TWEET CLASSIFICATION AND TREND DETECTION USING NLP

A CAPSTONE PROJECT REPORT

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

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

by

NAMAN VEERAMACHENENI (17BCE7003)

Under the Guidance of

DR. KARTHIKEYAN S

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
VIT-AP UNIVERSITY
AMARAVATI- 522237

JANUARY 2020

CERTIFICATE

This is to certify that the Capstone Project work titled "**Tweet Classification and Trend Detection using NLP**" that is being submitted by **Naman Veeramacheneni(17BCE7003)** is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of bonafide work done under my guidance. The contents of this Project work have not been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Dr. KARTHIKEYAN S
Project Guide

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to thank Dr. Karthikeyan Saminathan for his valuable guidance and immense support throughout the project work as my project guide and advisor. Data analytics is growing very fast, Dr. Karthikeyan always keeps herself up to date with the latest research and encourages his students to work towards cut edge technologies. I am amazed by his positive energy and patience. His unparalleled knowledge, agile and prompt feedback coupled with smart ideas have helped me immensely in putting up the whole work.

I wish to thank the members of my review committee: Dr. Mangalraj P, for generously offering his time and support. His feedback and suggestions helped me in achieving

I would like to thank Vellore Institute of Technology, AP for providing me with a platform to plan and execute my project work.

ABSTRACT

Social media often plays a crucial role in disseminating information to warn the public about health concerns. Twitter is the most popular social media that allows its users to spread and share information. They publish these topics on the list called “Trending Topics”. It shows what is happening in the world and what people's opinions are about it. In this project, proposes a plan to develop a novel framework for topic sentiment trend detection and prediction in social media.The proposed framework copes with the following tasks: topic trend detection, sentiment analysis, and topic prediction. The VADER-based time series sentiment analysis methods were applied to analyze the social media data from the public, social media news, and newspapers. The results in this project have shown some exciting findings from topic sentiment detection and high accuracies from the topic trend prediction.

INTRODUCTION

In social media, millions of active users express their opinions and interact with each other daily. Such users' content in the form of posts or tweets provides a vast amount of useful information if analyzed carefully. Therefore, the data streamed from social media such as Twitter, Facebook, or Instagram is so precious for researchers to perceive the users' social behavior through NLP. A massive amount of user-generated online content is freely available to the real-time monitoring of public sentiment.

It is difficult to find the contextual sentiment of a text. Sentiment analysis is one of the critical issues today. The primary job is to fast-pace the process of opinion extraction from the given subject. The subject here can be an excerpt from the written text, debate, or day to day conversation. In sentiment analysis, we also evaluate the positive and negative intensities of symbols and words. Sentiment analysis helps to improve customer services, Political planning Policies, and manufacturing quality products.

Online platforms like social media and blogs are widely used by public and mass media to express their opinions during the crisis. Moreover, sentiment analysis is also performed on those opinions to better understand the emotion attached to those opinions. Topic modeling on Social media gives us better insights into public view during an epidemic. Notably, in social media, people are widely expressing their problems related to the government and elections. Moreover, the analysis of the data is useful to decipher the change in opinions and trends of people.

PROJECT TOOLS

- Anaconda Navigator — Virtual Environment (Jupyter Notebook)
- Python — a programming language
- Tweepy — a type of RESTful API specifically for Twitter
- Textblob — processed textual data library tool (already trained on numerous textual data.)
- Pandas — data manipulation and analysis library
- NumPy — scientific computing library
- Matplotlib — plotting library
- Plotly — plotting library
- Seaborn — Data visualization library based on Matplotlib
- Wordcloud — library for a visual representation of textual data

PROJECT METHODOLOGY

The basic methodology of the project:

•Collecting high volumes of data to create an efficient prediction model.

•Pre-processing data and cleansing of data.

•Detecting topic trends and applying sentimental analysis

•Displaying result interpretations and prediction.

This project could be divided into 3 parts

1. **Dataset creation:**

A huge amount of data is required to get accurate results. To create the dataset we would first need to scrape public tweets from home pages of both presidential candidates. Tweepy tool allows us to get tweets and re-tweets and also connects us to the twitter API. One would require a twitter developer account to get access to the API by using authentication keys. By giving account handle and account ID, we would be able to store all the tweets in a CSV file

- 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.

2. Data Analysis

The next task is to analyze data and clean up the text which isn't returning any meanings and apply our algorithm for classifying text into either positive sentiments or negative sentiments.

The dataset contains two attributes in total, and only the replies column is for consideration, the other one wouldn't add any value to the sentiment analysis. A correlation between different attributes is necessary to choose the most important ones which is also known as feature selection, a widely used technique for dimensionality reduction.

Sentiment analysis using TextBlob:

TextBlob is a python library and offers a simple API to access its methods and perform basic NLP tasks. Here, I am 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 can easily use its functions. It provides an elaborate functionality that can easily summarize the text, provide with sentiments of the text, spelling correction, translation, and language detection etc.

The two important tools to classify data between positive and negative are:

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.

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

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. Such data can be dropped for better results.

It is essential to balance both datasets after elimination neutral comments for proportional analysis and fair results.

3. **Data Visualization:**

Visualizing data gives us a clearer picture of what we are actually doing. It is an important step before applying any analysis and modeling.

Here, I am comparing Negative tweets on Trump's tweets with that of Joe Biden to get a better understanding through visualization.

