Project Report For

SOCIAL MEDIA & WEB ANALYTICS

on

RULING POLITICAL PARTY SENTIMENTAL ANALYSIS AND HATE SPEECH DETECTION FOR TWITTER

Submitted By:

Jiya Midha	**401703010**
Manan Vyas	**401703014**
Shourya Marwaha	**401708013**
Sumit Arora	**401708015**

Submitted To:

Dr. Nitin Arvind Shelke

L.M. THAPAR SCHOOL OF MANAGEMENT

THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY, PATIALA

Session: 2021-22

DECLARATION

I/We hereby declare that the work which is being presented in the report entitled “Ruling Political Party Sentimental Analysis and Hate speech detection for Twitter” under the subject Social Media and Web Analytics, is an authentic record of our own work carried out during the period from JUNE, 2021 to December, 2021 at Department of LMTSM.

The matters and the results presented in this report has not been submitted by us for the award of any other degree elsewhere.

Digital Signature of Candidate

Jiya Midha
[401703010]

Manan Vyas
[401703014]

Shourya Marwaha
[401708013]

Sumit Arora
[401708015]

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

With the advancement in technology, social media is now much more than just a medium to connect across globe. It enables people to share their videos, photos, information to and much more with such ease. Also, social media has changed the way business use to operate the things, it provides new opportunities for the business to market their products, to understand their customer and to make communication much easier than before. Also, it enables government and political parties to analyze public opinion based on their decisions or policies. One of the biggest and perfect social media platforms for the government to collect these types of data is twitter. Twitter being 3rd largest social media platform enjoys more than 330 million users base and enables big political to understand whether their policies are taken positively by the public or it resulted in negative expression. Thus, aim of this project is to analyze tweets of common man and classify them whether they are positive, negative or neutral. This project uses sentimental analyses using Vader which is available in NLTK in python to better understand emotions of the text.

INTRODUCTION

In this fast-paced life where people were struggling to maintain their relationships with their friends, long distant family members and many others, social media was introduced as a medium to connect with everyone across the globe. First version of social media was introduced in 1996 in the form of social networking (chatrooms, horoscope etc.) and video website called Bolt (1996-2007). Then it was followed by a site called six degrees, which was the first to introduce feature of connecting with users as friends, based on social-circles network model which is now used by many popular companies like Facebook, LinkedIn etc. In 1999 first popular blogging platform was introduced, which was named as LiveJournal. Then, in 2003 LinkedIn was launched, which was followed by one of the most popular social media platforms, Facebook in 2004 and Twitter in 2006, which later changed the world.

Now, in last few years social media has become the integral part of our life and it's not just limited to connecting with friends but it enables people to share their opinions, pictures, news, blogs and many other things. According to research, out of 7 billion people in this world 3.5 billion people are very active on social media platforms and average person spends about minimum 142 minutes per day and 36.5 days in a year just on social media. Further situations like covid 19 boost social media usage and became integral part of everyone life. In today's era social media is not just limited for personal use but it has opened many opportunities for business as well as for government. Many businesses are using social media platforms for marketing purpose, to understand their customers, their needs and wants to make their product or service more customer friendly and to hire new employees for their firms. Also, government or political parties are using social media platforms to advertise themselves and to aware people about their new policies. Also, social media platform is one of the best methods for the government to understand and analyze the opinion of the public on their policies and leaders. It helps them to get broader view about the actions taken by them like whether those actions are liked or disliked by the common man.

Although government also uses various other methods to analyze public opinion like surveys, debates on news channels but those methods are not as reliable as social media platform, because those methods contain opinion of only few experts which can be biased in nature, whereas social media platform contains opinion of common man and provides extra advantage of huge database and is unbiased in nature. One of the popular media for the government to understand the opinion of public is twitter.

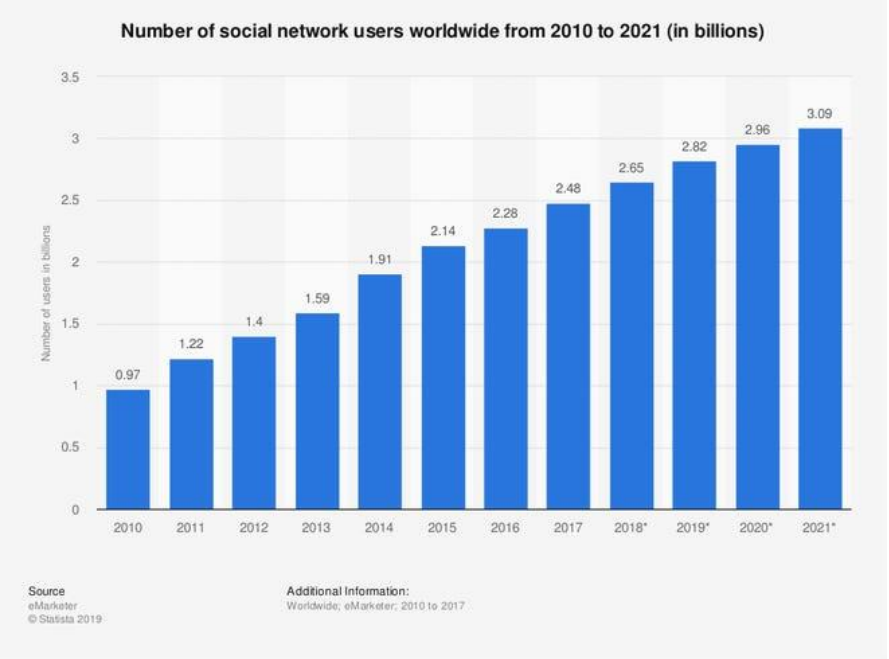


Figure 1: Number of social network users worldwide from 2010 to 2021

Twitter, an American microblogging and social networking service which enable its users to post or interact with messages which are called as "tweets", is a platform of more than 330 million active users in the world. 500 million people tweets every day and 6000 tweets every second on this platform and as twitter allow people to read tweets or post without login, thus about 500 million people reads tweets without login every month. According to a research, most common subject of tweets is politics, especially during election period it became one of the hottest topics of discussion on twitter. Users use twitter as medium to express their opinions and to discuss about various government policies and leaders. Twitter is basically now a modern way of freedom of speech specially when it comes to politics. It is stated that in year 2021, 26% of tweet's subjects were only related to politics. Thus, due to its huge popularity and database, it became a perfect medium for the government or political parties to analyze tweets to get better insights about their policies and popularity of their leader. Thus, this project aims to differentiate between positive and negative tweets to help political parties to understand the public opinion about their policies or their leader. Further this project can help to classify that to which party particular tweets belongs. In order to differentiate between the positive and negative tweets this project uses concept of sentimental analysis on database which consist of tweets from twitter. Sentimental analysis is basically a NLP (Natural language processing) technique which helps in contextual mining of text, it helps to identify the emotion with which user has written the text, like whether the text is positive, negative or neutral. It can be done by using various machine learning techniques, statistical models. It can also be implemented at various levels like if we want to understand context of whole document,

phrase or just a sentence. It can be also used in various other scenarios like to get review of any product or movie, to review financial report of the companies or can also be used for marketing purpose. Further, this project uses VADER, which is available in NLTK library to perform the analysis.

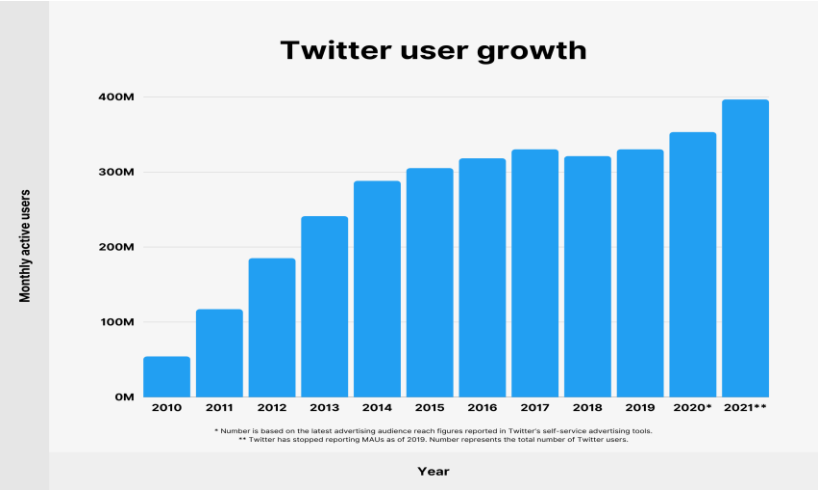


Figure 2 Number of active users on twitter

1.1. Problem Statement

During election, twitter became medium for the common man to express his views about the government or other political party. At the same time political parties use these tweets to understand their public image. But many times, twitter also becomes medium for some people to spread hate speech and rumors to hinder the image of their opposition party. Thus, there is need for political parties to differentiate between positive and negative tweets to predict future results and area of concerns which can help them to improve their future policies.

