phrase extraction, sentiment analysis, classification, translation, and more.

### **5.4. PANDAS:**

It is the most popular Python library that is used for data analysis. It provides highly optimized performance with back-end source code is purely written in C or Python. We can analyze data in Pandas with Series.

### **5.5. NUMPY:**

It is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, Fourier transform, and matrices.

### **5.6. MATPLOTLIB:**

It is a collection of functions that make matplotlib work like MATLAB. 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.

### **5.7. WORDCLOUD:**

A tag cloud is a novelty visual representation of text data, typically used to depict keyword metadata on websites, or to visualize free form text.

### **5.8. TENSORFLOW:**

It is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

## **6. DEPLOYMENT**

## **6.1. VERSION CONTROL SYSTEM**

- 6.1.1.** We have a GitHub repository and we will push all the code data into it. One important Thing, we also need to make a file called requirements.txt file and include all the Dependencies into that file.
- 6.1.2.** Repository distribution should be based on HEROKU deployment steps
- 6.1.3.** With respect to the project, our main.py file which we have made through FLASK app And pickle file in accordance with HEROKU deployment steps.

## **6.2. Install Heroku Command line interface**

- 6.2.1.** Create an account on Heroku cloud platform
- 6.2.2.** Install Heroku command line interface on local system

## **6.3. Create Procfile**

- 6.3.1.** Procfile is responsible for the commands that are executed by Heroku app on startup. a gunicorn driver is used.

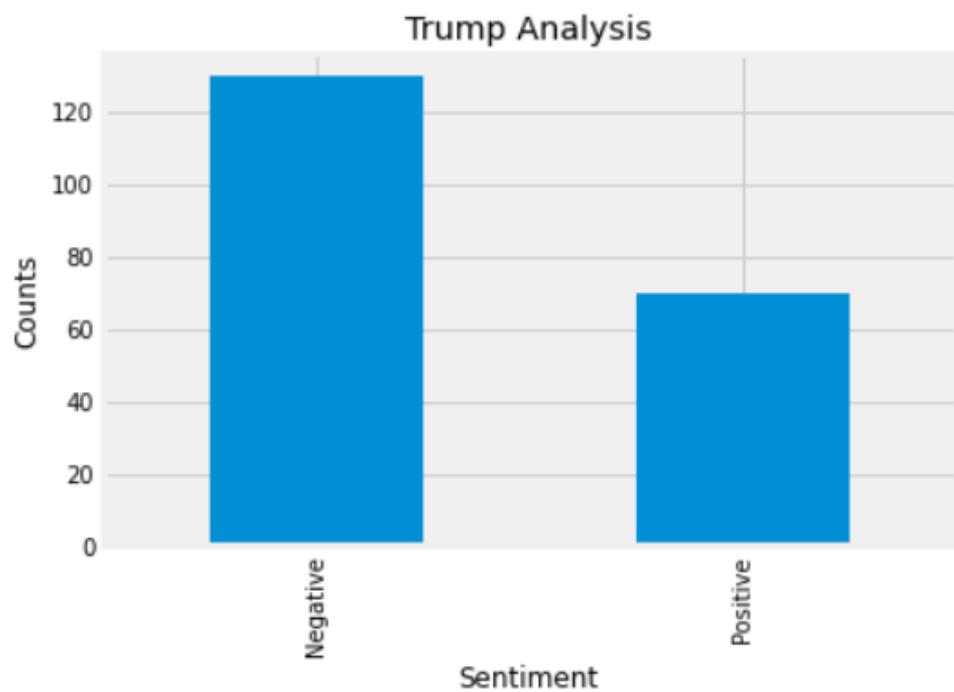
## **6.4. Heroku Dashboard**

- 6.4.1.** Create a repository on Heroku and link that repository with the project repository on Github. Push all the files using Heroku command line interface on that repository