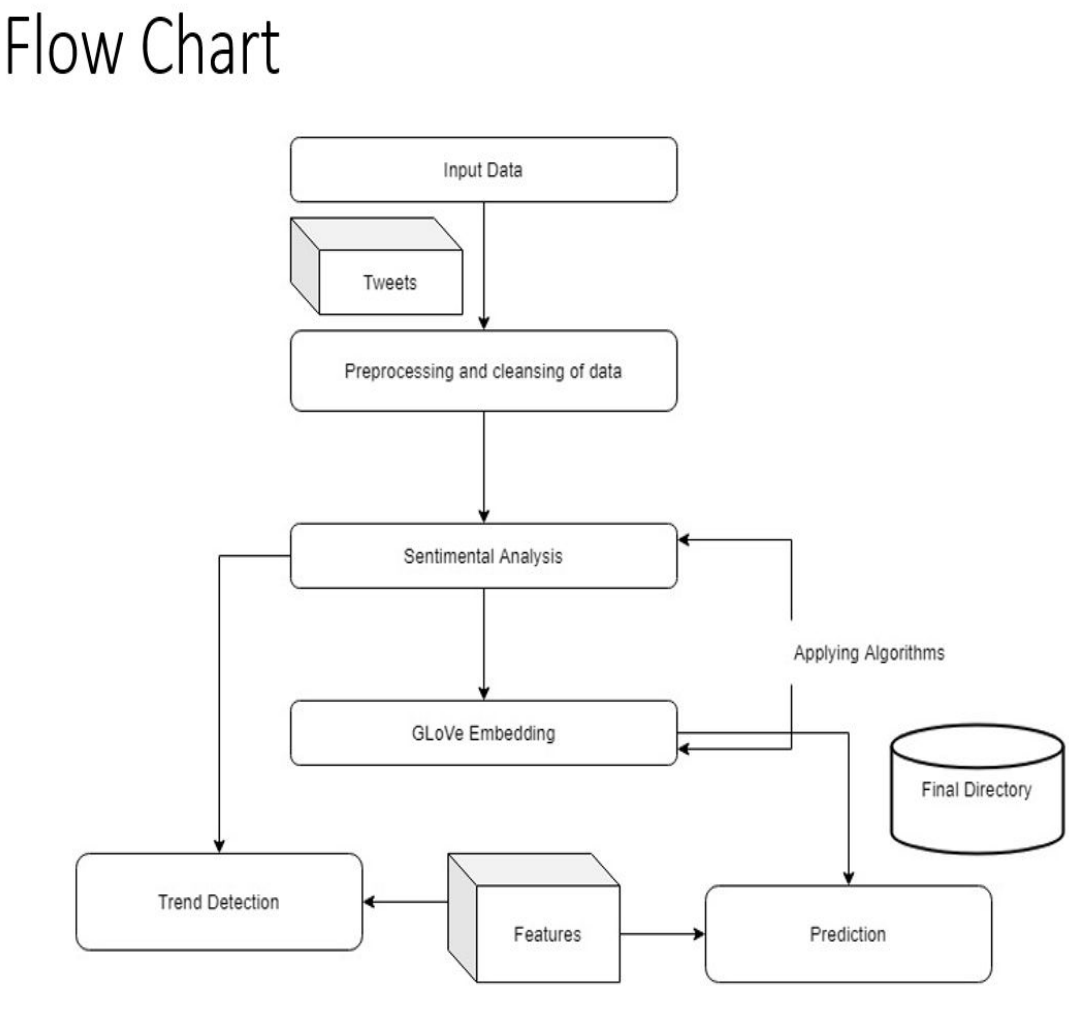
Representing negatives and positive comments of both candidates using boxplot gives a more mathematical understanding of the analysis and makes it easy to deduce results.

WordCloud for both candidates: 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.

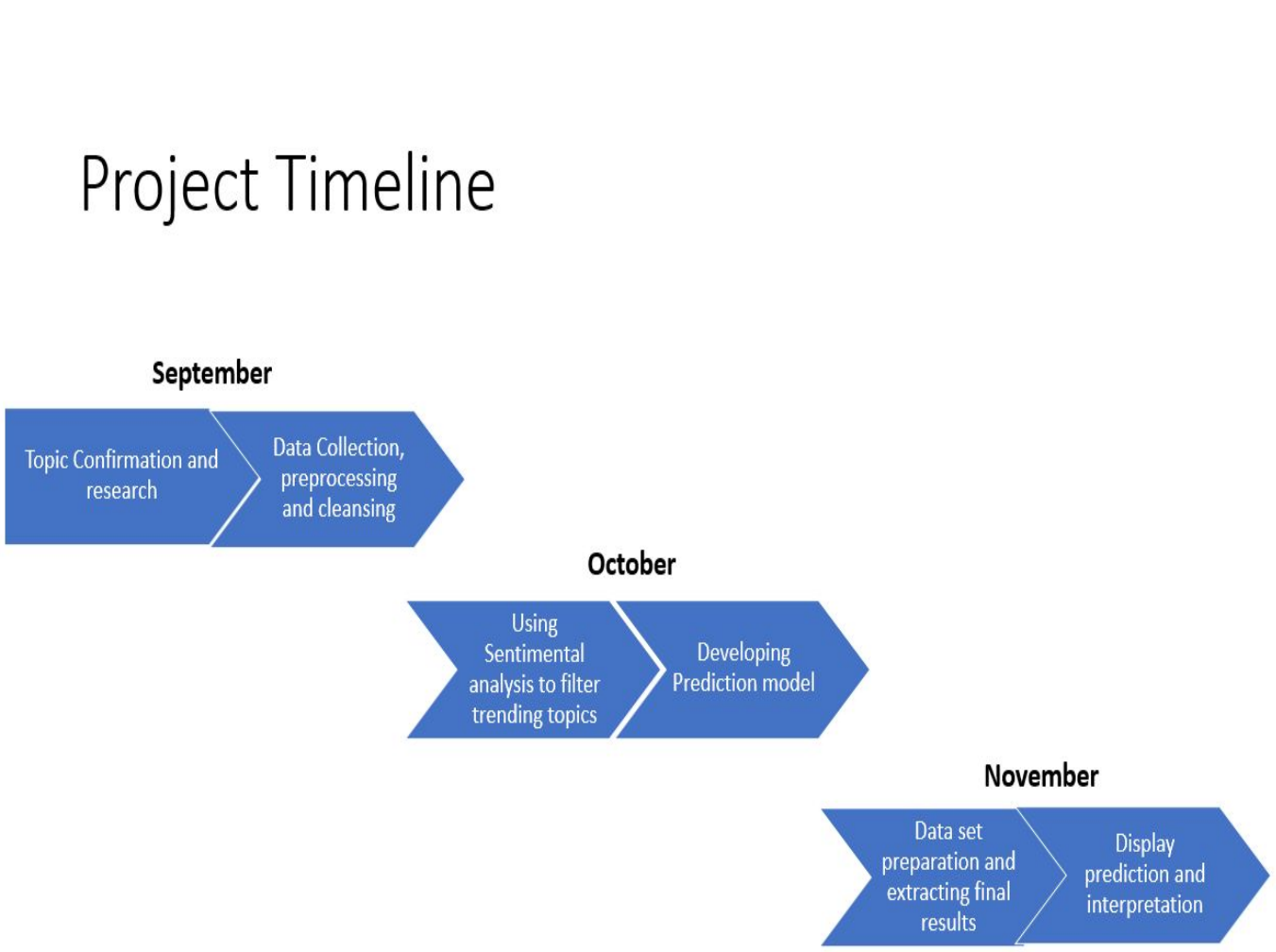
Word clouds can be useful to find customers' pain points in business purposes, I am hereby using it to get insights of public opinion about their leader and most frequently used keywords by the citizens against their leaders.