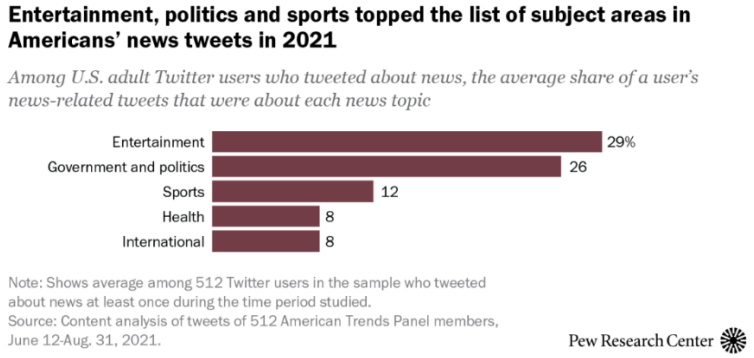


Figure 3 Popular tweeted topics in US in year 2021

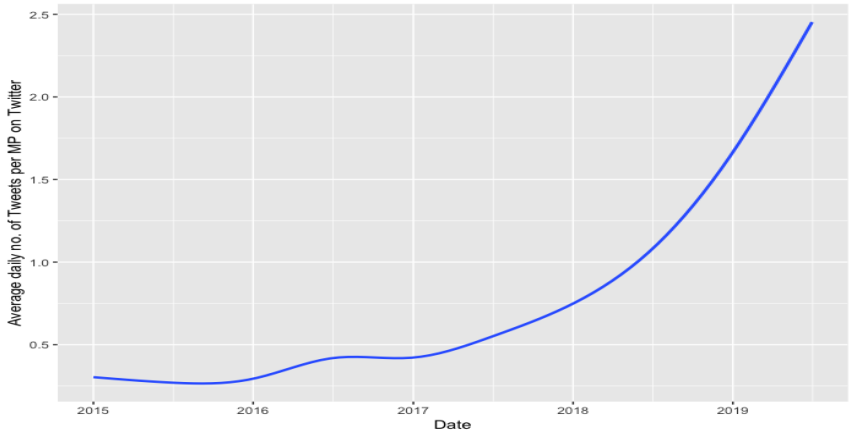


Figure 4 Daily number of tweets per MP

BACKGROUND RESEARCH

In 2010 Garcia was one of the first researcher who used analysis to understand consumer behavior based on social media platforms. In his research he stated that 67% of consumers who followed brand on twitter buy product from that brand and 51% are likely to buy from the one they followed on Facebook. Turney in 2002 explains unsupervised machine learning algorithm for classifying whether reviews are justified or not. His classification was done on the basis of average semantic orientation of phrases that contain adjectives and adverbs in already given reviews. Pang lee and Vaithyanathan in 2002 were the first one to work on sentimental analysis. They tried to classify text not just by topic but by inclusive sentiment. They used machine learning algorithm and proved that these algorithm outsmarts human algorithms. They also used other machine learning algorithm like Naïve Bayes, support vector machines etc. They also showed that supervised machine learning algorithms are the basis of sentimental analysis. Also, E loper, S Bird developed NLTK library which consist of various program modules, set of organized files, tutorials, problem sets, built in classifiers and linguistic program which enables NLTK to provide more structured programs and better results. Pak and Paroubek in 2010, used tweets with emoticons as training as training corpus, he divided them in two categories where in one category happy emoticons represent positive set whereas sad emoticon represents negative set. Further, H wang, D Canm F bar, S Narayana proposed a system which helped to perform real time analysis of public reactions in 2012 president elections in US. They used twitter as their medium to obtain reactions and analyzed sentiment on various public events. They performed their analysis on data like election, candidates, promotions etc. At the end of their research, they stated sentimental analysis to be very quicker than traditional content analysis. Then O. Almatrafi, S. Park, B.Chavan, introduced a system which aims to perform sentimental analysis based on location. They also used twitter as their medium for data and extract location from the tweets through script. They collected about 6 lakh tweets of two political for period of 7 days. Further they used supervised machine learning approach, like Naïve Bayes algorithm to classify tweets as positive or negative. They also recognized the views and opinion of common man about these two political Party according to different location. L. Jiang, M Yu, M Zhou, builds a system that helps in target dependent classification. In 2014, Isah, trundle & Neagu, used lexicon based and machine learning method to analyze product safety using text mining and sentimental analysis. Further, Martin-Domingo & Mandsberg in 2019, used machine learning to analyze sentimental analysis for airport service quality with the help of data obtained from twitter.

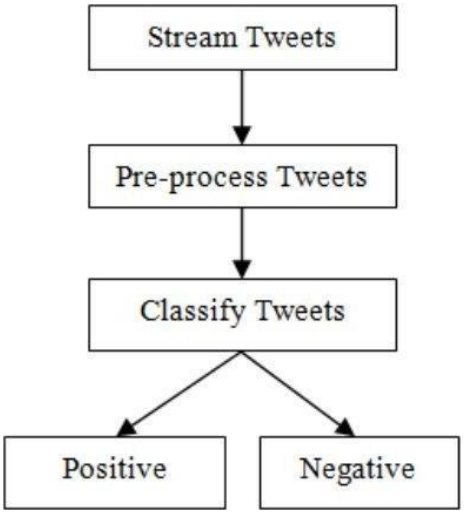
1.2. Goals and Objectives

Main aim of this project is to select some tweets related to politics from twitter database and to differentiate between positive, negative and neutral tweets. This project aims to identify the manner in which users have expressed their opinions about their government or about political parties. It aims to provide better understanding about the public opinions with regard to any new policy or about any particular decision of political parties. Although currently political parties use various different techniques like surveys, debates on televisions etc to understand the view of the public but these methods are not that precise due to biasness and small database. Thus, this project solves these problems as well, since it uses twitter as a source of database to extract opinions of people. So, huge database enables better analysis and includes opinion of common man which is not biased, thus resulted in better accurate analysis. Also, as twitter allows people to express their thought within 280 characters about any topic in natural language. Thus, it also became easy to use this data for analysis in python using NLP. Further, analyzing such huge database is not easy but with the use of NLTK library we can classify our data which can further help in improving the accuracy of the results based on different classifiers. In this project we aim to perform our analysis using Vader (Valence Aware dictionary for sentimental reasoning) which is available in NLTK package and is used for text sentiment analysis based on polarity and intensity of emotion.

PROPOSED METHODLOGY

Data collecting is not an easy process, despite appearances. For data collection, a number of decisions must be made. We keep a dataset for training, testing, and sentiment analysis on Twitter for our thesis. We'll look at how data is acquired, stored, processed, and categorized in this chapter. Let's talk about our recommended architecture before we go into these processes and distinct datasets.

As our goal is to achieve sentiment analysis for data provided from Twitter.



- **Step-1** First, we'll use the Tweepy Python package to stream tweets into our build classifier.
- **Step-2** The tweets are then pre-processed to make them suitable for mining and feature extraction.
- **Step-3** We run this data via VADER sentimental analysis after pre-processing, and based on learned findings, we identify it as positive, negative, or neutral.

Since Twitter is our data source, we'll be analyzing it. The tweets from Twitter will be streamed into our database. We'll utilize the Twitter API for this purpose.

3.1 Proposed Architecture

Fig. 5. represents the flow of data through various modules used in the proposed methodology.