## 6.5. DEMO ANALYSIS:

### 6.5.1. TRUMP'S SENTIMENT

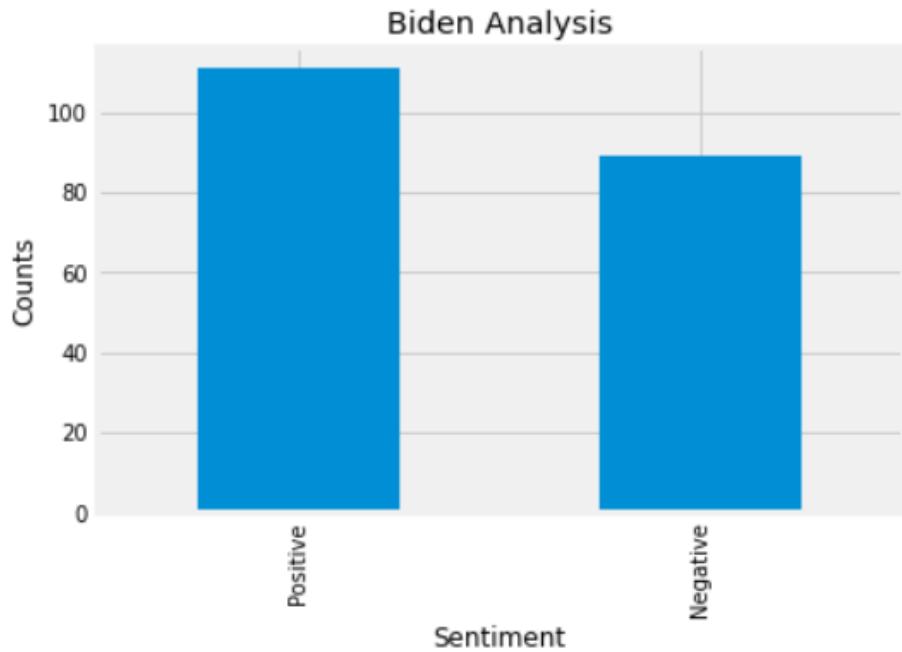
```
▶ import matplotlib.pyplot  
  
    matplotlib.pyplot.title('Trump Analysis')  
    matplotlib.pyplot.xlabel('Sentiment')  
    matplotlib.pyplot.ylabel('Counts')  
    df['Analysis'].value_counts().plot(kind='bar')  
    matplotlib.pyplot.show()
```



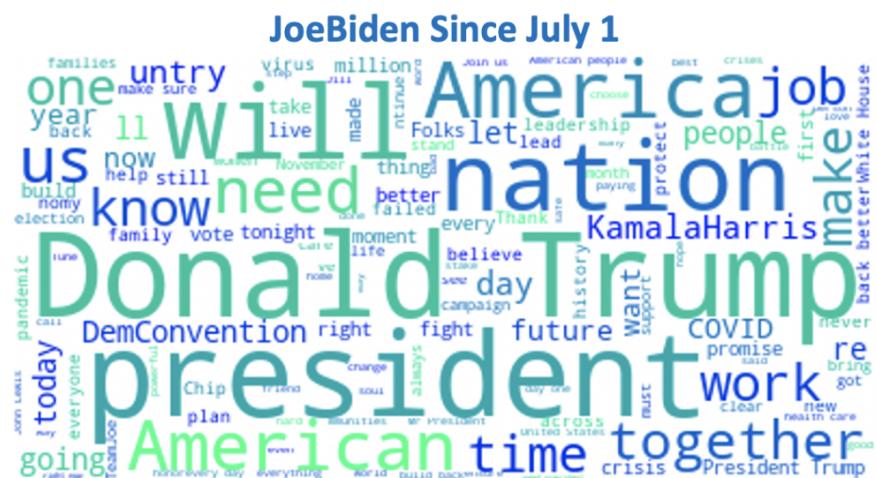
### 6.5.2. BIDEN'S SENTIMENT

```
import matplotlib.pyplot

matplotlib.pyplot.title('Biden Analysis')
matplotlib.pyplot.xlabel('Sentiment')
matplotlib.pyplot.ylabel('Counts')
df1['Analysis'].value_counts().plot(kind='bar')
matplotlib.pyplot.show()
```



6.6. **Analysis Result:** We can clearly see Biden's Tweets get more Positive Response than Trump's.





## **7. TESTING**

## 7.1. TESTING

We performed some tests to check if our model was working fine or not. Now the case mentioned below, is the result of all the successful tests.

### 7.2. Text Case No. 1

**Title:** Sentiment Analysis on US Election Campaign

**Aim:** To check is everything is working fine i.e we are getting the graph of polarity and subjectivity correctly or not.

**Procedure:** Run The Code , we can see which person tweets getting a positive response and which one is getting a negative response.

**Expected Output:** Joe Biden's Polarity graph is more accurate than Donald Trump's graph.

**Observed Output:** Joe Biden's Polarity graph is slightly more accurate than Donald Trump's graph.

**Passed/Failed:** Passed.

### 7.3. Text Case No. 2

**Title:** Sentiment Analysis on West Bengal Election Campaign

**Aim:** To check is everything is working fine i.e we are getting the graph of polarity and subjectivity correctly or not.

**Procedure:** Run The Code , we can see which person tweets getting a positive response and which one is getting a negative response.