Pie chart representation of positive comments of both candidates can be used to conclude the winner of the election by public's twitter opinion.

PROJECT FLOWCHART

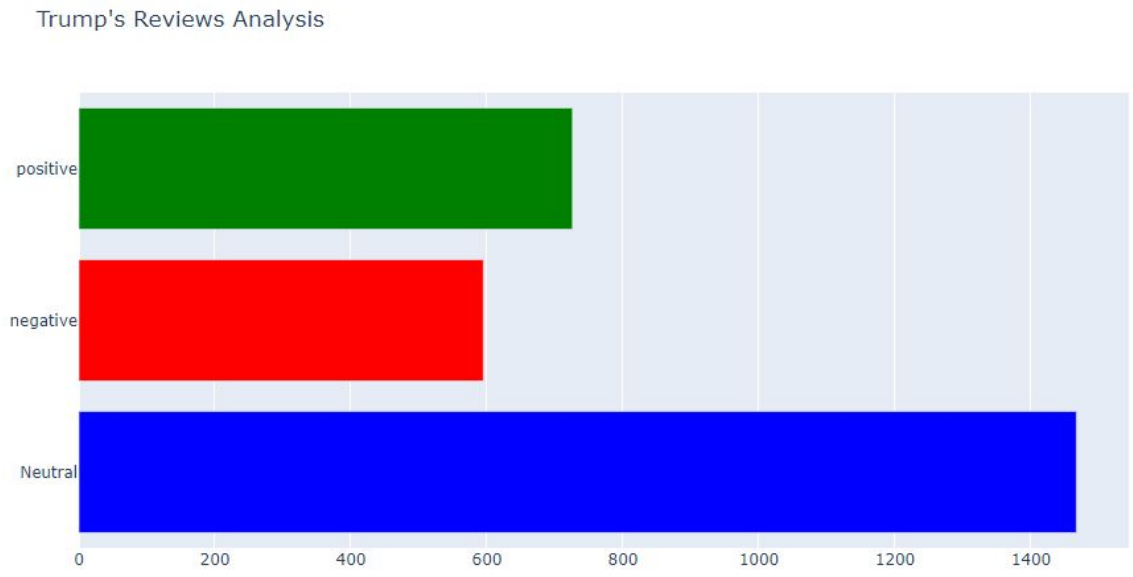


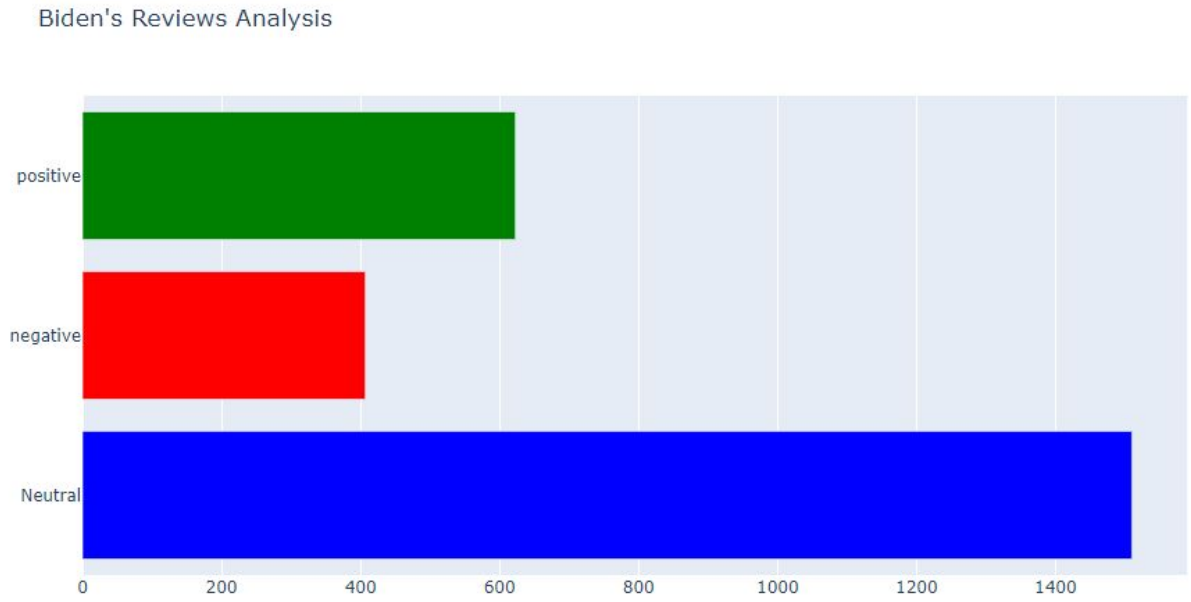
PROJECT TIMELINE



PROJECT RESULTS

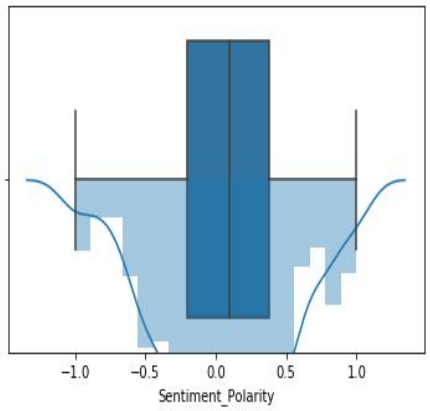
Result Screenshots:





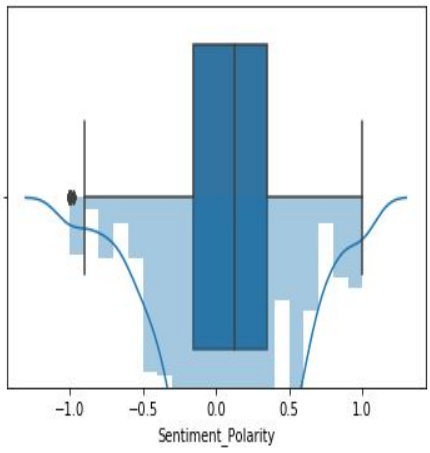
Donald Trump's boxplot

```
sns.distplot(df_subset_trump['Sentiment_Polarity'])
sns.boxplot([df_subset_trump.Sentiment_Polarity])
plt.show()
```

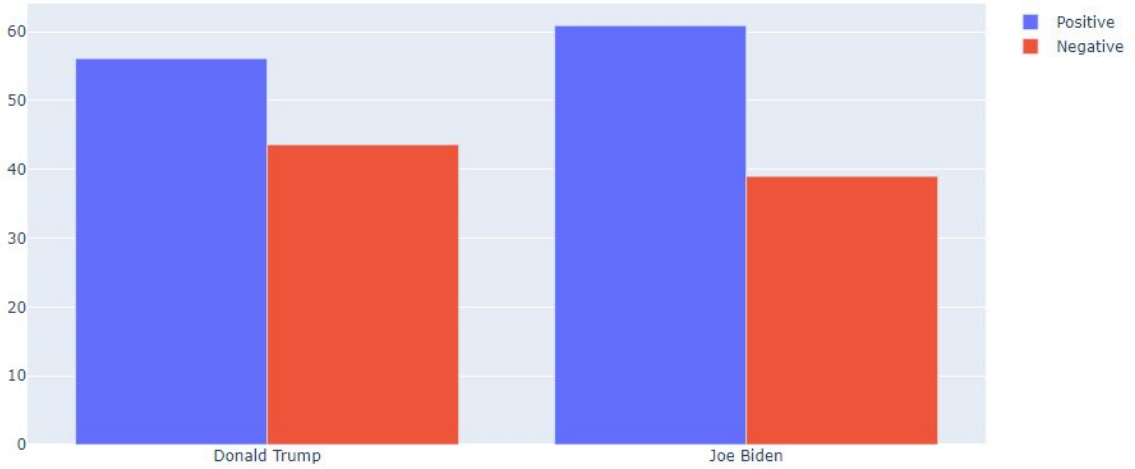


Joe Biden's boxplot

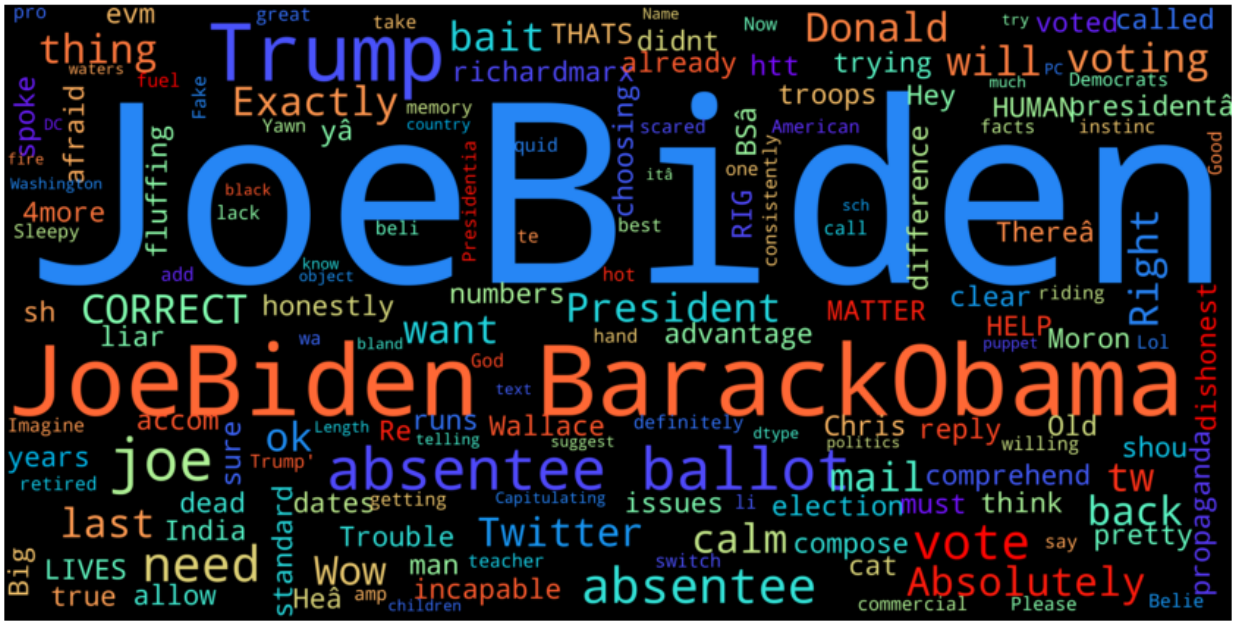
```
sns.distplot(df_subset_biden['Sentiment_Polarity'])
sns.boxplot([df_subset_biden.Sentiment_Polarity])
plt.show()
```