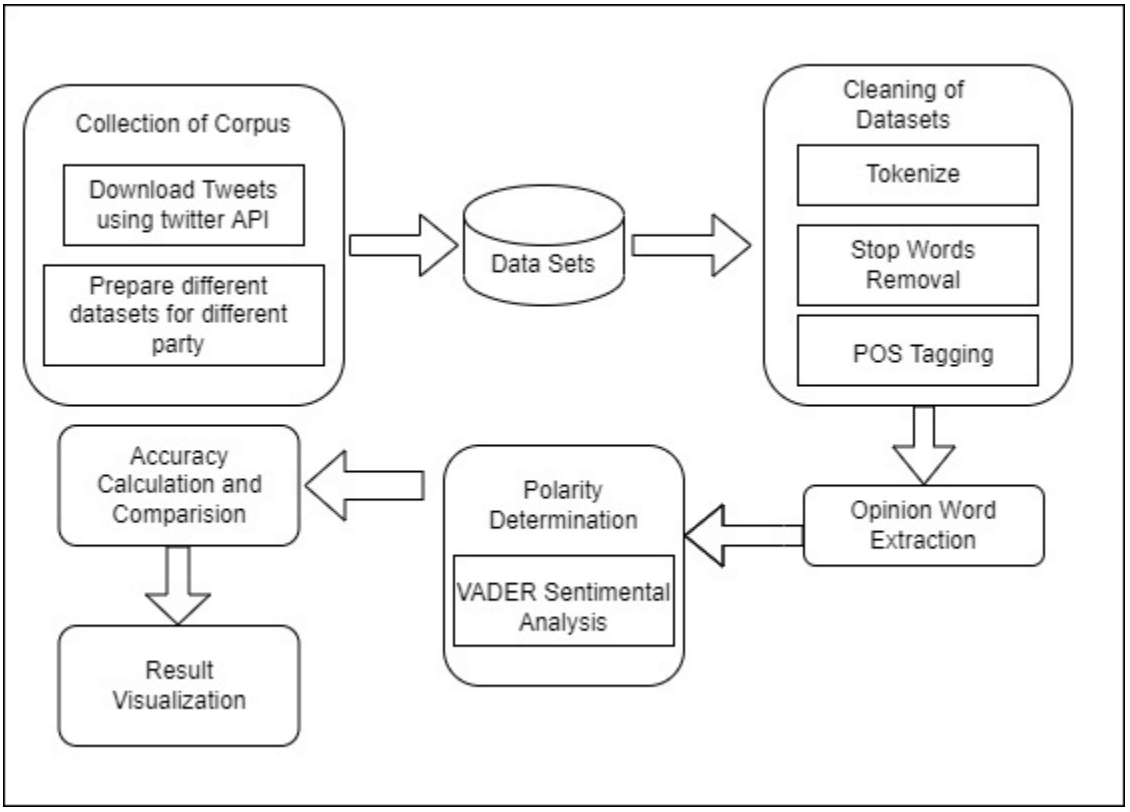


Figure 5 Flow Chart for the proposed methodology

3.2 Twitter API (Application Programming Interface)

With the use of the Twitter API, users may gather tweets. REST API and Streaming API are the two types of APIs offered by Twitter. The following are the distinctions: REST APIs provide for short-term connections and only a limited amount of data to be gathered at a time, whereas Streaming APIs allow for real-time tweets and long-term connections. For our study, we use the Streaming API. We need a long-lasting connection with no data rate constraint to gather a huge number of tweets.

3.3 Data Collection

3.3.1 Twitter Data

We must first have a Twitter account in order to access the Twitter API. It is simple to create simply filling out the sign-up information on the twitter.com website. Following that, you will be given a username and password to use for login purposes. You may now read and send tweets on any topic you desire after your account has been setup.

Twitter provides a platform through which we may access and utilize data from our Twitter accounts for our own purposes. To do so, go to dev.twitter.com and log in with our Twitter credentials. We begin by creating an application that will be used to stream tweets by supplying the relevant information on this page. We can obtain customer key, customer secret key, access token key, and access secret key after our API has been setup. When a user wants to access Twitter data, these keys are used to authenticate the user.

Only tweets regarding political parties should be gathered since the goal of this thesis is to assess the emotion of tweets posed for political parties. As a result, we develop a Python script that will be used to get tweets from Twitter. Before we begin writing this script, we must first install the Tweepy Python package. Python is a very sophisticated language that can do a wide range of tasks thanks to a large number of Python packages. Tweepy is an open-source Python module that allows Python to interface with Twitter and collect data via its API for use in our programs. To install Tweepy, just type 'pip install Tweepy in a command line or bash shell, and we're ready to run our script.

```
In [2]: pip install tweepy

Requirement already satisfied: tweepy in c:\users\admin\annaconda\lib\site-
packages (4.4.0)
Requirement already satisfied: requests-oauthlib<2,>=1.0.0 in c:\users\admi
n\annaconda\lib\site-packages (from tweepy) (1.3.0)
Requirement already satisfied: requests<3,>=2.11.1 in c:\users\admin\annaco
nda\lib\site-packages (from tweepy) (2.24.0)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\admin\annaconda
\lib\site-packages (from requests-oauthlib<2,>=1.0.0->tweepy) (3.1.1)
Requirement already satisfied: idna<3,>=2.5 in c:\users\admin\annaconda\lib
\site-packages (from requests<3,>=2.11.1->tweepy) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\admin\annacon
da\lib\site-packages (from requests<3,>=2.11.1->tweepy) (2020.6.20)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\admin\annacond
a\lib\site-packages (from requests<3,>=2.11.1->tweepy) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\admin\annaconda\lib\site-packages (from requests<3,>=2.11.1->tweep
y) (1.25.11)
Note: you may need to restart the kernel to use updated packages.

In [3]: pip install preprocessor

Requirement already satisfied: preprocessor in c:\users\admin\annaconda\lib
\site-packages (1.1.3)
Note: you may need to restart the kernel to use updated packages.
```

Figure 6 Install Tweepy and Preprocessor.

We utilize all of the keys and secrets obtained from the API in this script. First, we construct a listener class, which is used to import data from Twitter. To collect data, we must first set up the 'OAuth' protocol. OAuth is an authorization mechanism that is widely used. It allows users to check in to third-party websites using their social media accounts without revealing their passwords. OAuth offers users with security and authorization.

```
In [5]: search_words = "biden, @joebiden, bidenharris, @kamalaharris, election, vote, #HunterBiden, #JoeBidenMustStepDown, trump, DemocraticParty"
        new_search = search_words + " -filter:retweets"

In [6]: for tweet in tweepy.Cursor(api.search_tweets,q=new_search,count=10,lang="en",since_id=0).items():
            csvWriter.writerow([tweet.created_at, tweet.text.encode('utf-8'),tweet.user.screen_name.encode('utf-8'), tweet.user.location.encode('utf-8')])
```

Figure 7 Code for collecting tweets using Tweepy

In this script, we must provide all of the keys provided by the Twitter API. To obtain the tweets on a certain topic, we use the Tweepy 'Stream' module. This is where we supply the authorization information as well as the class that will be used to import tweets. We also use a filter in the stream to assist us find tweets relevant to a specific topic by entering a term connected to that topic in the filter. When we execute our script, we'll observe that tweets have been imported from Twitter, which we may use for our purposes.

3.3.2 Training Data

Training data is another type of data that we gathered for this thesis. This information will be utilized to train the classifier that we will create. We utilize Python's NLTK module to collect this information. NLTK is made up of corpora, which are enormous collections of structured text files that are used for analysis. There are numerous sorts of text files in these corpora, such as quotations, reviews, discussion, history, and so on. For our training purposes, we shall choose files from these corpora.

In [1]:
```
# numerical computation
import numpy as np

# data processing/manipulation
import pandas as pd
pd.options.mode.chained_assignment = None
import re

# data visualization
import matplotlib.pyplot as plt
from os import path
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import plotly.express as px
```

In [2]:
```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.probability import FreqDist

# spell correction, lemmatization
from textblob import TextBlob
from textblob import Word

# sklearn
from sklearn.model_selection import train_test_split
```

Figure 8 Importing Basic Libraries.

Table 1 Sample in NLTK Corpus

Reviews	CLASS
foolish, idiotic and boring it's so lad dish and youngish , only teenagers could find it funny	NEGATIVE
the rock is destined to be the 21st century's new Conan and that he's going to make a splash even greater than Arnold Schwarzenegger	POSITIVE
Barry Sonnenfeld owes frank the pug big time the biggest problem with roger Avary's uproar against the map	NEGATIVE
the seaside splendor and shallow, beautiful people are nice to look at while you wait for the story to get going	POSITIVE

3.4 Data Storage

Once we've started obtaining data from the Twitter API, we'll need to store it so we can utilize it for sentiment analysis. We ran our scripts for a month and gathered tweets from various political parties. A.csv (comma separated values) file containing tweets collected from the Twitter API is produced each time the script indicated in figure is run. Because our gathered data files contain many fields, we employ the.csv format. Each field in CSV is separated by a comma, which makes it much easier to access the text-based field. In comparison to other file types, CSV files have a quicker read/write time.

```
In [3]: # Loading each dataset
        biden_df = pd.read_csv('hashtag_joebiden.csv', lineterminator='\n')

In [4]: #first 5 rows of biden_df
        biden_df.head()
```

Out[4]:

	created_at	tweet_id	tweet	likes	retweet_count	source	u
0	2020-10-15 00:00:01	1.316529e+18	#Elecciones2020 | En #Florida: #JoeBiden dice ...	0.0	0.0	TweetDeck	3.60666
1	2020-10-15 00:00:18	1.316529e+18	#HunterBiden #HunterBidenEmails #JoeBiden #Joe...	0.0	0.0	Twitter for iPad	8.09904

Figure 9 Reading Data Set.