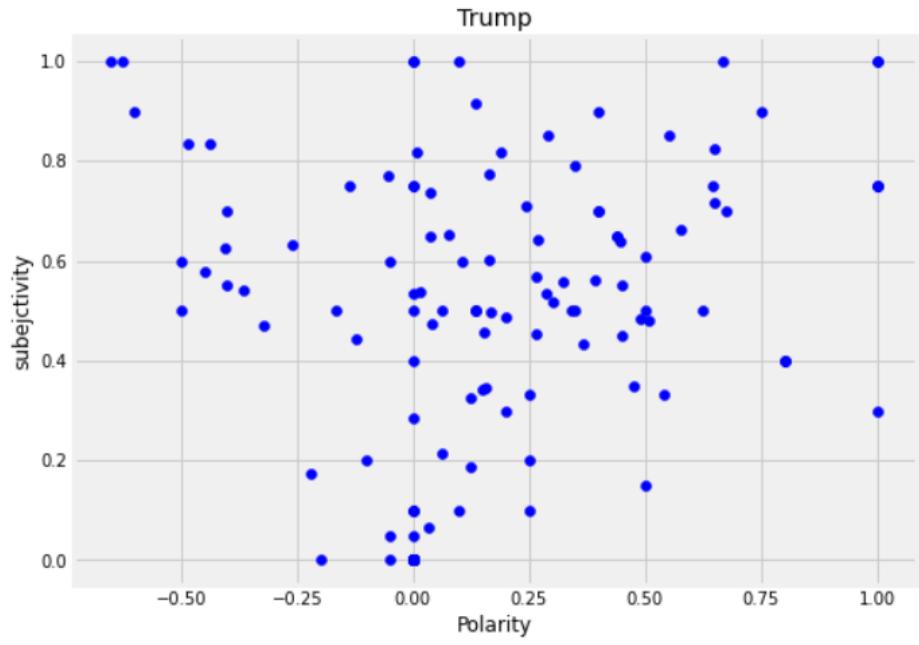
**Expected Output:** Mamta banerjee's Polarity graph is more accurate than Narendra Modi's graph.

**Observed Output:** Mamta banerjee's Polarity graph is slightly more accurate than Narendra Modi's graph.

**Passed/Failed:** Passed.

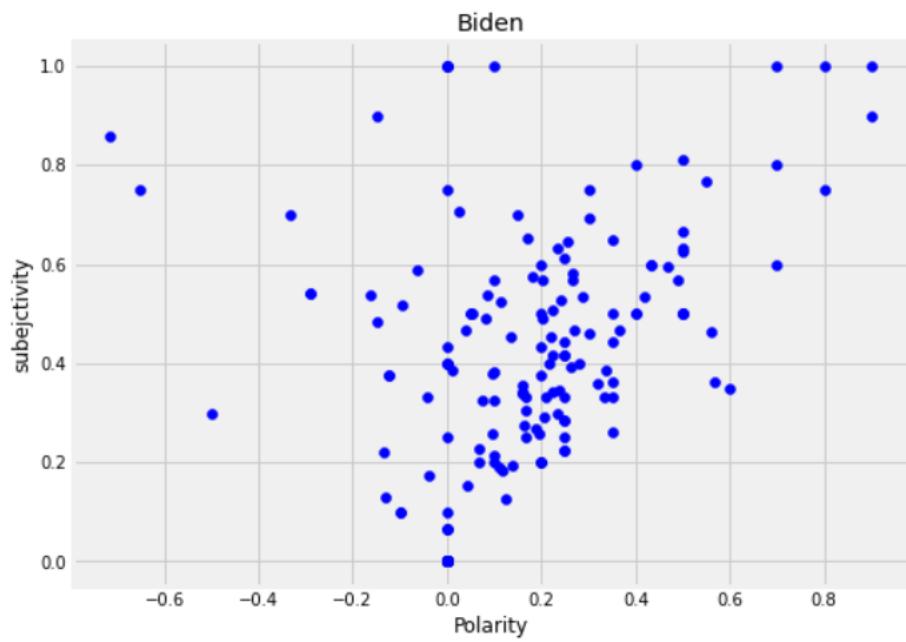
### 7.4. TRUMP'S POLARITY

```
▶ import matplotlib.pyplot
    matplotlib.pyplot.figure(figsize=(8,6))
    for i in range(0,df.shape[0]):
        matplotlib.pyplot.scatter(df['polarity'][i],df['subjectivity'][i],color='Blue')
    matplotlib.pyplot.title('Trump')
    matplotlib.pyplot.xlabel('Polarity')
    matplotlib.pyplot.ylabel('subjectivity')
    matplotlib.pyplot.show()
```



## 7.5. BIDEN'S POLARITY

```
import matplotlib.pyplot
matplotlib.pyplot.figure(figsize=(8,6))
for i in range(0,df.shape[0]):
    matplotlib.pyplot.scatter(df1['polarity'][i],df1['subjectivity'][i],color='Blue')
matplotlib.pyplot.title('Biden')
matplotlib.pyplot.xlabel('Polarity')
matplotlib.pyplot.ylabel('subjectivity')
matplotlib.pyplot.show()
```



