A

Major Project

On

TWTTER SENTIMENTAL ANALYSIS TO PREDICT ELECTION RESULT

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

BY

Pudi Sandeep	(177R1A05A1)
Nethi Amarnath	(177R1A0596)
Joel Thankchan	(177R1A0584)

Under the Guidance of

N. BHASKAR

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)
Recognized Under Section 2(f) & 12(B) of the UGC Act.1956,
Kandlakoya (V), Medchal Road, Hyderabad-501401.
2017-2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**TWITTER SENTIMENTAL ANALYSIS TO PREDICT ELECTION RESULT**" being submitted by **PUDI SANDEEP (177R1A05A1), NETHI AMARNATH (177R1A0596) and JOEL THANKACHAN (177R1A0584)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering of the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2020-21.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

INTERNAL GUIDE
N. Bhaskar
Associate Professor

DIRECTOR
Dr. A. Raji Reddy

HOD
Dr. K. Srujan Raju

EXTERNAL EXAMINER

Submitted for viva voce Examination held on ____________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide

N. Bhaskar, Assoc. Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Mr. J. Narasimha Rao, Mr. B. P. Deepak Kumar, Mr. K. Murali, Dr. Suwarna Gothane** and **Mr. B. Ramji** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to the Head of the Department **Dr. K. Srujan Raju** for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We are obliged to our Director **Dr. A. Raji Reddy** for being cooperative throughout the course of this project. We would like to express our sincere gratitude to our Chairman Sri. **Ch. Gopal Reddy** for his encouragement throughout the course of this project

The guidance and support received from all the members of **CMR TECHNICAL CAMPUS** who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity thank our family for their constant encouragement without which this assignment would not be possible. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project.

PUDI SANDEEP (177R1A05A1)
NETHI AMARNATH (177R1A0596)
JOEL THANKACHAN (177R1A0584)

ABSTRACT

The entire world is changing at a breakneck pace, and technology is no exception. User-generated data is abundant on social networking platforms like Twitter. Users from all around the world offer their thoughts, opinions, ideas, and feelings about a variety of topics, including products, movies, and politics. Manual sentiment analysis is a time-consuming task. Opinion mining has recently gained popularity as a result of the large volume of opinionated data available on social networking sites such as Twitter. In this paper, we used a chronological approach to data collection, data pre-processing, emotional analysis, and machine learning analysis to forecast the outcome of the US 2020 presidential election using Twitter emotional analysis. We used a Random Forest classifier after completing a literature review and comparing all supervised ensemble machine learning algorithms to determine which one was the best. The proposed technique was tested on Twitter data, and it outperformed existing approaches.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

The project is titled as “Twitter Sentimental analysis to predict election result”. This project provides the beforehand election result to be exact public favorite and also provide public attitude with the winner. First tweets collected according to requirement directly from twitter with four authentication keys and preprocessed. Secondly, Sentiment extracted. Furtherly, moved to Random Forest Classifier to see the how accurate the prediction.

1.2 PROJECT PURPOSE

This has been developed to facilitate the prediction of election result in before the main election commencement and also can know public reaction about the election winner. Over the top Random Forest Classifier is employed in this project can show highest accuracy possible. Real time tweets been used in this project which are directly fetched from twitter.

1.3 PROJECT FEATURES

The term opinion mining has gained popularity as a result of the large volume of opinionated data available on social networking sites such as Twitter. Opinion mining is been employed in various fields like movie reviews, products and politics. Main motto is to know public opinion so this project serves the same. Major features of this project is to predict election result in before election through tweets and also can know public opinion on elected candidate.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

A detailed study of the process must be made by various techniques like data collection, feature extraction, sentiment extraction and choice of reliable machine learning approach. All the mentioned terms are major drawbacks causing chaos of not opting best approach to get accurate results. So, we did a survey came up with an out of the world idea is our proposed system twitter sentimental analysis to predict election result and also employed best of every approach in above mentioned.

2.2 EXISTING SYSTEM

There are so many real time approaches to predict results through public opinion as mentioned in problem definition every step selection is crucial this is where existing system is falling behind and final results are not desired.

2.3 PROPOSED SYSTEM

Our proposed system in keeping view of drawbacks of existing system we came up with best approach called Twitter sentimental analysis to predict election result. This project main aim is to predict election result through real time public tweets. Firstly, tweets collected from twitter using four authentication keys. Secondly, using TextBlob sentimental analysis performed to extract subjectivity and polarity. Next data is preprocessed to avoid noisy data for efficient results. Eventually, data is diversified into test and train set directed to Random Forest Classifier to get accuracy score of prediction.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires moderate system resources and the system will work in almost all configurations. It has got following features

- prediction accuracy
- time efficient
- cost efficient

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company.
Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.

- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

• Processor	:	Intel Quad Core@ CPU 2.90GHz.
• Hard disk	:	40GB and Above.
• RAM	:	2GB and Above.
• Monitor	:	15.6 inches or above

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

• Operating System	:	Windows or Linux
• Languages	:	Python 3.2.x and above
• IDE	:	Jupyter Notebook

3.ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

In this paper, we offer a Random Forest Classifier-based technique for forecasting election results on Twitter sentiment. The use of a Random Forest Classifier is intended to improve accuracy. The suggested method entails data collecting, pre-processing, sentiment analysis, and machine learning analysis, with the findings visualized in a bar graph. The detailed architecture is explained below.

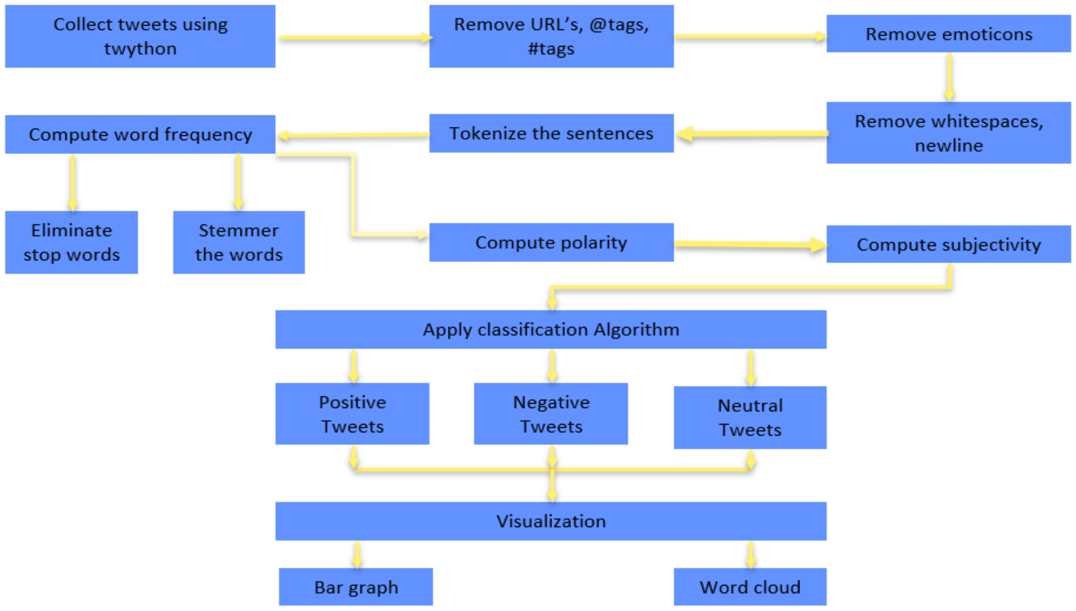


Fig. 3.1 Project Architecture of Twitter sentimental analysis to predict election result

3.2 DESCRIPTION

3.2.1 Data Collection

Because we chose Twitter as our real-time data source, Python provides a library called Twython for retrieving data from Twitter. Twython makes it simple to use the Twitter API without leaving the current window. To use twython to retrieve tweets, we'll need 4 keys: the Consumer Key, Consumer Secret, Access Token, Access Token Secret, and

Authentication Keys. We created a Twython instance with the application keys and the users' OAuth tokens using these. The search function is displayed in Fig 3 after the tag has been set up.