We create distinct directories for each month's tweets from various politicians. We save them to our hard disc, where they can be quickly entered into our software and analyzed further. We must pre-process the data before applying it to the classifier once we have saved our tweet since the data, we obtain from API is not suitable for mining. As a result, we'll proceed to pre-processing the data.

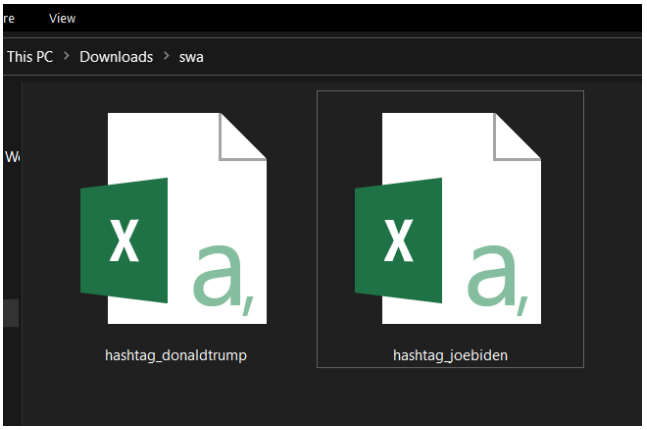


Figure 10 Database of collected tweets

3.5 Data Pre-Processing

Twitter data is not suitable for extracting features. Messages are usually accompanied with usernames, empty spaces, special characters, stop words, emojis, abbreviations, hash tags, time stamps, URLs, and other elements. As a result, we pre-process this data using NLTK's numerous functions in order to make it suitable for mining. We extract our primary message from the tweet first, then delete any empty spaces, stop words (such as is, a, the, he, them, and so on), hash tags, repetitive phrases, URLs, and so on. All emoticons and abbreviations are then substituted with their relevant meanings, such as:-), =D, =), LOL, Rolf, and so on. When we're done, we'll hand over the processed tweet to the classifier, who will use it to provide the desired results. Table 2 shows an example of a processed tweet.

Table 2 Sample Tweet and Processed Tweet

Tweet Type	Result
Original tweet	@xyz I think Kejriwal is a habitual liar, even where he don’t needs to lie he tells a lie >☹#AAP
Processed tweet	think, habit, lie, even, don’t, need, tell, angry

In [10]:
```
# Remove unneeded columns
biden_df = biden_df.drop(columns=['tweet_id','user_id','user_name','user_screen_name','user_description','user_join_date','collected_at'])

# Renaming columns
biden_df = biden_df.rename(columns={"likes": "Likes", "retweet_count": "Retweets", "state": "State", "user_followers_count": "Followers"})

# Update United States country name for consistency
d = {"United States of America":"United States"}
biden_df['country'].replace(d, inplace=True)
biden_df = biden_df.loc[biden_df['country'] == "United States"]

# Drop null rows
biden_df = biden_df.dropna()
```

Figure 11 Removing Unnecessary columns.

Because tweets contain various grammatical elements that may not be beneficial for analysis, cleaning Twitter data is required. The pre-processing is done in such a way that the data is only represented in terms of words that are easy to categorize.

We write Python code to define a function that will be used to get processed tweets.

In [11]:
```
#Preprocessing Tweets
to_remove = r'\d+|http?\S+|[^A-Za-z0-9]+'
stop_words = set(stopwords.words('english'))
ps = PorterStemmer()

# Function to preprocess tweet
def clean_tweet(tweet, stem=False, lemmatize=False):

    # Make all text lowercase
    tweet = tweet.lower()

    # Remove links, special characters, punctuation, numbers, etc.
    tweet = re.sub(to_remove, ' ', tweet)

    filtered_tweet = []
    words = word_tokenize(tweet)

    # Remove stopwords and stem
    for word in words:
        if not word in stop_words:
            if stem:
                filtered_tweet.append(ps.stem(word))
            elif lemmatize:
                filtered_tweet.append(Word(word).lemmatize())
            else:
                filtered_tweet.append(word)

    return filtered_tweet
```

Figure 12 Preprocessing Dataset

Table 3 shows the various types of contents that are included in tweets and also the actions performed on these contents. Some of the examples of clean tweets is shown in Table 4

Table 3 Removed and modified content

CONTENT	**ACTION**
Punctuation (! ? , . ” : ;)	Removed
#word	Removed #word
@any_user	Remove @any_user or replaced with “AT_USER” and then added in stop words.
Uppercase characters	Lowercase all content
URLs and web links	Remove URLs or replaced with “URL” and then added in stop words
Number	Removed
Word not starting with alphabets	Removed
All Word	Stemmed all word (Converted into simple form)
Stop words	Removed
Emoticons	Replaced with respective meaning
White spaces	Removed

Table 4 Sample cleaned data

Raw data	**Clean data**
@jackstenhouse69 I really liked it, in my opinion it def is :)	Really, liked, opinion, def
:(\u201c@EW: How awful. Police: Driver kills 2, injures 23 at #SXSW http://t.co/8GmFiOuZbS\u201d	Sad, awful, police, driver, kills, injures

```
In [13]: #5 biden tweets after filtering
         biden_df.tweet.head()

Out[13]: 6     [nypost, censorship, censored, twitter, manipu...
         17    [comments, democrats, understand, ruthless, ch...
         22    [twitter, everything, help, democrats, win, el...
         25    [realjameswoods, bidencrimefamily, joebiden, h...
         29    [come, abc, please, right, thing, move, biden,...
         Name: tweet, dtype: object
```

Figure 13 Cleaned Dataset.

Once our data is cleaned and ready for processing our next step is to classify this cleaned data into different classes. For this we have to use supervise machine learning classifiers.

3.6 SENTIMENT ANALYSIS USING VADER

Sentiment analysis is a text analysis technique that finds polarity (e.g., a positive or negative view) in a text, whether it's an entire document, a paragraph, a phrase, or a clause. Sentiment analysis tries to quantify a speaker's or writer's attitude, sentiments, assessments, attitudes, and emotions using a computational approach of subjectivity in a text.

Why is it so difficult to do a sentiment analysis?

Sentiment Analysis is a difficult subject, even if it appears simple on paper. A sentence can include numerous emotions at the same time.

3.3.1 VADER

VADER (Valence Aware Dictionary for Sentiment Reasoning) is a text sentiment analysis model that takes into account both the polarity (positive/negative) and the intensity (strong) of emotion. It's included in the NLTK package and may be used on unlabeled text data right away.

The sentimental analysis of VADER is based on a lexicon that maps lexical elements to emotion intensities, which are referred to as sentiment scores. A text's sentiment score may be calculated by adding the intensity of each word in the text.

Words like 'love,' 'enjoy,' 'glad,' and 'like,' for example, all express a pleasant attitude. VADER is also smart enough to recognize the underlying meaning of certain phrases, such as "did not love" as a negative remark. It also recognizes the importance of capitalization and punctuation, as as in the phrase "ENJOY."

3.3.2 Polarity classification

We won't try to figure out if a sentence is objective or subjective, true or false. Rather, we're simply interested in whether the text represents a good, negative, or neutral viewpoint.

3.3.3 Document-level scope

We'll also attempt to compile all of the sentences in a document or paragraph into a single viewpoint.

3.3.4 Coarse analysis

We will not attempt a fine-grained analysis to establish the degree of positive or negative. That is, we aren't attempting to predict how many stars a reviewer gave a product, only if the review was good or negative.

Basic Steps:

It's possible that a paradigm based on paragraph-length reviews may be ineffective.

Next, choose the sort of analysis. Some primitive sentiment analysis algorithms go a step farther and consider bigrams, which are two-word combinations. We'll concentrate on entire sentences, and we'll use a VADER-trained NLTK lexicon for that.

Use a number of datasets for this model, such as Amazon reviews, movie reviews, or any other product reviews.

