

A  
Project Report On  
**Natural Language Processing**  
Mini Project  
**“Election Results Prediction by analyzing Tweets”**  
SUBMITTED TOWARDS THE  
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF  
Bachelor of Computer Engineering  
By

<b>Mr. Bhosale Bhushan Sunil</b>	<b>Roll No - 408</b>
<b>Mr. Kunjir Shubham Ganesh</b>	<b>Roll No - 457</b>
<b>Miss. Markad Prajakta Dhananjay</b>	<b>Roll No - 460</b>
<b>Miss. Salvi Neha Lilachand</b>	<b>Roll No - 478</b>

**Under The Guidance Of**  
**Prof. Jagtap V.G**



Department Of Computer Engineering  
SVPM's College Of Engineering, Malegaon(Bk.), Baramati, Pune-413115.

SAVITRIBAI PHULE PUNE UNIVERSITY

2022-23



SVPM's COLLEGE OF ENGINEERING,  
DEPARTMENT OF COMPUTER ENGINEERING

**CERTIFICATE**

This is to certify that the Project Entitled

**“Election Results Prediction by analyzing Tweets”**

Submitted by

Mr. Bhosale Bhushan Sunil

Roll No - 408

Mr. Kunjir Shubham Ganesh

Roll No - 457

Miss. Markad Prajakta Dhananjay

Roll No - 460

Miss. Salvi Neha Lilachand

Roll No - 478

is a bonafide work carried out by Students under the supervision of Prof Jagtap V.G and it is submitted towards the partial fulfillment of the requirement of Bachelor of Computer Engineering Project.

**Prof .Jagtap V.G**  
**Internal Guide**

**Dr.Sinkar Y.D**  
**H.O.D**

## Abstract

The Election Results Prediction project focuses on leveraging Natural Language Processing (NLP) techniques and machine learning algorithms to predict election outcomes by analysing tweets. With the increasing influence of social media platforms like Twitter in shaping public opinion, it becomes essential to harness the power of NLP to extract valuable insights from large volumes of user-generated content.

This report presents a comprehensive overview of the project, starting with the motivation behind the research and the need to predict election results using tweet analysis. The project's problem statement is defined, followed by the objectives and scope of the study. The system architecture is described, highlighting the different modules involved in collecting, pre-processing, and analysing the tweets.

To implement the Election Results Prediction system, several tools and datasets are required. Python, along with libraries like NLTK and Scikit-learn, is used for development. Access to the Twitter API enables the collection of relevant tweets, while an election-related dataset with labelled data is utilized for training and testing the sentiment analysis model.

The report also discusses the various features employed in the project. Text pre-processing features such as noise removal, tokenization, stop-word removal, and stemming/lemmatization are utilized to enhance data quality. Sentiment analysis features like Bag-of-Words, TF-IDF, N-grams, and sentiment lexicons aid in determining the sentiment of the tweets. Machine learning features, including feature selection, feature engineering, and model-based features, contribute to accurate sentiment classification and prediction.

An implementation screenshot showcasing the project's progress, such as a user interface displaying sentiment analysis results or the prediction module generating election result predictions, is included. Finally, the report concludes by highlighting the significance of the project's contributions to political analysis and decision-making based on real-time social media data.

Overall, the Election Results Prediction project demonstrates the potential of NLP techniques and machine learning algorithms in analysing tweets to predict election outcomes. By harnessing the power of sentiment analysis and feature extraction, this research aims to provide valuable insights into public sentiment and assist in making informed decisions in the context of elections.

## **Contents:**

- 1.Introduction
- 2.Problem Statement
- 3.Objectives and scope
- 4.System Architecture
- 5.Tools and Dataset required
- 6.Implementation screenshot
- 7.Conclusion

# **Introduction**

## **1.1 Motivation**

The motivation behind this project is to leverage Natural Language Processing (NLP) techniques and machine learning algorithms to predict election results by analysing tweets. With the growing influence of social media platforms, such as Twitter, in shaping public opinion, it becomes crucial to harness the power of NLP to extract valuable insights from large volumes of user-generated content.

## **1.2 Need of Problem**

The need for predicting election results using tweet analysis arises from the desire to gain real-time insights into public sentiment and gauge the popularity of political candidates. By analysing the sentiments expressed in tweets, we can identify trends, potential winners, and understand the factors influencing the outcome of an election.

# Problem Statement

## 2.1 Problem Statement

The objective of this project is to develop a system that can predict election results by analyzing tweets related to the candidates. The system will utilize NLP techniques and machine learning algorithms to classify tweets into sentiment categories and aggregate the results to generate predictions.

## 2.2 Features

8. In the Election Results Prediction project, various features play a crucial role in extracting valuable insights from tweets and predicting election outcomes. The following features are utilized in the project:

### 2.2.1 Text Preprocessing Features:

- Noise Removal: Removing irrelevant characters, symbols, URLs, and hashtags from the tweets to reduce noise and improve the quality of the data.
- Tokenization: Splitting the text into individual words or tokens to facilitate further analysis and feature extraction.
- Stop-word Removal: Eliminating common and non-informative words (e.g., "and," "the," "is") that do not contribute much to the sentiment analysis process.
- Stemming and Lemmatization: Reducing words to their base or root form (e.g., "running" to "run") to handle different word variations and improve feature representation.