Authentication keys

Fig 3. Twitter Data Collection

3.2.2 Data Pre-processing

Data mining on Twitter is a difficult task. Raw tweets, often known as noisy data, are what we get from Twitter. For further machine learning analysis, this must be pre-processed, also known as feature extraction. Feature engineering is the name given to the process of extracting features. Because the accuracy of raw tweets can be affected. Lower case conversion, punctuation, removal of hashtags and other Twitter notations (@ RT), URLs, stop words, Tokenization, and stemming are all part of the pre-processing activity. In this work we used following 7 steps for Data Pre-processing:

a. Lower case conversion: The system is unable to distinguish between uppercase and lowercase letters. Lower casing can be accomplished in Python by utilising the lower () built-in function.

b. Remove Punctuations: regular expressions are used all punctuation marks should be replaced with blanks.

c. Remove hashtags, @RT and URL: Replace all hashtag, @, and https preceding content with blanks using regular expression.

d. Remove stop words: Import stop-words from the Natural Language Toolkit (NLTK) dataset and delete them during feature extraction.

e. Stemming of words: WordNet is an English dictionary based on Natural Language Processing (NLP) that lemmatizes words to the root word. It may be downloaded from the NLTK library.

f. Tokenize of words: To tokenize the statement, the NLTK library download punkt uses an unsupervised tokenizer algorithm.

g. Remove missing values: First, see if there are any missing values in the fetched data, and if so, eliminate them.

3.2.3 Sentiment Analysis

A frequent NLP task is sentiment analysis, often known as opinion mining. The most important part of sentiment analysis is analysing a body of text in order to comprehend the sentiment communicated by the content. TextBlob is a direct library for sentiment analysis in Python that yields two properties: polarity score (PS) and subjectivity score (SS). Each tweet's polarity and subjectivity were retrieved using TextBlob. We categorised the tweets as favourable, negative, or neutral using the polarity score, as shown in Fig 4.

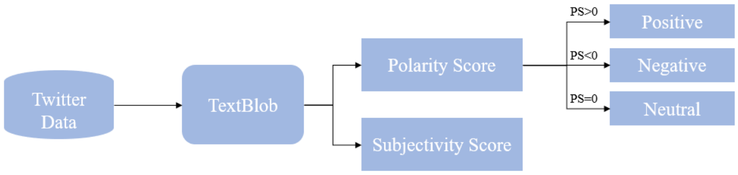


Fig 4. Sentimental Analysis with TextBlob

3.2.4 Random Forest Classifier

A random forest classifier is an ensemble learning classification method that works by generating a large number of decision trees during training and then outputs the classification class. This means that each decision tree should collect the results and output the most important ones.

The classifier data must be handled before moving on to Random Forest (RF). Bag of Words (BOW) is a feature extraction method that may be applied to the training of a Random Forest classification machine. It generates vectors after creating a vocabulary of all the unique terms found in all of the papers in the training set. A large document with a large produced vocabulary may produce a vector with many 0 values. This is referred to as a sparse vector. When modelling using sparse vectors, more memory and computing resources are required. To circumvent this, the countvectorizer import from the sci-kit learn package is used to convert a given text into a vector 0 based on the frequency of each word in a tweet. Then, as illustrated in Fig 5, fit the result to a sparse matrix that can handle big data.

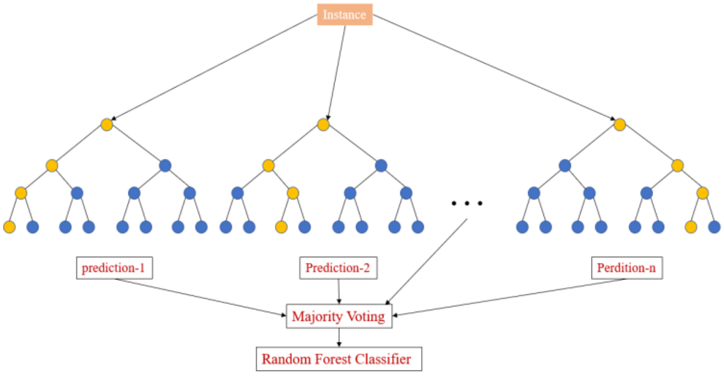


Fig 5. Random Forest Classifier

For splitting data arrays into two subsets, training set and testing set, the machine uses the sci-kit learn import train test split module. The set test was set to 30%. It's now completely prepared for machine learning analysis. Import random forest classifier from sci-kit learn sub module ensemble learning, fit the training data into the RF classifier, then predict the test data. Finally, determine the accuracy of the test and training set.

Because we retrieved real-time tweets, the number of tweets varies from time to time. We worked on Donald Trump and Joe Biden's tweets because they are the front-runners in the US presidential election in 2020. The accuracy of the model we developed and trained on two tweets ranges from 71% to 89% for Trump tweets and 73% to 85% for Biden tweets.

3.3 USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the source and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

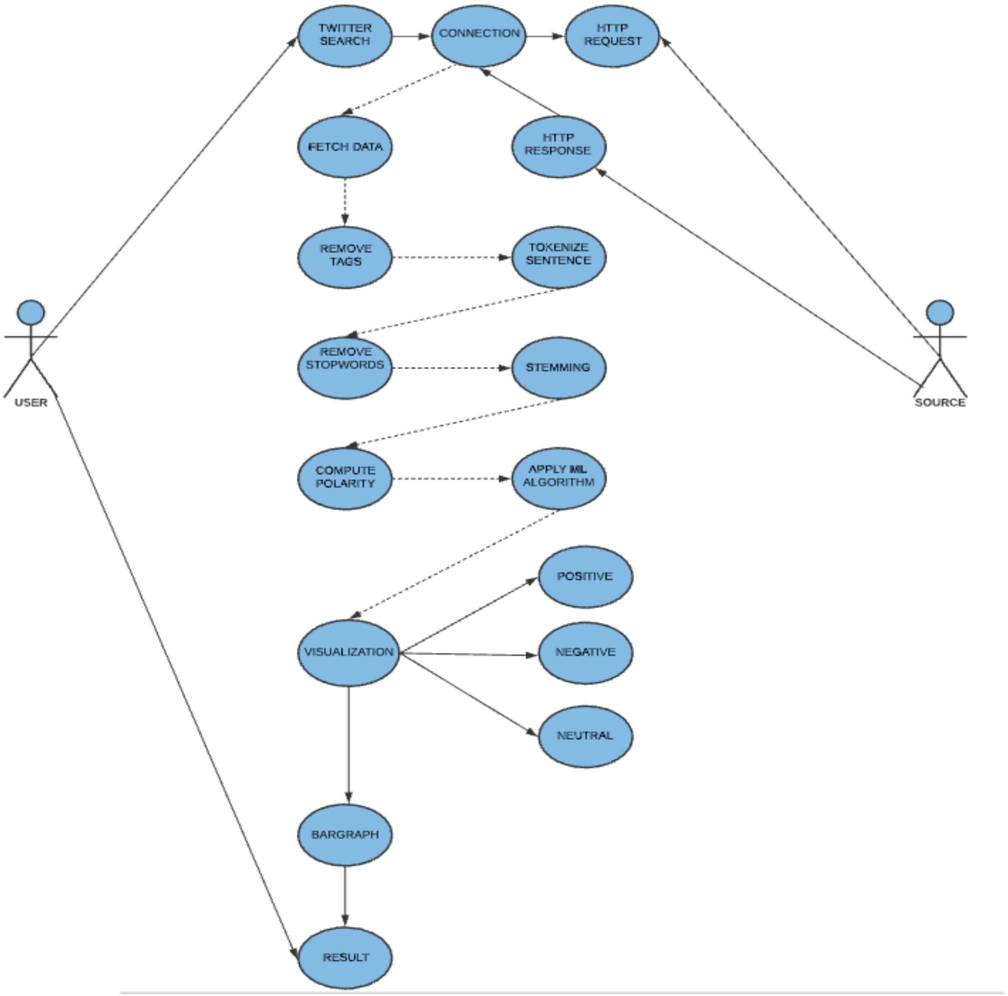


Fig. 3.2 Use Case Diagram for Twitter Sentimental Analysis to predict election result