The Sentiment Intensity Analyzer () from VADER accepts a text and produces a dictionary of scores in four categories:

• negative

• neutral

• positive

• compound (calculated by averaging the aforementioned scores)

In [14]:
```
# Sentiment Analysis
def sentiment_analysis(df):

    # Determine polarity and subjectivity
    df['Polarity'] = df['tweet'].apply(lambda x: TextBlob(' '.join(x)).sentiment.polarity)
    df['Subjectivity'] = df['tweet'].apply(lambda x: TextBlob(' '.join(x)).sentiment.subjectivity)

    # Classify overall sentiment
    df.loc[df.Polarity > 0,'Sentiment'] = 'positive'
    df.loc[df.Polarity == 0,'Sentiment'] = 'neutral'
    df.loc[df.Polarity < 0,'Sentiment'] = 'negative'

    return df[['tweet','Polarity','Subjectivity','Sentiment']].head()
```

Figure 14 Code for Sentiment Analysis

In [15]:
```
sentiment_analysis(biden_df)
```

Out[15]:

	tweet	Polarity	Subjectivity	Sentiment
6	[nypost, censorship, censored, twitter, manipu...	-0.148810	0.678571	negative
17	[comments, democrats, understand, ruthless, ch...	-1.000000	1.000000	negative
22	[twitter, everything, help, democrats, win, el...	0.175000	0.522222	positive
25	[realjameswoods, bidencrimefamily, joebiden, h...	0.000000	0.000000	neutral
29	[come, abc, please, right, thing, move, biden,...	0.078571	0.178571	positive

Figure 15 Output of Sentiment Analysis into positive, negative & neutral

RESULTS & CONCLUSION

After the sentimental analysis of the data set of tweets collected above, following inferences were obtained which are as follows:-

1. The number of positive tweets towards the ruling party were 37.5 percent, the neutral tweets were 47.3 percent and 15.2 percent. Overall maximum number of tweets are neutral and number of positive tweets are more than negative tweets.

```
# Number of Tweets by Sentiment
# Overall sentiment breakdown - Biden
print("Biden Tweet Sentiment Breakdown")

biden_positive = len(biden_df.loc[biden_df.Sentiment=='positive'])
biden_neutral = len(biden_df.loc[biden_df.Sentiment=='neutral'])
biden_negative = len(biden_df.loc[biden_df.Sentiment=='negative'])

print("Number of Positive Tweets: ", biden_positive)
print("Number of Neutral Tweets: ", biden_neutral)
print("Number of Negative Tweets: ", biden_negative)
```

```
Biden Tweet Sentiment Breakdown
Number of Positive Tweets:  33984
Number of Neutral Tweets:  42853
Number of Negative Tweets:  13802
```

```
# Graphing the number of biden tweets by sentiment
data_b = {'Positive':biden_positive,'Neutral':biden_neutral,'Negative':biden_negative}
sentiment_b = list(data_b.keys())
num_tweets_b = list(data_b.values())

plt.figure(figsize = (8, 5))

plt.bar(sentiment_b, num_tweets_b, color ='blue', width = 0.5, edgecolor='black')

plt.xlabel("Sentiment", fontweight ='bold')
plt.ylabel("Number of Tweets", fontweight ='bold')
plt.title("Biden Tweets by Sentiment", fontweight ='bold')
plt.show()
```

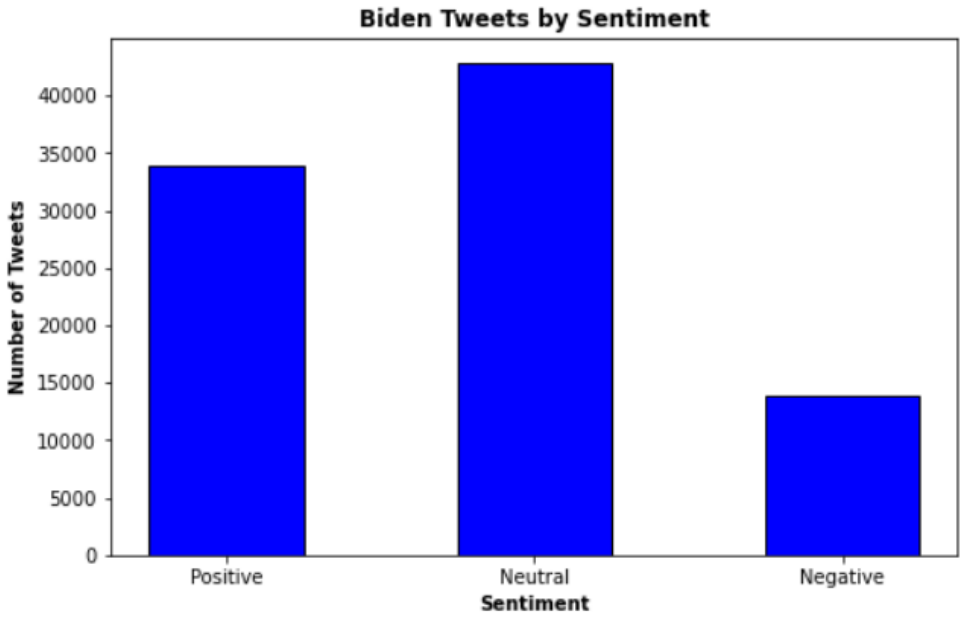


Figure 16 Biden tweets by sentiment (No. of tweets VS Sentiments)

```
# Calculate relative percentages by sentiment - Biden
total_tweets_b = len(biden_df.Sentiment)
prop_tweets_b = list(map(lambda x: round(x/total_tweets_b,2), num_tweets_b))
```

```
# Graphing relative percentages of biden tweets
bar_width = 0.25
plt.subplots(figsize=(8,8))

br1 = np.arange(3)
br2 = [x + bar_width for x in br1]

b = plt.bar(br2, prop_tweets_b, color ='b', width = bar_width, edgecolor ='black', label ='Biden')

plt.xlabel('Sentiment',fontweight ='bold')
plt.ylabel('Percentage of Tweets',fontweight ='bold')
plt.xticks([r + bar_width/2 for r in range(3)],['Positive','Neutral','Negative'])
plt.legend([b],['Percentage of Biden Tweets'])
plt.ylim(0.0, 1.0)
plt.title('Proportions of Tweets By Sentiment',fontweight ='bold')
```