### 2.2.2 Sentiment Analysis Features:

- Bag-of-Words (BoW): Representing tweets as a collection of words, disregarding grammar and word order, to create a numerical feature vector for sentiment analysis.
- Term Frequency-Inverse Document Frequency (TF-IDF): Assigning weights to words based on their frequency in the tweet and rarity in the entire dataset, helping to identify important and distinctive terms.
- N-grams: Considering sequences of adjacent words as features (e.g., "good" and "not good") to capture contextual information and improve sentiment classification accuracy.
- Sentiment Lexicons: Using pre-built sentiment lexicons (e.g., AFINN, Sent WordNet) that associate words with sentiment scores to determine the overall sentiment of a tweet.

### 2.2.3 Machine Learning Features:

- **Feature Selection:** Identifying the most informative and relevant features from the dataset to reduce dimensionality and enhance the performance of the machine learning model.
- **Feature Engineering:** Creating new features based on domain knowledge or specific characteristics of the tweets that may contribute to sentiment analysis and election result prediction.
- **Model-Based Features:** Extracting features from pre-trained language models (e.g., BERT, GPT) that capture semantic information and contextual understanding of the text.

These features collectively contribute to the accuracy and effectiveness of sentiment analysis and prediction models in the Election Results Prediction project. Proper selection, pre-processing, and utilization of these features are crucial for obtaining meaningful insights from tweets and making accurate predictions regarding election outcome

## 2.3 Objectives

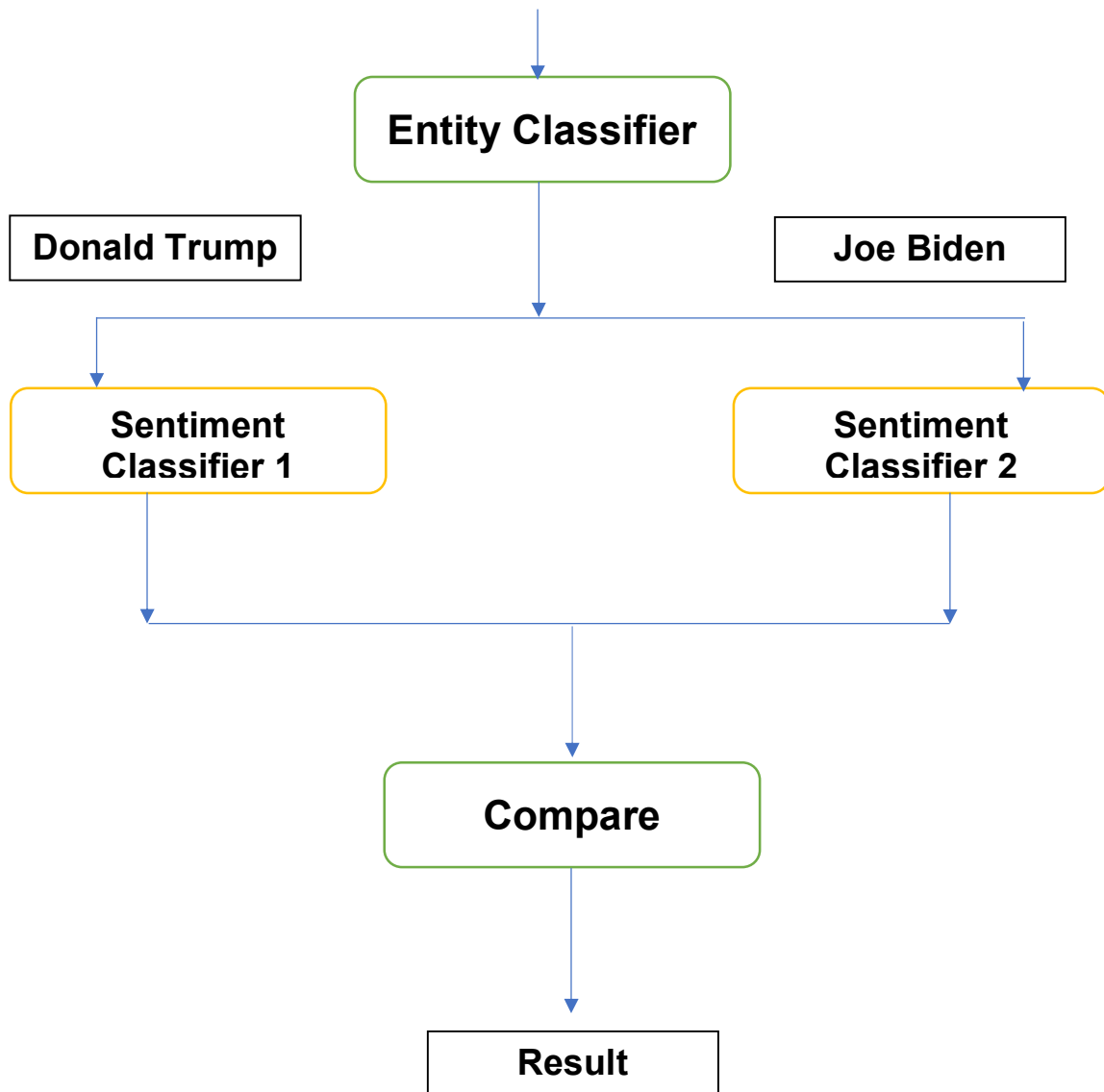
2. The main objectives of this project are as follows:
  - Collect a large dataset of tweets related to the election.
  - Pre-process the dataset by removing noise, handling text normalization, and feature extraction.
  - Train a machine learning model using the pre-processed data to classify tweets into sentiment categories (positive, negative, neutral).
  - Analyse the classified tweets to understand public sentiment towards candidates.
  - Utilize the sentiment analysis results to predict election outcomes.