3.4 SEQUENCE DIAGRAM

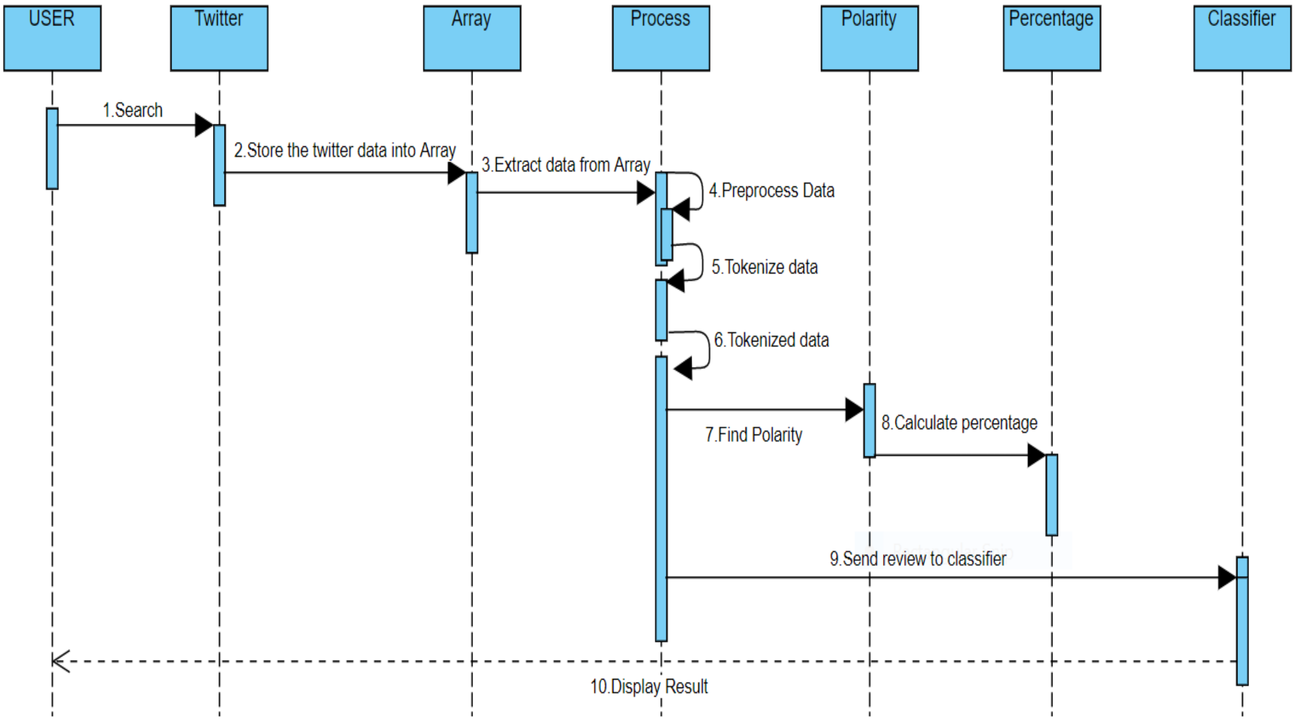


Fig.3.3.2 Sequence Diagram for Twitter Sentimental Analysis to predict election result

3.5 CLASS DIAGRAM

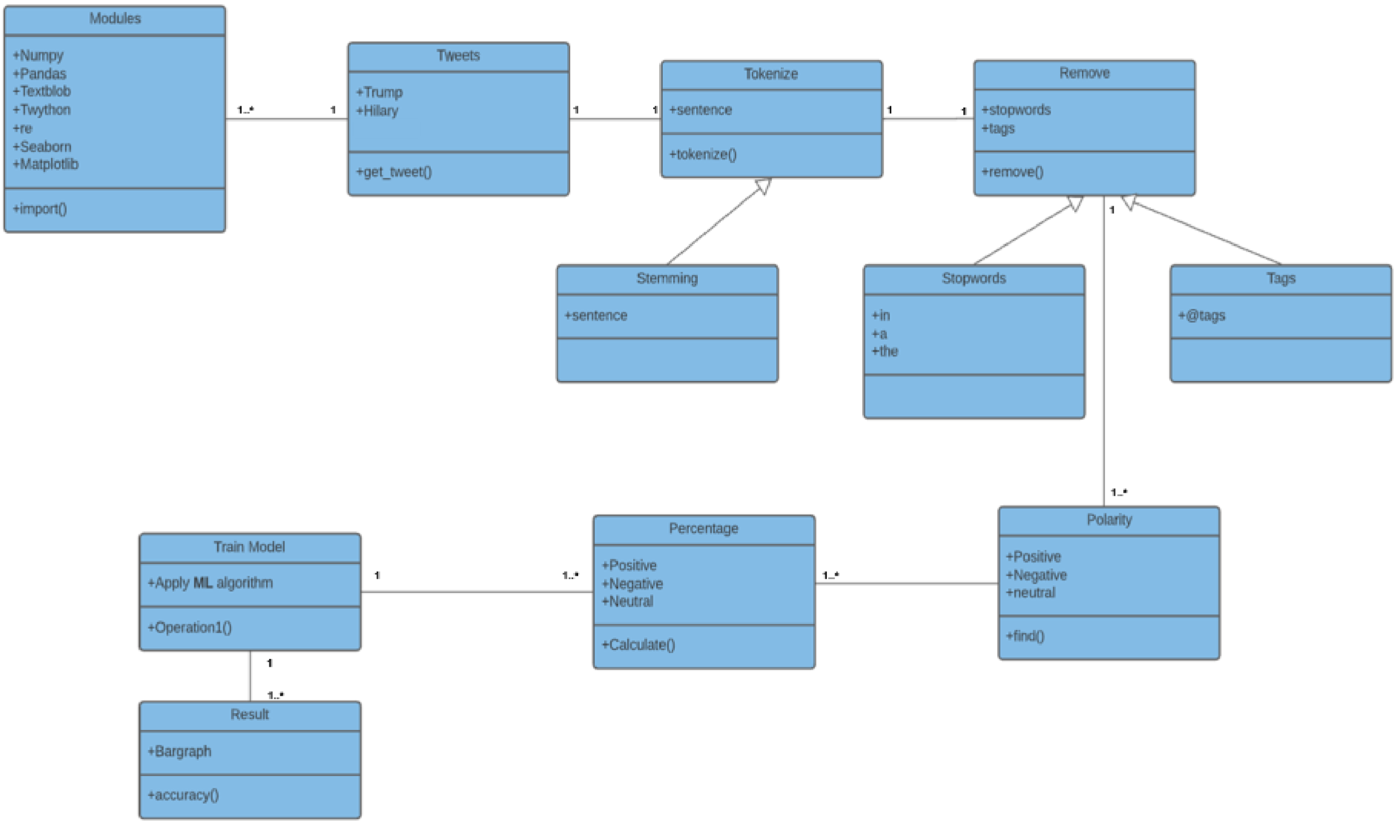


Fig. 3.4 Class Diagram for Twitter Sentimental Analysis to predict election result

3.6 ACTIVITY DIAGRAM

It describes about flow of activity states.