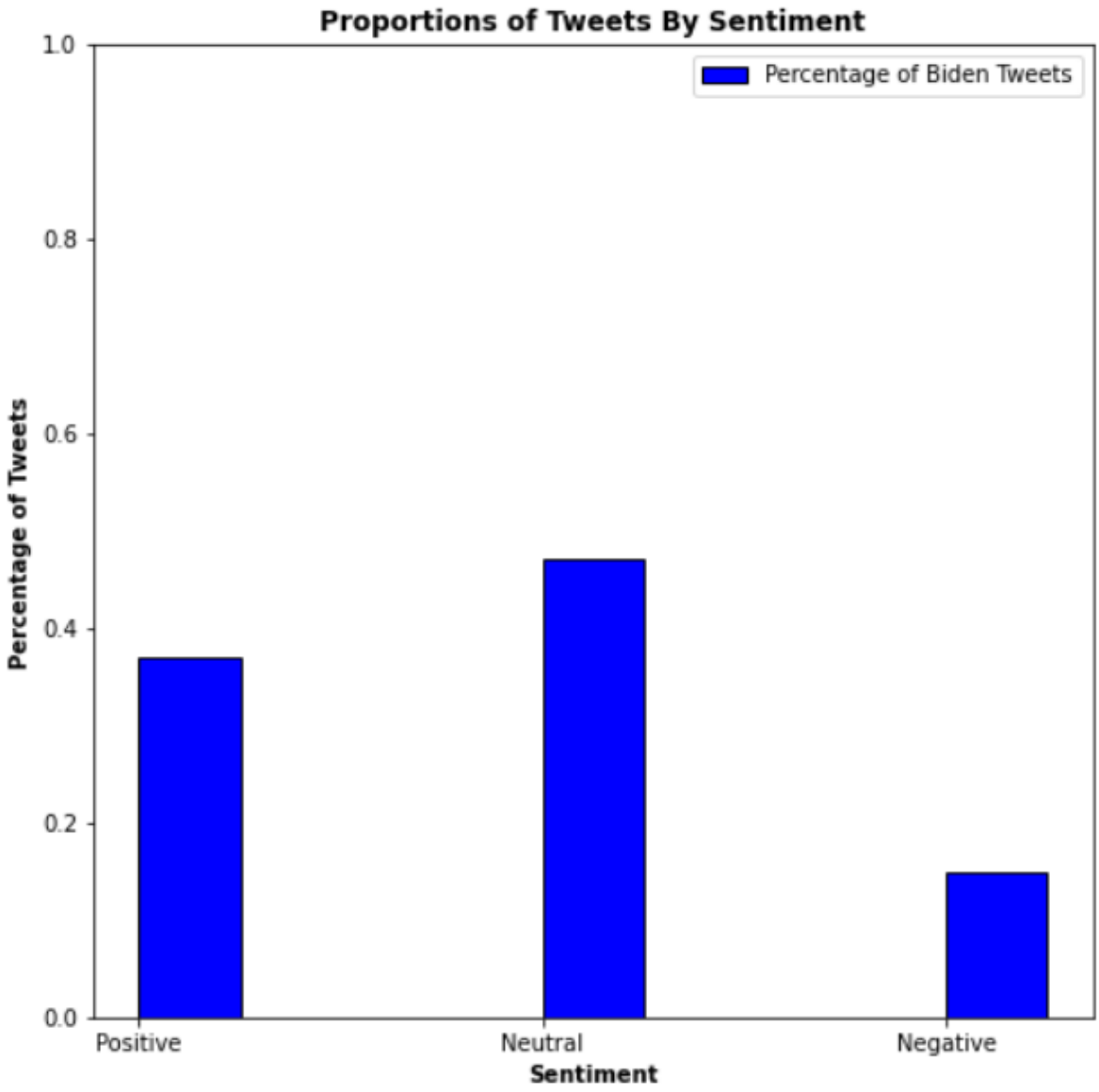


Figure 17 Percentage of tweets by sentiment (No. of tweets Percentage VS Sentiments)

```
# Create word strings
words_biden = get_all_tweets(biden_df)
words_pos_biden = get_all_tweets(biden_df,True,"positive")
words_neu_biden = get_all_tweets(biden_df,True,"neutral")
words_neg_biden = get_all_tweets(biden_df,True,"negative")
```

```
# Function to plot most frequent words
def plot_word_freq(tokens,sentiment,t_or_b,color):
    fdist = FreqDist(tokens)
    fdist_df = pd.DataFrame(fdist.most_common(10), columns = ["Word","Frequency"])
    fig = px.bar(fdist_df, x="Word", y="Frequency",title="<b>Most Frequently Used Words in </b>" + sentiment + " " + t_or_b + "<b>-Related Tweets</b>")
    fig.update_traces(marker=dict(color=color),selector=dict(type="bar"),marker_line_color='black', marker_line_width=1.5, opacity=0.6)
    fig.show()
```

```
# Most frequent words in all biden tweets
plot_word_freq(tokens_biden,"<b>ALL</b>","<b>Biden</b>","blue")
```

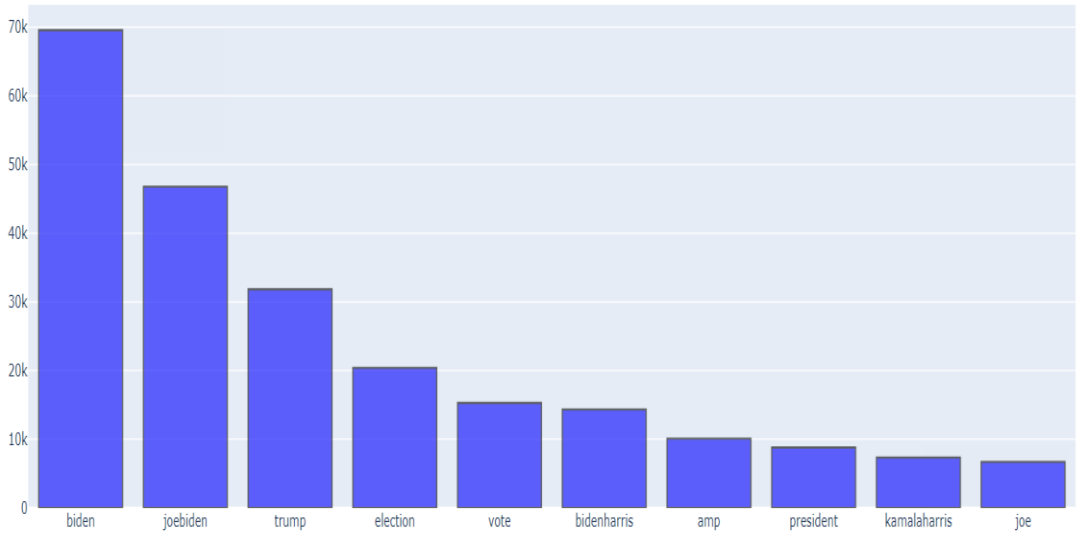


Figure 18 Most frequent words in Biden tweets

```
# Most frequent words in positive biden tweets
plot_word_freq(tokens_pos_biden,"<b>POSITIVE</b>","<b>Biden</b>","green")
```

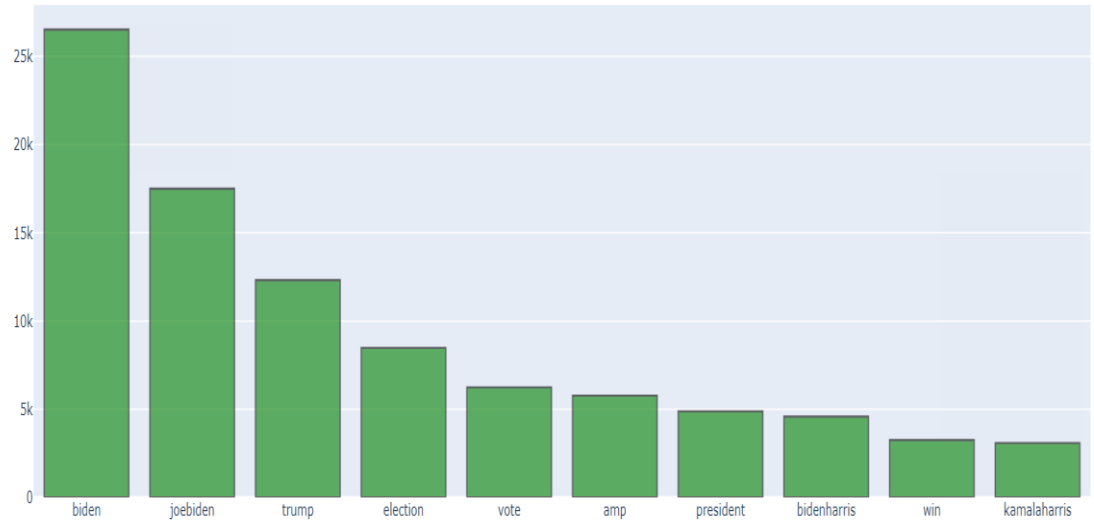


Figure 19 The most frequent words in positive Biden tweets.

```
# Most frequent words in negative biden tweets
plot_word_freq(tokens_neg_biden,"<b>NEGATIVE</b>","<b>Biden</b>","red")
```