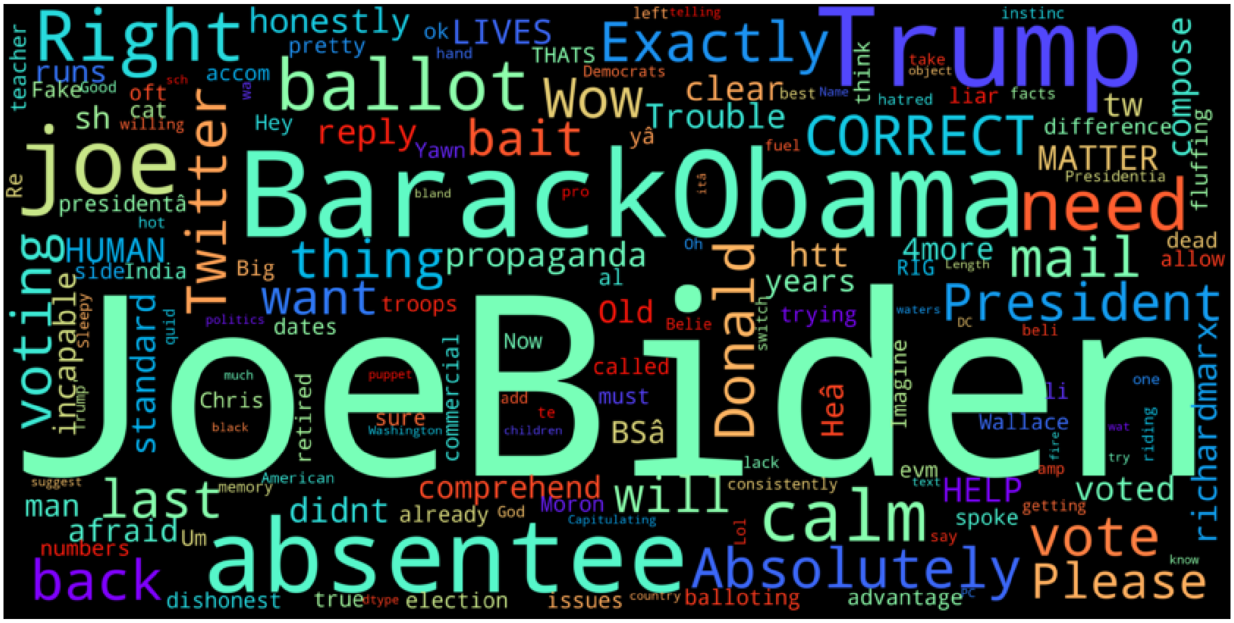


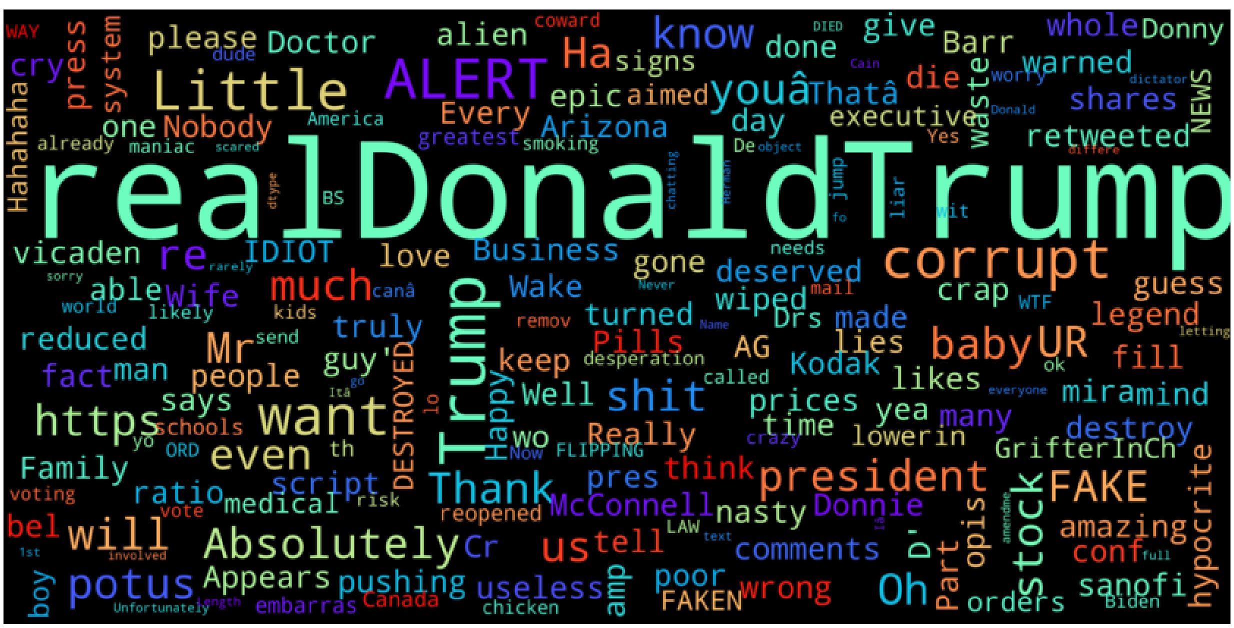
Barmode representation of positive and negative comments on both candidates:

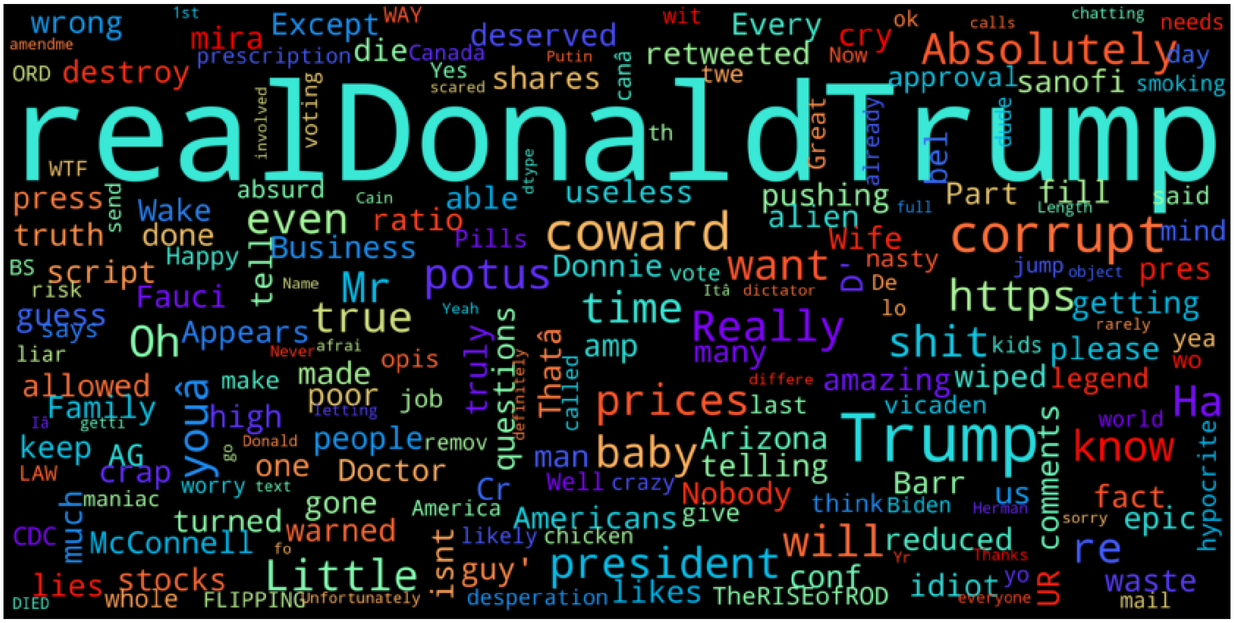


Word Clouds:

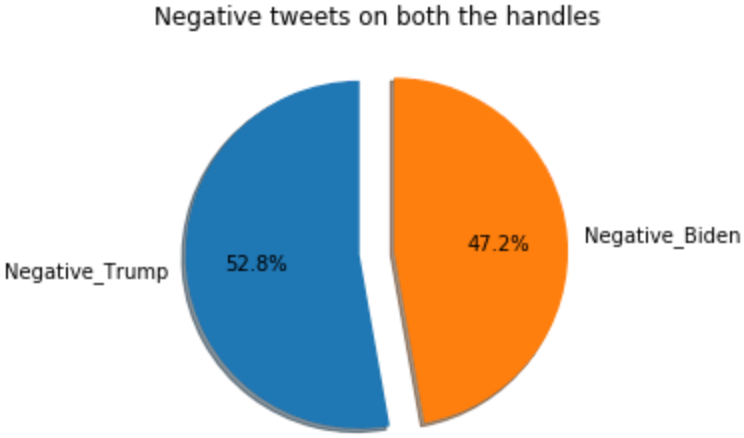


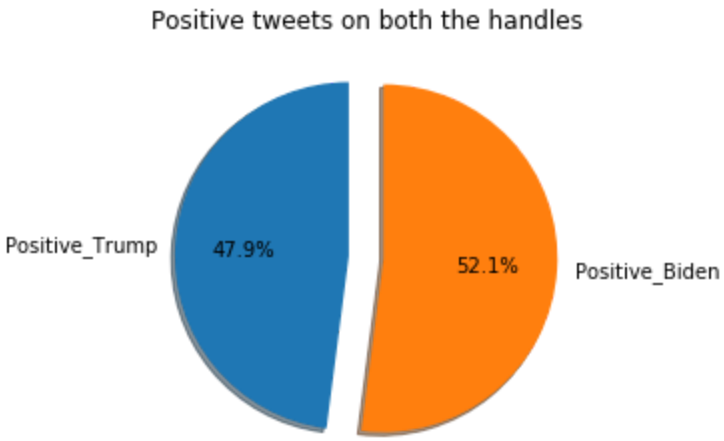

JoeBiden
BarackObama
Trump
absentee
ballot
joe
Right
Exactly
need
calm
Absolutely
Twitter
voting
Donald
President
mail
thing
want
last
back
vote
Please
will
CORRECT
bait
Wow
clear
Trouble
reply
propaganda
honestly
LIVES
compose
richardmarx
standard
incapable
didnt
comprehend
voted
HELP
HUMAN
afraid
Old
called
MATTER
difference
years
4more
htt
Wallace
imagine
retired
commercial
Chris
BSâ
Heâ
dishonest
election
balloting
advantage
issues
true
man
sh
runs
cat
Fake
teacher
pretty
accom
Hey
Yawn
THATS
think
tw
fluffing
dead
allow
troops
dates
India
side
Big
Now
must
sure
trying
sure
spoke
evm
lack
consistently
already
afraid
numbers
Um
God
Moron
getting


realDonaldTrump
Trump
Little
ALERT
corrupt
president
want
much
Absolutely
know
Thank
potus
https
even
will
Mr
baby
UR
FAKE
stock
shit
Donnie
McConnell
nasty
comments
Oh
hypocrite
amazing
sanofi
orders
wrong
FAKEN
poor
useless
pushing
Appears
people
says
Family
ratio
medical
script
reopened
tell
us
Cr
bel
boy
Canada
embarras
chicken
legend
fill
guess
miramind
destroy
GrifterInCh
likes
many
lowerin
time
prices
yea
lies
made
Drs
wiped
deserved
crap
Kodak
AG
Pills
turned
Wake
truly
keep
guy
Happy
DESTROYED
Really
think
pres
FLIPPING
Business
love
IDIOT
re
vicaden
gone
able
Wife
reduced
fact
man
alien
Doctor
please
press
system
cry
Hahahaha
Nobody
Every
Arizona
epic
aimed
youâ
Thatâ
die
Barr
warned
whole
Donny
shares
retweeted
executive
waste
NEWS
day
give
done
signs
Ha
one
America
greatest
smoking
already
maniac
Part
opis
wo
desperation
called


realDonaldTrump
Trump
Absolutely
corrupt
coward
president
Little
prices
Really
potus
https
shit
know
baby
time
true
even
want
will
Mr
Oh
re
Ha
wrong
Except
deserved
retweeted
Every
cry
destroy
shares
sanofi
approval
prescription
press
truth
useless
pushing
alien
Part
fill
said
able
Wake
done
ratio
Business
Wife
nasty
mind
script
Donnie
pres
Fauci
getting
guess
Appears
amp
many
please
allowed
amazing
wiped
legend
Family
high
people
Arizona
telling
keep
AG
crap
one
Doctor
man
Barr
us
fact
McConnell
turned
gone
warned
Americans
Nobody
think
Biden
epic
hypocrite
lies
stocks
guy
conf
reduced
idiot
waste
TheRISEofROD
likes
desperation
questions
comments
truly
isnt
poor
made
job
make
opis
last
vicaden
world
whole
FLIPPING
mail
Unfortunately

FINAL RESULTS: Pie-chart representation





From the pie-chart representation, it can be concluded that Joe Biden will win the 2020 US presidential elections with 52-53% votes.

REAL-TIME APPLICATIONS

This analysis model can be used to:

- Categorize any product feedback (ex: medicine, electronics etc)
- Compare product competitors based on reviews.
- Election results based on public opinion on social media.
- Analysis of Restaurant/ Movie reviews.