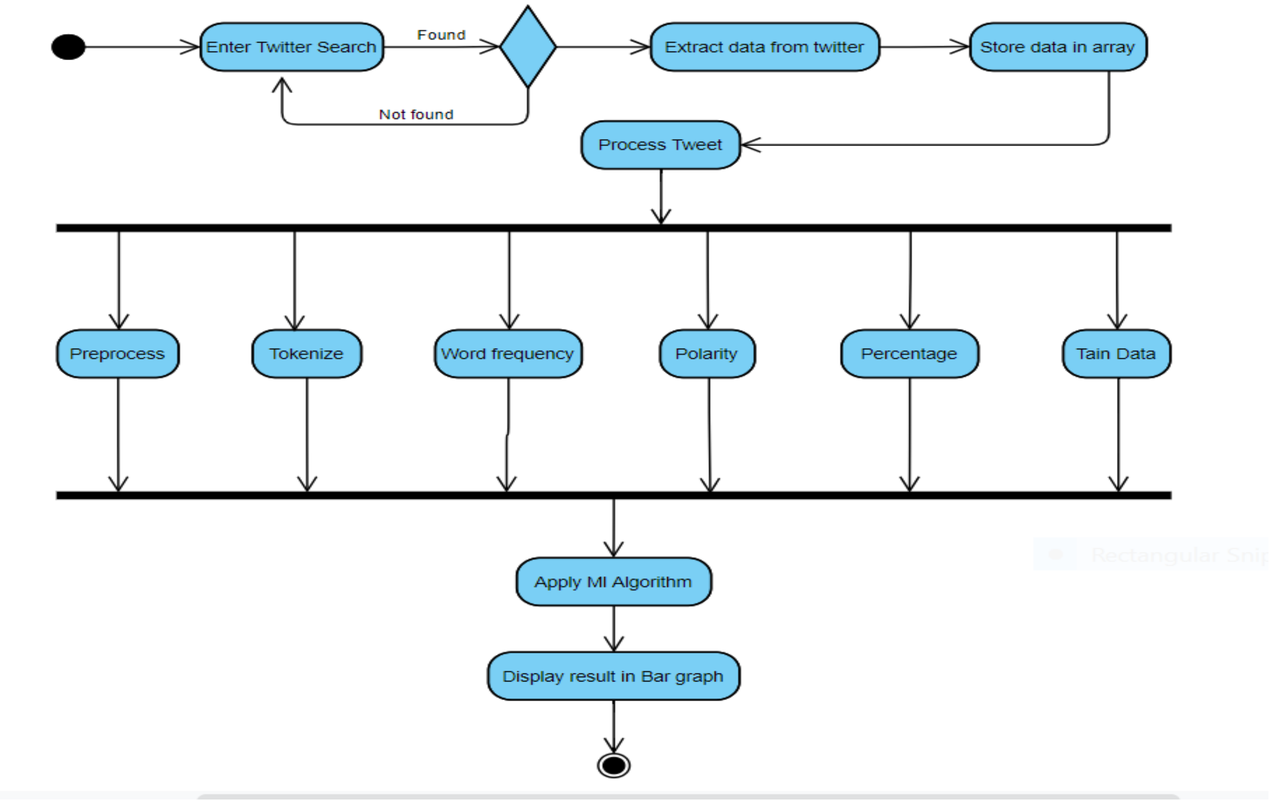


Fig. 3.5 Activity Diagram for Twitter Sentimental analysis to predict election result