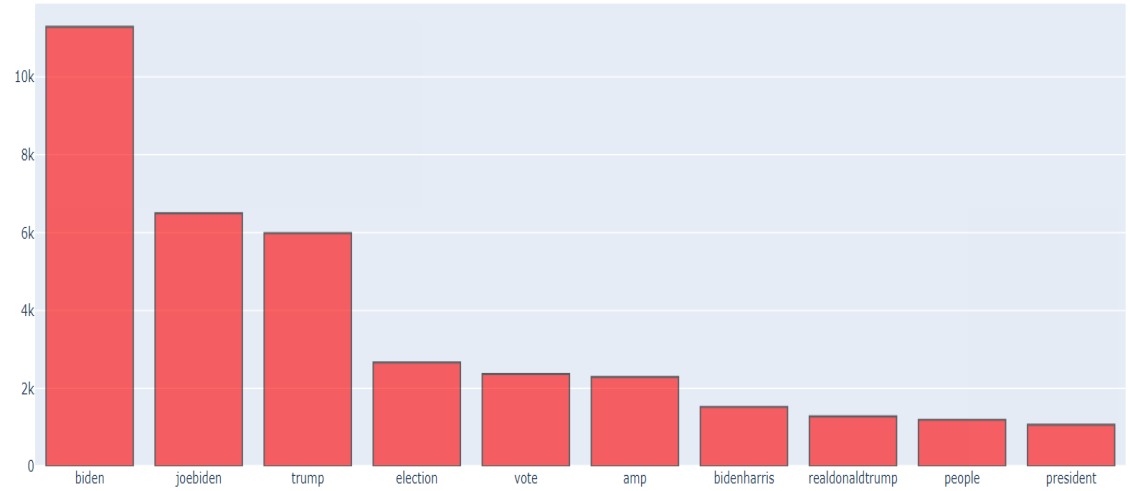


Figure 20 Most frequent words in negative Biden tweets

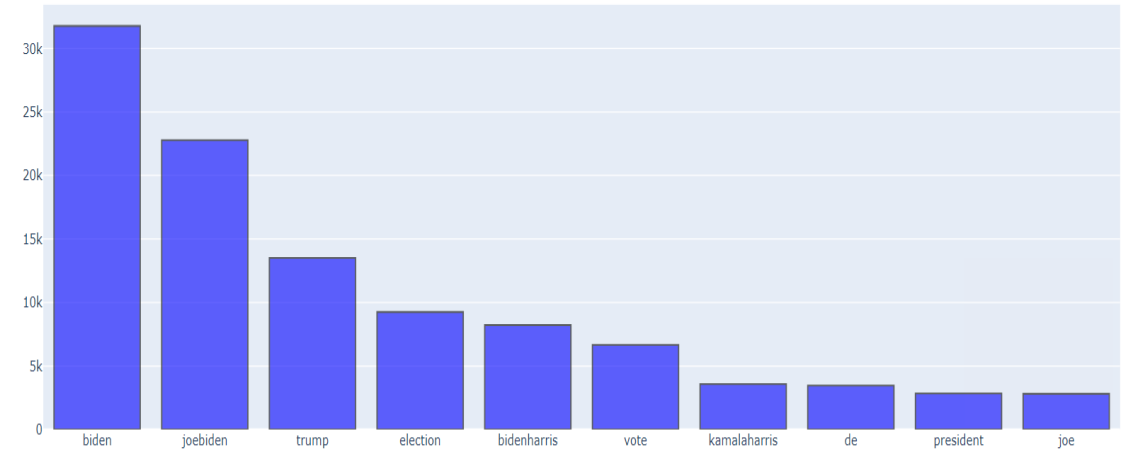


Figure 21 Most frequent neutral Biden tweets.

```
# Average polarity by state - Biden
biden_state_polarity = biden_df.groupby("State",as_index=False).mean()

fig = px.bar(biden_state_polarity, x="State", y="Polarity",
             title="<b>Average Polarity of Biden-Related Tweets by State</b>")
fig.update_traces(marker=dict(color="blue"),selector=dict(type="bar"),
                  marker_line_color='black', marker_line_width=0.8, opacity=0.6)
fig.show()
```

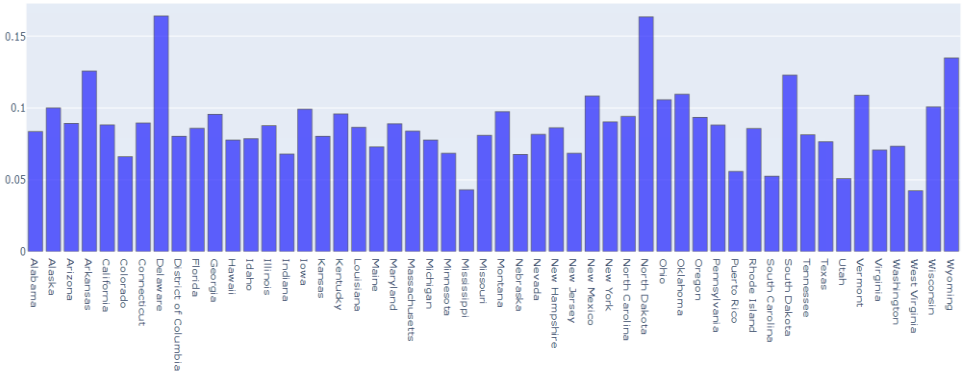
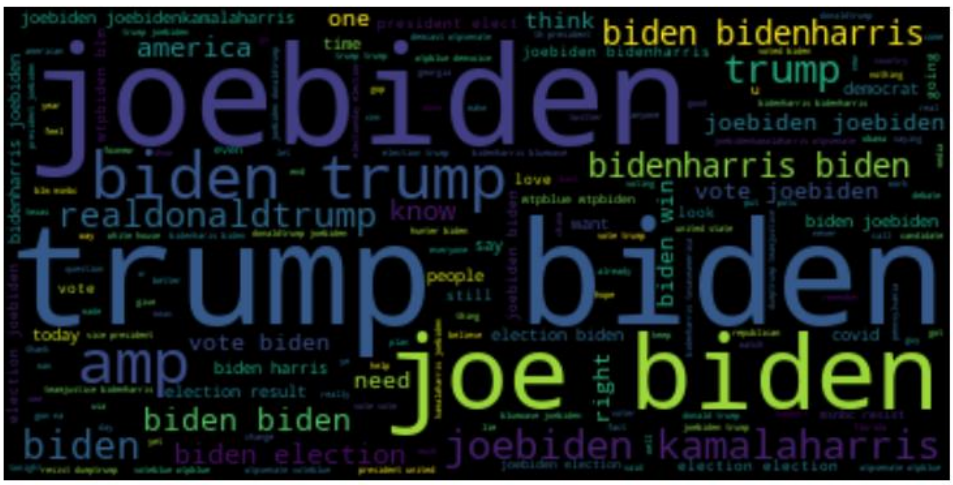