APPENDIX

Twitter data extraction:

```
Import csv

import tweepy

import ssl


# Oauth keys

consumer_key = "QN3CzI2gScYvDsrhhaL2SRbOPrC"

consumer_secret =

"AQU3NwlOqUb1aKxgy0Nk22H5k8jjj0tYJ4nlFRLFZQJCA07TLCJMm"

access_token =

"969527167221563392-35WKxHqmuLkkqfe1zqQbmSN276vZTFAbz"
```

```
access_token_secret =
"wplE6EPMtyqNRESaBV175jRzU5ffgq934nX3h2dNQ7rnzarg"


# Authentication with Twitter
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
ssl._create_default_https_context = ssl._create_unverified_context
api = tweepy.API(auth)
api = tweepy.API(auth, wait_on_rate_limit=True)


user = api.me()
print (user.name)


# update these for the tweet you want to process replies to 'name' = the
account username and you can find the tweet id within the tweet URL
name = 'realDonaldTrump'
tweet_id = ['1290967953542909952']
```

```
replies=[]
for tweet in tweepy.Cursor(api.search,q='to:'+name,
result_type='recent', timeout=999999).items(100):
   if hasattr(tweet, 'in_reply_to_status_id_str'):
      if (tweet.in_reply_to_status_id_str==tweet_id):
         replies.append(tweet)


with open('trump_data.csv', 'a+') as f:
   csv_writer = csv.DictWriter(f, fieldnames=('user', 'text'))
   csv_writer.writeheader()
   for tweet in replies:
      row = {'user': tweet.user.screen_name, 'text': tweet.text.replace('\n',
' ')}
      csv_writer.writerow(row)
```