4. IMPLEMENTATION

4. IMPLEMENTATION

4.1 SAMPLE CODE

adminhome.jsp:

```
 #!pip install sklearn

#!pip install seaborn

#!pip install textblob

#!pip install twython


import numpy as np
import pandas as pd
from textblob import TextBlob
from twython import Twython
import re
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import matplotlib.pyplot as plt
import nltk
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import warnings
warnings.filterwarnings('ignore')
def get_tweet_sentiment(tweet):
    analysis = TextBlob(tweet)
    if analysis.sentiment.polarity > 0:
        return 'positive'
    elif analysis.sentiment.polarity == 0:
        return 'neutral'
    else:
        return 'negative'
def getSubjectivity(tweet):
    return TextBlob(tweet).sentiment.subjectivity
def getPolarity(tweet):
    return TextBlob(tweet).sentiment.polarity
def get_tweet(q):
    app_key = 'bdKaYkyCcSmf9f6Wfqj8VvIuR'
    app_secret =
            '1S1WMaZJxdIFyiniKjDeLFCBDRPP72bx2HCxI2cwMTx0
            Y7bYOy'
    oauth_token = '786603866498269184-
            DSnfKXCZhYNRUjboLDwHWaZ7InaFslf'
    oauth_token_secret =
            'QDRQK2huoL8hpo3MvaqrNdrGtvWWxCd6sCCI8B0X95v
            8y'
    try:
        t = Twython(app_key=app_key, app_secret=app_secret,
            oauth_token=oauth_token,
            oauth_token_secret=oauth_token_secret)
        search =t.search(q=q,count=200,since=2020-11-15)
        fetched_tweets = search['statuses']
        text=[]
        for i in fetched_tweets:
            text.append(i.get('text'))
        sentiment=[]
        for tweet in text:
            sentiment.append(get_tweet_sentiment(tweet))
        subjectivity=[]
        for tweet in text:
            subjectivity.append(getSubjectivity(tweet))
        polarity=[]
        for tweet in text:
            polarity.append(getPolarity(tweet))

    except:
        print("error")
    ptweets = [tweet for tweet in sentiment if tweet == 'positive']
    ntweets = [tweet for tweet in sentiment if tweet == 'negative']
    positive_tweet=int(100*len(ptweets)/len(sentiment))
    negative_tweets=int(100*len(ntweets)/len(sentiment))
    neutral_tweets=int(100*(len(sentiment) - len(ntweets) -
            len(ptweets))/len(sentiment))
    return
            text,sentiment,positive_tweet,negative_tweets,neutral_tweets
            ,subjectivity,polarity

text_trump,sentiment_trump,x1,y1,z1,subjectivity_trump,polarity_tr
            ump=get_tweet('donald trump')
text_biden,sentiment_biden,x2,y2,z2,subjectivity_biden,polarity_bid
            en=get_tweet('joe biden')
df_trump=pd.DataFrame()
df_trump['text']=text_trump
df_trump['sentiment']=sentiment_trump
df_trump['Subjectivity']=subjectivity_trump
df_trump['polarity']=polarity_trump
%matplotlib inline
print(len(df_trump))
print(x1)
df_trumpdf_trump['sentiment'].value_counts().plot(kind='bar')

df_biden=pd.DataFrame()
df_biden['text']=text_biden
df_biden['sentiment']=sentiment_biden
```

```
df_biden['Subjectivity']=subjectivity_biden
df_biden['polarity']=polarity_biden
print(x2)
df_biden

df_biden['sentiment'].value_counts().plot(kind='bar')
df_trump.isnull().sum()
df_biden.isnull().sum()
from sklearn.preprocessing import LabelEncoder
lb = LabelEncoder()
df_trump['sentiment']=lb.fit_transform(df_trump['sentiment'])
df_trump['Candidate'] = 'trump'
df_trump

from sklearn.preprocessing import LabelEncoder
lb = LabelEncoder()
df_biden['sentiment']=lb.fit_transform(df_biden['sentiment'])
df_biden['Candidate'] = 'biden'
df_biden

df1 = pd.concat([df_trump,df_biden])

df1.shape

df1['Candidate'].value_counts()

df1['sentiment'].value_counts()
import seaborn as sns
import matplotlib.pyplot as plt
#2 : Positive, 1 : Negative ,0 : Neutral
# visualization by frequency table
plt.rcParams['figure.figsize'] = (18,7)
color = plt.cm.copper(np.linspace(0, 1, 40)) #Returns number spaces
          evenly w.r.t interval. Similar to arange but instead of step it
          uses sample number.
df1['sentiment'].value_counts().head(40).plot.bar(color = color)
plt.title('Sentiments for all candidate', fontsize = 20)
plt.xticks(rotation = 80 )
plt.grid()
plt.show()

import seaborn as sns
sns.set(style="darkgrid",color_codes = "g")
sns.countplot(x="sentiment",hue = 'Candidate', data=df1)

def prediction():
   points=[x1,x2]
   points.sort()
   max = points[1]
```

```
    if max == x1:
        print('winner is trump')
    else:
        print('winner is biden')

prediction()
import re
first_text_trump=df_trump.text[0]
text_re_trump=re.sub(r"RT @[\w]*:","",first_text_trump)
text_re_trump=re.sub(r"https?://[^a-zA-Z]"," ",first_text_trump)
            #changing characters with space
text_re_trump=re.sub(r'^https?:\/\/.*[\r\n]*', '', first_text_trump,
            flags=re.MULTILINE)
text_re_trump=re.sub(r"http\S+|www\S+|https:\S+", '',
            first_text_trump)
text_re_trump=re.sub(r'\@\w+|\#','',first_text_trump)
text_re_trump=text_re_trump.lower()
text_re_trump
#!pip install nltk
import warnings
warnings.filterwarnings('ignore')
import nltk
#nltk.download('punkt')
from nltk.corpus import stopwords
text_re_trump=nltk.word_tokenize(text_re_trump) #separate all
            words
text_re_trump
import nltk as nlp
import nltk
#nltk.download('wordnet')
lemma=nlp.WordNetLemmatizer()
text_re_trump=[lemma.lemmatize(i) for i in text_re_trump]
text_re_trump=" ".join(text_re_trump)
text_re_trump
import nltk as nlp
text_list=[]
for i in df_trump.text:
    text_re_trump=re.sub("[^a-zA-Z]"," ",i)
    text_re_trump=text_re_trump.lower()
    text_re_trump=nltk.word_tokenize(text_re_trump)
    lemma=nlp.WordNetLemmatizer()
    text_re_trump=[lemma.lemmatize(word) for word in
            text_re_trump]
    text_re_trump=" ".join(text_re_trump)
    text_list.append(text_re_trump)
text_list[:100]
#bag of words
# COUNT VECTORIZER
```

```
from sklearn.feature_extraction.text import CountVectorizer
max_features=100
cou_vec=CountVectorizer(max_features=max_features,stop_words=
        "english")
sparse_matrix= cou_vec.fit_transform(text_list).toarray()
Y = df_trump.iloc[:,1].values
X = sparse_matrix
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.30)
Y
X
print(X.shape)
print(Y.shape)
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier()
rf.fit(X_train,Y_train)
y_pred_rf = rf.predict(X_test)
RF = accuracy_score(y_pred_rf,Y_test)
print("\nRF classifier:-
        ",round(accuracy_score(Y_test,y_pred_rf)*100))
import re
first_text_biden=df_biden.text[0]
text_re_biden=re.sub(r"RT @[\w]*:","",first_text_biden)
text_re_biden=re.sub(r"https?://[^a-zA-Z]"," ",first_text_biden)
        #changing characters with space
text_re_biden=re.sub(r'^https?:\/\/.*[\r\n]*', ", first_text_biden,
        flags=re.MULTILINE)
text_re_biden=re.sub(r"http\S+|www\S+|https:\S+", ",
        first_text_biden)
text_re_biden=re.sub(r'\@\w+|\#',",first_text_biden)
text_re_biden=text_re_biden.lower()
text_re_biden
import warnings
warnings.filterwarnings('ignore')
import nltk
from nltk.corpus import stopwords
text_re_biden=nltk.word_tokenize(text_re_biden)
text_re_biden
import nltk as nlp
import nltk
#nltk.download('wordnet')
lemma=nlp.WordNetLemmatizer()
text_re_biden=[lemma.lemmatize(i) for i in text_re_biden]
text_re_biden=" ".join(text_re_biden)
```

```
text_re_biden
import nltk as nlp
text_list=[]
for i in df_biden.text:
   text_re_biden=re.sub("[^a-zA-Z]"," ",i)
   text_re_biden=text_re_biden.lower()
   text_re_biden=nltk.word_tokenize(text_re_biden)
   lemma=nlp.WordNetLemmatizer()
   text_re_biden=[lemma.lemmatize(word) for word in
         text_re_biden]
   text_re_biden=" ".join(text_re_biden)
   text_list.append(text_re_biden)
text_list[:100]
from sklearn.feature_extraction.text import CountVectorizer
max_features=100
cou_vec=CountVectorizer(max_features=max_features,stop_words=
         "english")
sparse_matrix= cou_vec.fit_transform(text_list).toarray()
Y = df_biden.iloc[:,1].values
X = sparse_matrix
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.30)

Y
X
print(X.shape)
print(Y.shape)

print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier()
rf.fit(X_train,Y_train)
y_pred_rf = rf.predict(X_test)
RF = accuracy_score(y_pred_rf,Y_test)
print("\nRF classifier:-
         ",round(accuracy_score(Y_test,y_pred_rf)*100))

#!pip install wordcloud
def plot_cloud(wordcloud):
   plt.figure(figsize=(40, 30))
   plt.imshow(wordcloud)
   plt.axis("off")
   plt.savefig('txt.png',facecolor='k',bbox_inches='tight')
from wordcloud import WordCloud, STOPWORDS,
         ImageColorGenerator
txt=''.join(df_trump['text'].tolist())
```

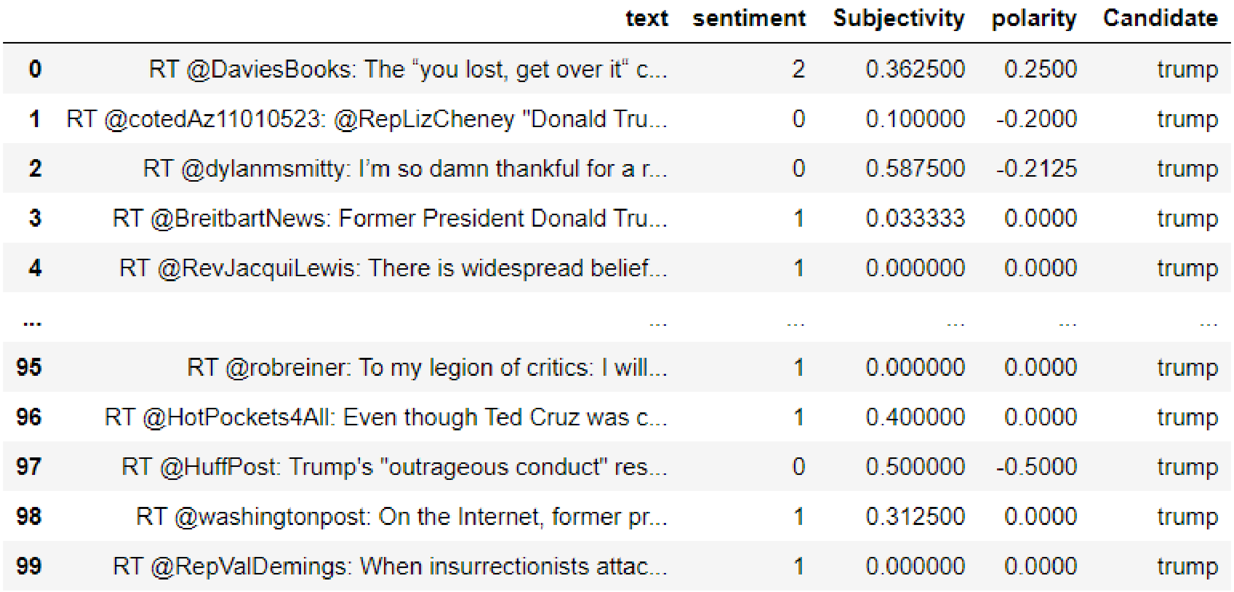
```
#wordcloud = WordCloud(width = 3000, height = 2000,
        random_state=1, background_color='salmon',
        colormap='Pastel1', collocations=False, stopwords =
        STOPWORDS).generate(txt)
wordcloud=
        WordCloud(background_color="white",width=1600,height=
        800,stopwords = STOPWORDS).generate(txt)
plot_cloud(wordcloud)
txt=".join(df_biden['text'].tolist())
#wordcloud = WordCloud(width = 3000, height = 2000,
        random_state=1, background_color='salmon',
        colormap='Pastel1', collocations=False, stopwords =
        STOPWORDS).generate(txt)
wordcloud=
        WordCloud(background_color="white",width=1600,height=
        800,stopwords = STOPWORDS).generate(txt)
plot_cloud(wordcloud)
```