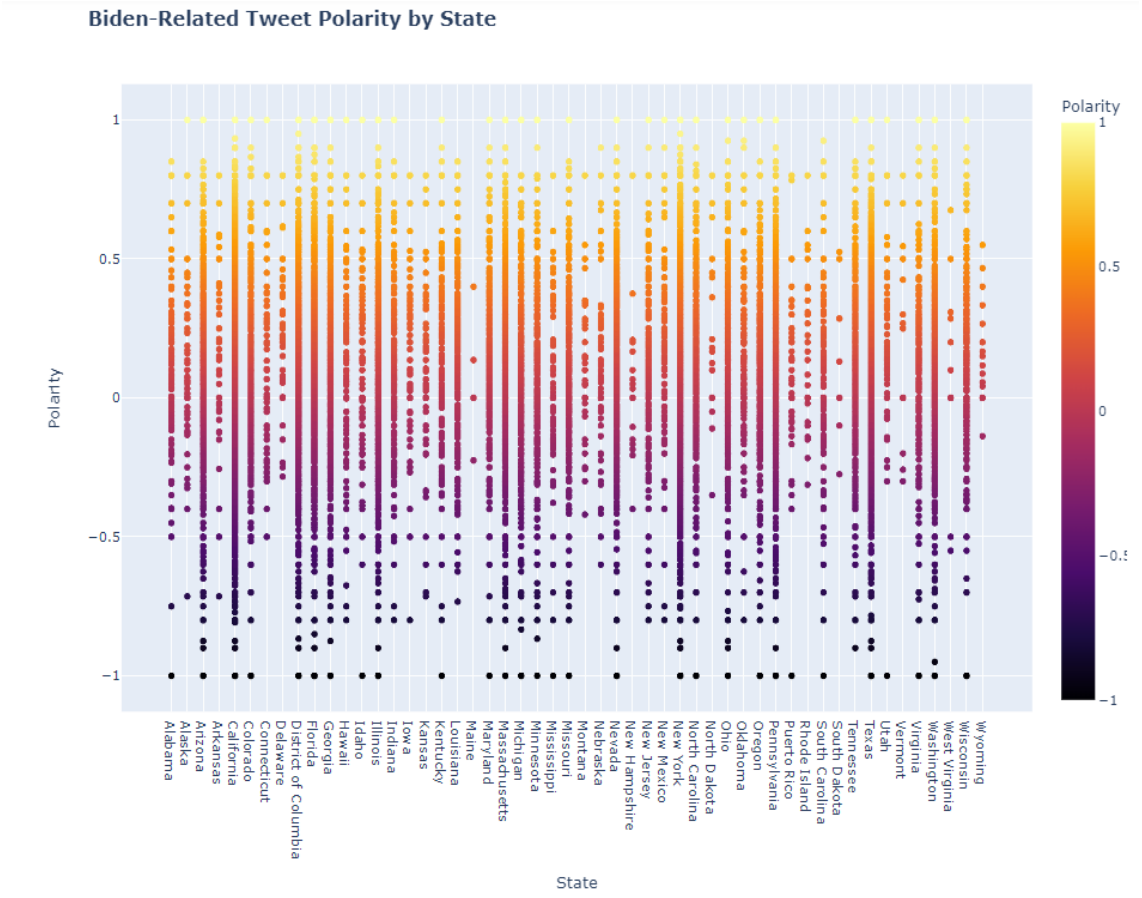
Figure 22 Average popularity by state

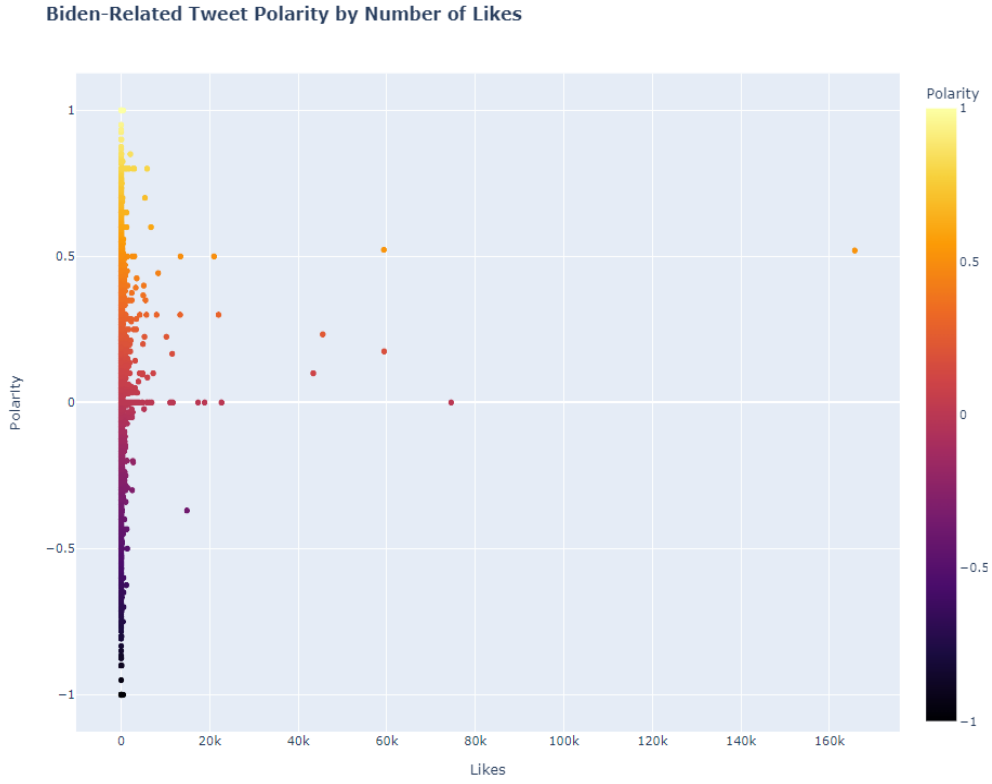
WordCloud

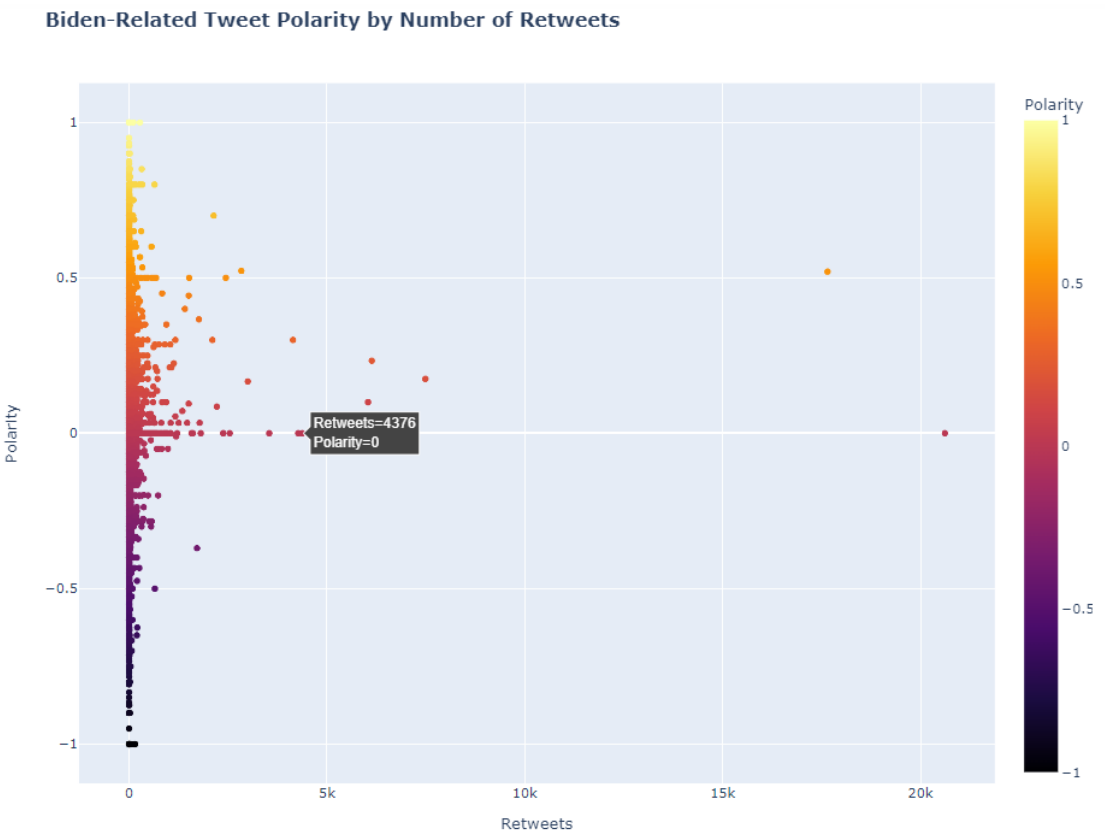
In [29]:
```
# Function to generate word cloud
def create_wordcloud(words):

    # create wordcloud
    wordcloud = WordCloud(max_font_size=200, max_words=200,
                          background_color="black").generate(words)

    # display the generated image
    plt.figure(1,figsize=(13, 13))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis("off")
    plt.show()
```

In [30]:
```
# Generate word cloud of biden tweets
create_wordcloud(words_biden)
```




Biden-Related Tweet Polarity by State
Polarity
1
0.5
0
−0.5
−1
Alabama
Alaska
Arizona
Arkansas
California
Colorado
Connecticut
Delaware
District of Columbia
Florida
Georgia
Hawaii
Idaho
Illinois
Indiana
Iowa
Kansas
Kentucky
Louisiana
Maine
Maryland
Massachusetts
Michigan
Minnesota
Mississippi
Missouri
Montana
Nebraska
Nevada
New Hampshire
New Jersey
New Mexico
New York
North Carolina
North Dakota
Ohio
Oklahoma
Oregon
Pennsylvania
Puerto Rico
Rhode Island
South Carolina
South Dakota
Tennessee
Texas
Utah
Vermont
Virginia
Washington
West Virginia
Wisconsin
Wyoming
State


Biden-Related Tweet Polarity by Number of Likes
Polarity
1
0.5
0
−0.5
−1
0
20k
40k
60k
80k
100k
120k
140k
160k
Likes



ACCURACY OF THE MODEL

Vader Is A Pre-Trained Model And Has An Accuracy Of 96% Which Outperforms Individual Human Raters (84% Accuracy) At Correctly Labelling The Sentiment Of Tweets Into Positive, Neutral, Or Negative Classes.

REFERENCES

1. Sujeet. Sharma, Nisha P Shetty, “Determining the popularity of political parties using twitter sentimental analysis”, pp. 5 -7, 2018
2. Mai Elsherief, Vivek Kulkarni, Dana Nguyen,William Yangwang, Elizabeth Belding
3. Deepak kumar Gupta , Raj kumar singh ,”Sentimental analyses of user data on Punjab Legislative assembly election
4. https://www.kaggle.com/thesammarks/2020-us-election-tweets-sentiment-analysis
5. https://towardsdatascience.com/detecting-hate-tweets-twitter-sentiment-analysis-780d8a82d4f6
6. https://www.askpython.com/python/examples/extracting-tweets-using-twitter-api
7. https://www.brandwatch.com/blog/amazing-social-media-statistics-and-facts
8. https://backlinko.com/social-media-users