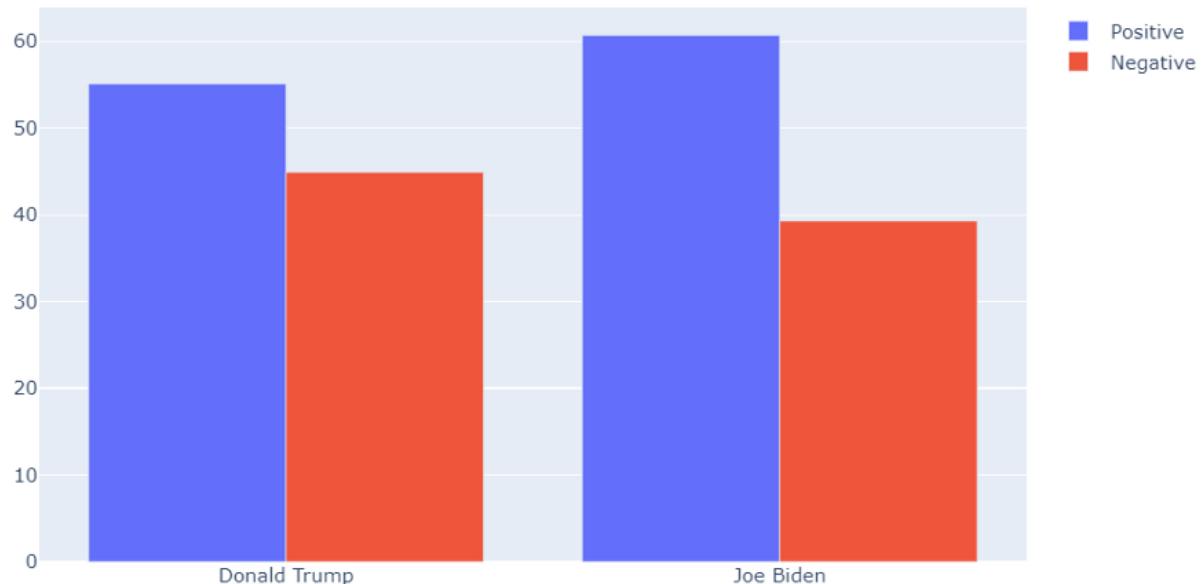
## **8. CONCLUSION**

Sentiment analysis has become an important factor in the decision making process in a particular field.

In this paper we discussed techniques for preprocessing and information retrieval of tweets through twitter. Also we studied the Deep Learning technique: TensorFlow and TextBlob for text categorization which can be used to find out the polarity of textual tweet.

From study we can conclude that TextBlob acknowledges some properties of text like High Dimensional feature space, few irrelevant features, sparse instance vectors. The performance of TensorFlow can be evaluated using precision and recall. Different results show that TextBlob gives good performance on text categorization as compared with Flair and Vader.

With the ability to generalize high dimensional feature space, TextBlob eliminates the need for feature selection.



## **9. FUTURE SCOPE**

### **9.1. BASIS:**

The future scope of this project is immense as it can be used to know the region wise mood of people regarding the present leader and opposition leader which can aid voters to get holistic view about how the leader and make their decision on the actual ground realities rather than going with recency bias and voting by propaganda.

### **9.2. 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.

### **9.3. AREAS OF IMPROVEMENT**

This analysis has some shortcomings and I would like to point them out over here. As per my observation, here are some of them:

#### **9.3.1. The distribution of the data is poor:**

I pointed this out earlier during my analysis and I feel that this may have included some bias in my analysis. Ideally, we should have the same number of tweets for all states for both candidates. But in the real world, this is difficult to achieve if not impossible. However, an area of improvement is including some distribution models to get my dataset towards a uniform distribution of states.

#### **9.3.2. The location identification part of the analysis needs improvement:**

As discussed earlier, ideally the location of the tweet can be fed to the Google Maps API to get the state from where the tweet was made if it is from the United States.

#### **9.3.3. The prediction of the judgement for a state can be improved:**

The logic which I used for predicting the overall judgement of a state whether Democratic or Republican can be improved. I used my limited technical knowledge and nous on this domain to make a logic for this calculation but this is open to further suggestions and improvements.

## **10. REFERENCES**

- 10.1. IEEE Paper: **Document Analysis Support for the Manual Auditing of Elections** by D. Lopresti
- 10.2. IEEE Paper: 2012 Presidential Elections on Twitter -- **An Analysis of How the US and French Election were Reflected in Tweets** by V.Arunachalam.
- 10.3. **The Year 2020: Analyzing Twitter Users' Reflections using NLP: A Sentiment Analysis Project using Python and Tableau** by Jessica Uwoghiren (towardsdatascience.com)
- 10.4. **Predicting US Presidential Election Result Using Twitter Sentiment Analysis** with Python - Kajal Yadav