5. SCREEN SHOTS

5. SCREEN SHOTS

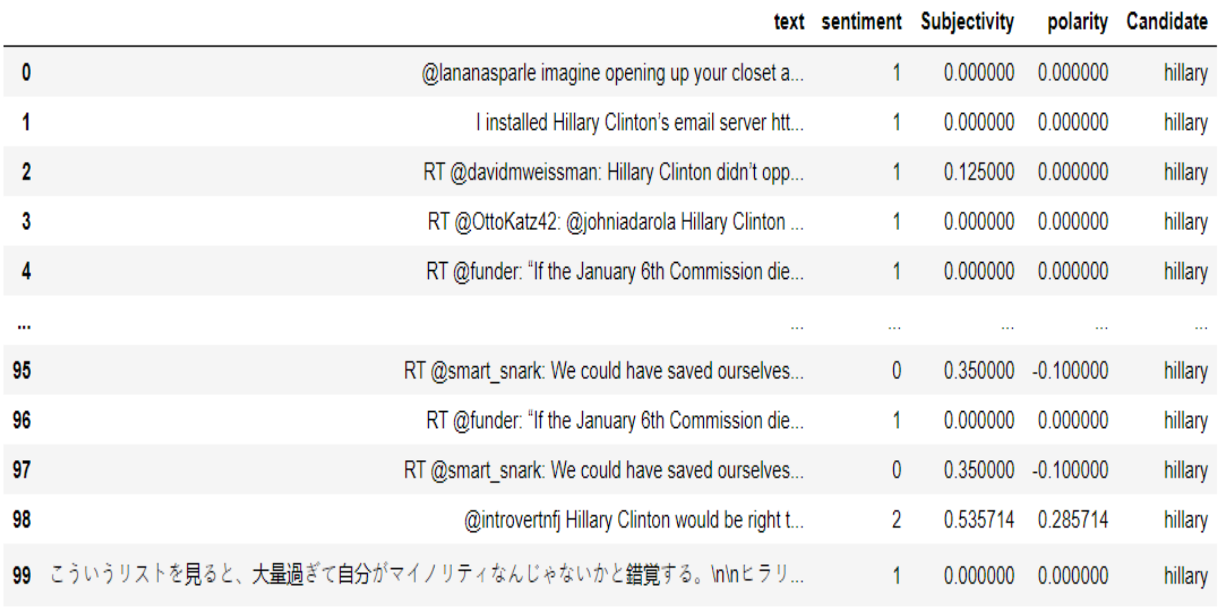
5.1 HOSTEL MANAGEMENT SYSTEM

This is the Home Page where user can see the modules in the project



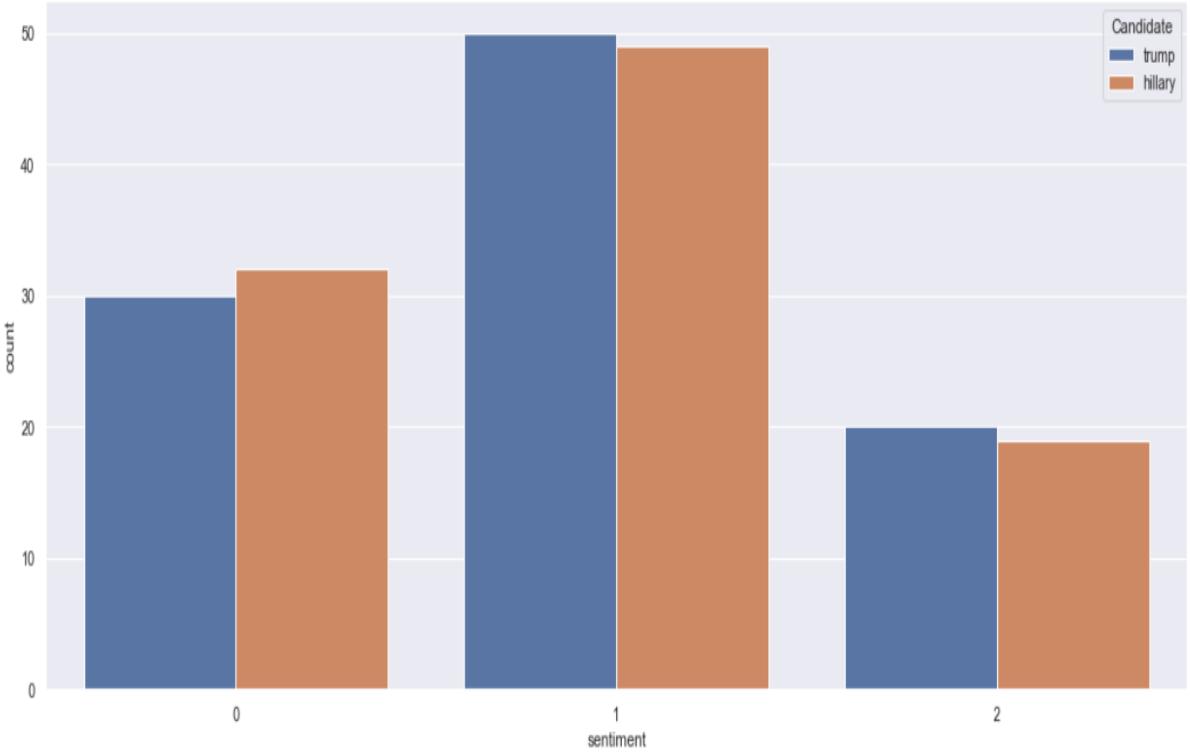
	text	sentiment	Subjectivity	polarity	Candidate
0	RT @DaviesBooks: The “you lost, get over it“ c...	2	0.362500	0.2500	trump
1	RT @cotedAz11010523: @RepLizCheney "Donald Tru...	0	0.100000	-0.2000	trump
2	RT @dylanmsmitty: I’m so damn thankful for a r...	0	0.587500	-0.2125	trump
3	RT @BreitbartNews: Former President Donald Tru...	1	0.033333	0.0000	trump
4	RT @RevJacquiLewis: There is widespread belief...	1	0.000000	0.0000	trump
...	...	...	...	...	...
95	RT @robreiner: To my legion of critics: I will...	1	0.000000	0.0000	trump
96	RT @HotPockets4All: Even though Ted Cruz was c...	1	0.400000	0.0000	trump
97	RT @HuffPost: Trump's "outrageous conduct" res...	0	0.500000	-0.5000	trump
98	RT @washingtonpost: On the Internet, former pr...	1	0.312500	0.0000	trump
99	RT @RepValDemings: When insurrectionists attac...	1	0.000000	0.0000	trump