Tweet Analysis (US Elections 2020):

```
# Install Libraries
pip install -U textblob
```

```
pip install pandas
pip install numpy
pip install plotly
pip install seaborn
pip install matplotlib
pip install wordcloud

# Import Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from textblob import TextBlob
from wordcloud import WordCloud
import plotly.graph_objects as go
import plotly.express as px

#Reading both the csv Files
Trump_reviews = pd.read_csv('/content/Trumpall2.csv', encoding = 'utf-8')
Biden_reviews = pd.read_csv('/content/Bidenall2.csv', encoding = 'utf-8')

#Visualizing Dataframes
Trump_reviews.head()
Biden_reviews.head()

#Visualizing text
```

```
Trump_reviews['text'][10]
Biden_reviews['text'][500]

# Finding sentiments using TextBlob
text_blob_object1 = TextBlob(Trump_reviews['text'][10])
print(text_blob_object1.sentiment)
text_blob_object2 = TextBlob(Biden_reviews['text'][500])
print(text_blob_object2.sentiment)

# Sentence  with zero polarity and subjectivity
text_blob_object2 = TextBlob(Biden_reviews['text'][100])
print(text_blob_object2.sentiment

# Finding Sentiment Polarity for each datasets

# Donald Trump
def find_pol(review):
    return TextBlob(review).sentiment.polarity

Trump_reviews['Sentiment_Polarity']                                    =
Trump_reviews['text'].apply(find_pol)
Trump_reviews.tail()

# Joe Biden
def find_pol(review):
    return TextBlob(review).sentiment.polarity
```

```
Biden_reviews['Sentiment_Polarity'] = Biden_reviews['text'].apply(find_pol)
Biden_reviews.tail()

# Adding one more attribute for Expression Label

# Donald Trump
Trump_reviews['Expression Label'] = np.where(Trump_reviews['Sentiment_Polarity']>0,'positive', 'negative')
Trump_reviews['Expression Label'][Trump_reviews.Sentiment_Polarity ==0] = "Neutral"
Trump_reviews.tail()

# Joe Biden
Biden_reviews['Expression Label'] = np.where(Biden_reviews['Sentiment_Polarity']>0,'positive', 'negative')
Biden_reviews['Expression Label'][Biden_reviews.Sentiment_Polarity ==0] = "Neutral"
Biden_reviews.tail()

# Analyzing Positive, Negative and Neutral replies on Trump's tweets.
new1 = Trump_reviews.groupby('Expression Label').count()
x = list(new1['Sentiment_Polarity'])
y = list(new1.index)
tuple_list = list(zip(x,y))
df = pd.DataFrame(tuple_list, columns=['x','y'])
df['color'] = 'blue'
df['color'][1] = 'red'
```

```
df['color'][2] = 'green'
fig = go.Figure(go.Bar(x=df['x'],
                y=df['y'],
                orientation ='h',
                marker={'color': df['color']}))
fig.update_layout(title_text='Trump\'s Reviews Analysis' )
fig.show()

# Analyzing Positive, Negative and Neutral replies on Biden's tweets
new2 = Biden_reviews.groupby('Expression Label').count()
x = list(new2['Sentiment_Polarity'])
y = list(new2.index)
tuple_list = list(zip(x,y))
df = pd.DataFrame(tuple_list, columns=['x','y'])
df['color'] = 'blue'
df['color'][1] = 'red'
df['color'][2] = 'green'
fig = go.Figure(go.Bar(x=df['x'],
                y=df['y'],
                orientation ='h',
                marker={'color': df['color']}))
fig.update_layout(title_text='Biden\'s Reviews Analysis' )
fig.show()

# Dropping all the statements having zero polarity

# Donald Trump
```

```
reviews1 = Trump_reviews[Trump_reviews['Sentiment_Polarity'] == 0.0000]
reviews1.shape
cond1 =
Trump_reviews['Sentiment_Polarity'].isin(reviews1['Sentiment_Polarity'])
Trump_reviews.drop(Trump_reviews[cond1].index, inplace = True)
Trump_reviews.shape

# Joe Biden
reviews2 = Biden_reviews[Biden_reviews['Sentiment_Polarity'] == 0.0000]
reviews2.shape
cond2 =
Biden_reviews['Sentiment_Polarity'].isin(reviews1['Sentiment_Polarity'])
Biden_reviews.drop(Biden_reviews[cond2].index, inplace = True)
Biden_reviews.shape

# Let's make both the datasets balanced now. So we will just take 1000
rows from both datasets and drop rest of them.

# Donald Trump
np.random.seed(10)
remove_n =324
drop_indices = np.random.choice(Trump_reviews.index, remove_n,
replace=False)
df_subset_trump = Trump_reviews.drop(drop_indices)
df_subset_trump.shape

# Joe biden
```

```
np.random.seed(10)
remove_n =31
drop_indices = np.random.choice(Biden_reviews.index, remove_n,
replace=False)
df_subset_biden = Biden_reviews.drop(drop_indices)
df_subset_biden.shape

# Data Visualiization

# Donald Trump
sns.distplot(df_subset_trump['Sentiment_Polarity'])
sns.boxplot([df_subset_trump.Sentiment_Polarity])
plt.show()

# Joe Biden
sns.distplot(df_subset_biden['Sentiment_Polarity'])
sns.boxplot([df_subset_biden.Sentiment_Polarity])
plt.show()

# Percentage count for Donald Trump
count_1 = df_subset_trump.groupby('Expression Label').count()
print(count_1)
negative_per1 = (count_1['Sentiment_Polarity'][0]/1000)*100
positive_per1 = (count_1['Sentiment_Polarity'][1]/1000)*100

# Percentage count for Joe Biden
count_2 = df_subset_biden.groupby('Expression Label').count()
```

```
print(count_2)
negative_per2 = (count_2['Sentiment_Polarity'][0]/1000)*100
positive_per2 = (count_2['Sentiment_Polarity'][1]/1000)*100


# Analysis of Positive and Negative comments on both the handle

Politicians = ['Donald Trump', 'Joe Biden']
lis_pos = [positive_per1, positive_per2]
lis_neg = [negative_per1, negative_per2]

fig = go.Figure(data=[
    go.Bar(name='Positive', x=Politicians, y=lis_pos),
    go.Bar(name='Negative', x=Politicians, y=lis_neg)
])
# Change the bar mode
fig.update_layout(barmode='group')
fig.show()

# Most Positive and Most Negative comments on both the Twitter handles

# Donald Trump
# Most positive replies
most_positive1 = df_subset_trump[df_subset_trump.Sentiment_Polarity ==
1].text.head()
pos_txt1 = list(most_positive1)
```

```
pos1 = df_subset_trump[df_subset_trump.Sentiment_Polarity == 1].Sentiment_Polarity.head()
pos_pol1 = list(pos1)
fig = go.Figure(data=[go.Table(columnorder = [1,2],
                    columnwidth = [50,400],
                     header=dict(values=['Polarity','Most Positive Replies on Trump\'s Handle'],
                    fill_color='paleturquoise',
                    align='left'),
          cells=dict(values=[pos_pol1, pos_txt1],
                    fill_color='lavender',
                    align='left'))])

fig.show()

# Most Negative Replies
most_negative1 = df_subset_trump[df_subset_trump.Sentiment_Polarity == -1].text.head()
neg_txt1 = list(most_negative1)
neg1 = df_subset_trump[df_subset_trump.Sentiment_Polarity == -1].Sentiment_Polarity.head()
neg_pol1 = list(neg1)
fig = go.Figure(data=[go.Table(columnorder = [1,2],
                    columnwidth = [50,400],
                       header=dict(values=['Polarity','Most Negative Replies on Trump\'s handle'],
                    fill_color='paleturquoise',
```

```
                    align='left'),
            cells=dict(values=[neg_pol1, neg_txt1],
                    fill_color='lavender',
                    align='left'))])

fig.show()

# Joe Biden
# Most Positive replies
most_positive2 = df_subset_biden[df_subset_biden.Sentiment_Polarity == 1].text.tail()
pos_txt2 = list(most_positive2)
pos2 = df_subset_biden[df_subset_biden.Sentiment_Polarity == 1].Sentiment_Polarity.tail()
pos_pol2 = list(pos2)
fig = go.Figure(data=[go.Table(columnorder = [1,2],
                    columnwidth = [50,400],
                     header=dict(values=['Polarity','Most Positive Replies on Biden\'s handle'],
                    fill_color='paleturquoise',
                    align='left'),
            cells=dict(values=[pos_pol2, pos_txt2],
                    fill_color='lavender',
                    align='left'))])

fig.show()
```

```
# Most negative replies
most_negative2 = df_subset_biden[df_subset_biden.Sentiment_Polarity ==
-1].text.head()
neg_txt2 = list(most_negative2)
neg2 = df_subset_biden[df_subset_biden.Sentiment_Polarity ==
-1].Sentiment_Polarity.head()
neg_pol2 = list(neg2)
fig = go.Figure(data=[go.Table(columnorder = [1,2],
                    columnwidth = [50,400],
                        header=dict(values=['Polarity','Most Negative Replies
on Biden\'s handle'],
                    fill_color='paleturquoise',
                    align='left'),
            cells=dict(values=[neg_pol2, neg_txt2],
                    fill_color='lavender',
                    align='left'))])

fig.show()

# WordCloud for Donald Trump
# Start with one review:
text = str(df_subset_biden.text)
# Create and generate a word cloud image:
wordcloud = WordCloud(max_font_size=100, max_words=500, scale=10,
relative_scaling=.6, background_color="black", colormap =
"rainbow").generate(text)
# Display the generated image:
```

```
plt.figure(figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()

# WordCloud for Joe Biden
# Start with one review:
text = str(Biden_reviews.text)
# Create and generate a word cloud image:
wordcloud = WordCloud(max_font_size=100,
max_words=500,scale=10,relative_scaling=.6,background_color="black",
colormap = "rainbow").generate(text)
# Display the generated image:
plt.figure(figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()

# Comparison between negative comments on both
labels =  ['Negative_Trump', 'Negative_Biden']
sizes = lis_neg
explode = (0.1, 0.1)
fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels = labels, autopct = '%1.1f%%',
shadow = True, startangle=90)
ax1.set_title('Negative tweets on both the handles')
plt.show()
```

```
# Comparison between Positive comments on both
labels =  ['Positive_Trump', 'Positive_Biden']
sizes = lis_pos
explode = (0.1, 0.1)
fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels = labels, autopct = '%1.1f%%',
shadow = True, startangle=90)
ax1.set_title('Positive tweets on both the handles')
plt.show()
```

REFERENCES

- https://medium.com/analytics-vidhya/tweet-analytics-using-nlp-f83b9f7f7349
- https://ieeexplore.ieee.org/abstract/document/7219856/
- https://ieeexplore.ieee.org/abstract/document/6113240/
- https://ieeexplore.ieee.org/abstract/document/8397589/
- https://www.aclweb.org/anthology/S16-1040.pdf
- https://towardsdatascience.com/simple-twitter-analytics-with-twitter-nlp-toolkit-7d7d79bf2535
- Topic Sentiment Trend Detection and Prediction for Social Media by Aashish Thota
- https://www.youtube.com/watch?v=gUFDtuz73gI