5.1. Screenshot: **Trump Tweets**

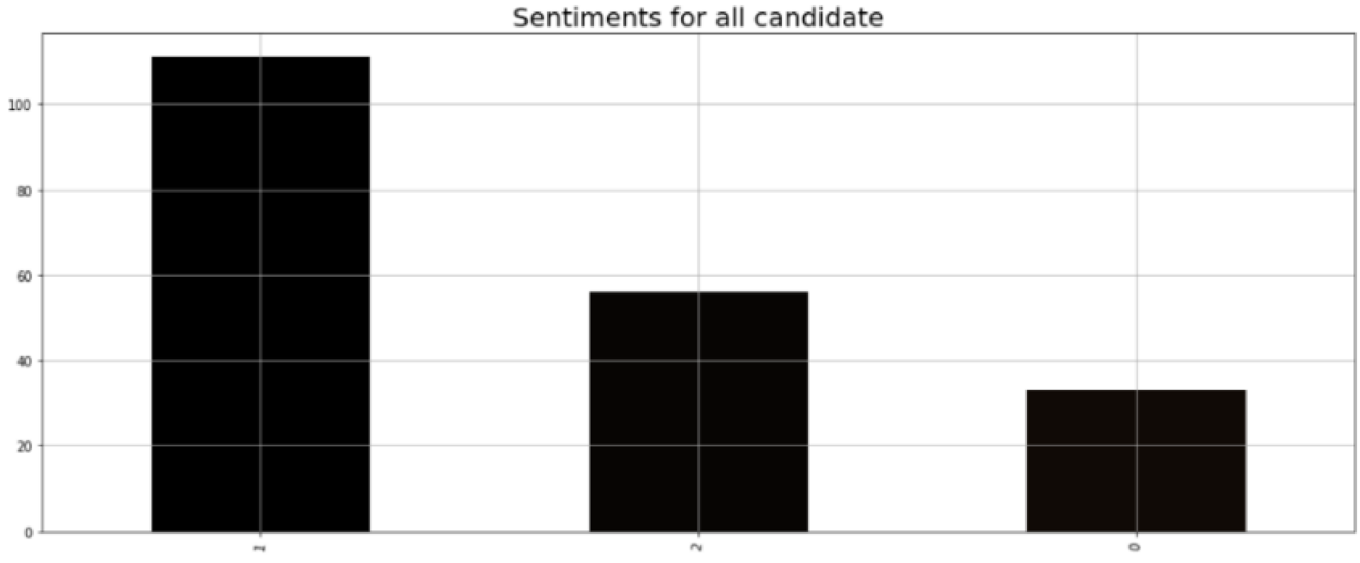


	text	sentiment	Subjectivity	polarity	Candidate
0	@lananasparle imagine opening up your closet a...	1	0.000000	0.000000	hillary
1	I installed Hillary Clinton's email server htt...	1	0.000000	0.000000	hillary
2	RT @davidmweissman: Hillary Clinton didn't opp...	1	0.125000	0.000000	hillary
3	RT @OttoKatz42: @johniadarola Hillary Clinton ...	1	0.000000	0.000000	hillary
4	RT @funder: "If the January 6th Commission die...	1	0.000000	0.000000	hillary
...	...	...	...	...	...
95	RT @smart_snark: We could have saved ourselves...	0	0.350000	-0.100000	hillary
96	RT @funder: "If the January 6th Commission die...	1	0.000000	0.000000	hillary
97	RT @smart_snark: We could have saved ourselves...	0	0.350000	-0.100000	hillary
98	@introvertnfj Hillary Clinton would be right t...	2	0.535714	0.285714	hillary
99	こういうリストを見ると、大量過ぎて自分がマイノリティなんじゃないかと錯覚する。\n\nヒラリ...	1	0.000000	0.000000	hillary

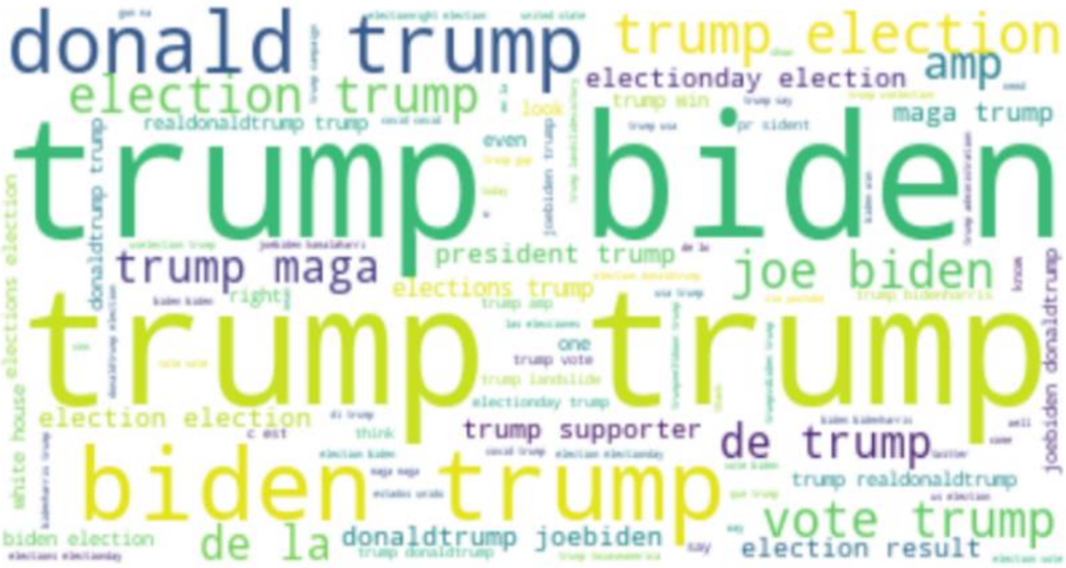
5.2 Screenshot : **Biden Tweets**



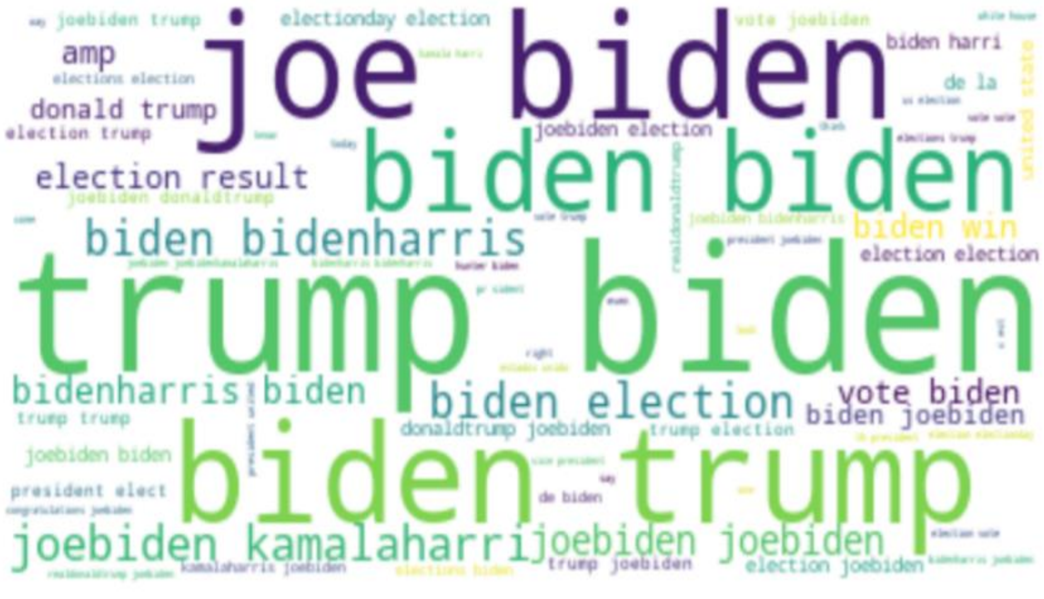
5.3 Screenshot: **Diversified polarity bar graph of trump and biden**



5.3Screenshot: **Comprehended polarity bar graph of trump and biden**



5.5 Screenshot: **Word Cloud of trump**



5.5 Screenshot: **Word Cloud of biden**

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

7. CONCLUSION

7. CONCLUSION & FUTURE ENHANCEMENTS

a. PROJECT CONCLUSION

The project “**TWITTER SENTIMENTAL ANALYSIS TO PREDICT ELECTION RESULT**” has been designed with much care, with the intention, easier and more complexity involved is presented in a simple and lucid style. Much care has been taken in designing database so that end users can handle any fact about the entries easily. At the time of the beginning in development of this project we had kept certain goals in the mind, and it is great pleasure that the system is meeting most of its requirements.

The goals that are expected to achieve by the software are:

- Simplification of the operation.
- Less processing time and increasing the productivity.
- Each transaction is updated and processed immediately.
- Avoiding the errors by minimizing human interaction through user friendly screens to enter data and retrieve the information from tables on like messages.
- Portable and flexible for further enhancement.

b. FUTURE ENHANCEMENTS

The system is developed in a modular fashion with different function performing their jobs. If any changes have to be done to the system then they can be easily achieved by making the changes to the functions and they won’t affect the remaining part of the system.

The design and development can definitely meet the user specifications. It also eradicates the limitations of the existing system. We made all our efforts to include all difficulties of existing system and the project is developed in python scripting language in jupyter notebook IDE.

8. BIBLIOGRAPHY

8. BIBILOGRAPHY

8.1 References

- World Wide Web

9 [1] Abbas, Alaa & Hussein, Qasim. (2020). Twitter Sentiment Analysis Using an Ensemble Majority Vote Classifier. Xinan Jiaotong Daxue XuebaoJournal of Southwest Jiaotong University. 55. 10.35741issn.0258-2724.55.1.9.

10 [2] Ahmad, Munir, et al. Machine learning techniques for sentiment analysis A review. Int. J. Multidiscip. Sci. Eng 8.3 (2017) 27.

11 [3] Alsaeedi, Abdullah, and Mohammad Zubair Khan. A study on sentiment analysis techniques of Twitter data. International Journal of Advanced Computer Science and Applications 10.2 (2019) 361-374.

12 [4] Ankit, & Saleena, Nabizath. (2018). An Ensemble Classification System for Twitter Sentiment Analysis. Procedia Computer Science. 132. 937-946. 10.1016j.procs.2018.05.109.

13 [5] Ankita Sharma, Udayan Ghose,Sentimental Analysis of Twitter Data with respect to General Elections in India,Procedia Computer Science,Volume 173,2020,Pages 325-334,ISSN 1877-0509,httpsdoi.org10.1016j.procs.2020.06.

14 [6] Asgarnezhad, Razieh & Monadjemi, S.A. & Soltanaghaei, M.. (2020). A High-Performance Model based on Ensembles for Twitter Sentiment Classification. 41-52. 10.22061JECEI.2020.7100.357.

15 [7] Bouazizi, Mondher & Ohtsuki, Tomoaki. (2017). A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter. IEEE Access. PP. 1-1. 10.1109ACCESS.2017.2740982.

16 [8] Budiharto, W., Meiliana, M. Prediction and analysis of Indonesia Presidential election from Twitter using sentiment analysis. J Big Data 5, 51 (2018).

17 [9] Çeliktuğ, Mestan. (2018). Twitter Sentiment Analysis, 3-Way Classification Positive, Negative or Neutral. 2098-2103. 10.1109BigData.2018.8621970.

18 [10] Da Silva, Nadia FF, Eduardo R. Hruschka, and Estevam R. Hruschka Jr. Tweet sentiment analysis with classifier ensembles. Decision Support Systems 66 (2014) 170-179.

19 [11] Desai, Mitali & Mehta, Mayuri. (2016). Techniques for sentiment analysis of Twitter data A comprehensive survey. 149-154. 10.1109CCAA.2016.7813707.

20 [12] Giachanou, Anastasia & Crestani, Fabio. (2016). Like It or Not: A Survey of Twitter Sentiment Analysis Methods. ACM Computing Surveys. 49. 1-41. 10.1145/2938640.

21 [13] Gull, Ratab & Shoaib, Umar & Rasheed, Saba & Abid, Washma & Zahoor, Beenish. (2016). Pre Processing of Twitter's Data for Opinion Mining in Political Context. Procedia Computer Science. 96. 1560-1570. 10.1016j.pro.

22 [14] Hasan, Ali & Moin, Sana & Karim, Ahmad & Band, Shahab. (2018). Machine Learning-Based Sentimental Analysis for Twitter Accounts. Mathematical and Computational Applications. 23. 11. 10.3390mca23010011.

23 [15] Hasan, Md & Maliha, Maisha & Arifuzzaman, M.. (2019). Sentiment Analysis with NLP on Twitter Data. 1-4. 10.1109IC4ME247184.2019.9036670.

24 [16] Ismail, Heba, Saad Harous, and Boumediene Belkhouche. A Comparative Analysis of Machine Learning Classifiers for Twitter Sentiment Analysis. Res. Comput. Sci. 110 (2016) 71-83.

25 [17] Kolchyna, Olga, et al. Twitter sentiment analysis Lexicon method, machine learning method and their combination. arXiv preprint arXiv1507.00955 (2015).

26 [18] Krouska, Akrivi & Troussas, Christos & Virvou, Maria. (2016). The effect of preprocessing techniques on Twitter sentiment analysis. 1-5. 10.1109IISA.2016.7785373.

27 [19] Lalji, T., and S. Deshmukh. Twitter sentiment analysis using hybrid approach. International Research Journal of Engineering and Technology 3.6 (2016) 2887-2890.

28 [20] Martinčić-Ipšić, Sanda & Miličić, Tanja & Todorovski, and. (2019). The Influence of Feature Representation of Text on the Performance of Document Classification. Applied Sciences. 9. 743. 10.3390app9040743.

29 [21] Pollacci, Laura & Sîrbu, Alina & Giannotti, Fosca & Pedreschi, Dino & Lucchese, Claudio & Muntean, Cristina. (2017). Sentiment Spreading An Epidemic Model for Lexicon-Based Sentiment Analysis on Twitter. 114-127.

30 [22] Sevim, Semih & Omurca, Sevinc & Ekinci, Ekin. (2018). An Ensemble Model using a BabelNet Enriched Document Space for Twitter Sentiment Classification. International Journal of Information Technology and Computer S.

31 [23] Singh, Tajinder & Kumari, Madhu. (2016). Role of Text Pre-processing in Twitter Sentiment Analysis.

Procedia Computer Science. 89. 549-554. 10.1016j.procs.2016.06.095.

32 [24] Yadollahi, Ali & Shahraki, Ameneh & Zaïane, Osmar. (2017). Current State of Text Sentiment Analysis from Opinion to Emotion Mining. ACM Computing Surveys. 50. 1-33. 10.11453057270.

33 [25] Zhao, Jianqiang & Gui, Xiaolin. (2017). Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis. IEEE Access. PP. 1-1. 10.1109ACCESS.2017.2672677.

34 [26] Jaiswal, Namrata, Prajakta Borkar, Prajakta Pirdurkar, Shivani Rahate, and S. V. Solanki. Polarity Detection using Twitter Sentiment Analysis.

35 [27] Chauhan, Priyavrat & Sharma, Nonita & Sikka, Geeta. (2021). The emergence of social media data and sentiment analysis in election prediction. Journal of Ambient Intelligence and Humanized Computing. 12. 10.1007s12.

-

35.2 Websites

- https://www.w3schools.com/
    - www.twitter.com
    - https://twython.readthedocs.io/en/latest/
    - https://textblob.readthedocs.io/en/dev/quickstart.html
-