## 2.4 Scope

The scope of this project includes:

- Developing a Python-based system for tweet collection, pre-processing, and sentiment analysis.
- Exploring and implementing various NLP techniques for feature extraction and sentiment classification.
- Evaluating the accuracy and performance of the trained machine learning model.
- Generating predictions for election results based on sentiment analysis.

## System Architecture





## Implementation screenshot

### Program Code:-

```
import pandas as pd
import numpy as np
import seaborn as sns
import textblob
import wordcloud
import matplotlib.pyplot as plt
from textblob import TextBlob
from wordcloud import WordCloud
import plotly.graph_objects as go
import plotly.express as px

trump_reviews = pd.read_csv("/Users/shubham/Desktop/NLP/Trumpall2.csv")
biden_reviews = pd.read_csv("/Users/shubham/Desktop/NLP/Bidenall2.csv")
print(trump_reviews.head())
print(biden_reviews.head())
textblob1 = TextBlob(trump_reviews["text"][10])
print("Trump :",textblob1.sentiment)
textblob2 = TextBlob(biden_reviews["text"][500])
print("Biden :",textblob2.sentiment)

def find_pol(review):
    return TextBlob(review).sentiment.polarity

trump_reviews["Sentiment Polarity"] = trump_reviews["text"].apply(find_pol)
print(trump_reviews.tail())
```

```
biden_reviews["Sentiment Polarity"] = biden_reviews["text"].apply(find_pol)
print(biden_reviews.tail())
```

```
trump_reviews["Expression Label"] = np.where(trump_reviews["Sentiment
Polarity"]>0, "positive", "negative")
trump_reviews["Expression Label"][trump_reviews["Sentiment
Polarity"]==0]="Neutral"
print(trump_reviews.tail())
```

```
biden_reviews["Expression Label"] = np.where(biden_reviews["Sentiment
Polarity"]>0, "positive", "negative")
```

```
biden_reviews["Expression Label"][trump_reviews["Sentiment
Polarity"]==0]="Neutral"
```

```
print(biden_reviews.tail())
```

```
#reviews1
```

```
reviews1 = trump_reviews[trump_reviews['Sentiment Polarity'] == 0.0000]
```

```
print(reviews1.shape)
```

```
cond1=trump_reviews['Sentiment Polarity'].isin(reviews1['Sentiment Polarity'])
```

```
trump_reviews.drop(trump_reviews[cond1].index, inplace = True)
```

```
print(trump_reviews.shape)
```

```
#reviews1
```

```
reviews2 = biden_reviews[biden_reviews['Sentiment Polarity'] == 0.0000]
```

```
print(reviews2.shape)
```

```
cond2=biden_reviews['Sentiment Polarity'].isin(reviews1['Sentiment Polarity'])
```

```
biden_reviews.drop(biden_reviews[cond2].index, inplace = True)
```

```
print(biden_reviews.shape)
```

```
# 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)
print(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)
print(df_subset_biden.shape)

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

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

Politicians = ['Joe Biden', 'Donald Trump']
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()
```

## Output:-

```

      user                                     text \
0  manny_rosen  @sanofi please tell us how many shares the Cr...
2  Patsyrw     Your AG Barr is as useless & corrupt as y...
3  seyedebrahimi_m  Mr. Trump! Wake Up! Most of the comments bel...
6  curtiswhitworth  Really? My prices have gone up. That's a fact.
8  CantStandya2018  How much on stocks have you made pushing mira...

Sentiment Polarity Expression Label
0          0.05          positive
2         -0.50          negative
3          0.50          positive
6          0.20          positive
8          0.35          positive

      user                                     text \
3  penblogger  @JoeBiden It's clear you didnt compose this tw...
5  FabrizioBenass4  @JoeBiden #ALL LIVES MATTER #HUMAN RIGHTS HELP...
13 candyisyummy333  @JoeBiden My standard reply to propaganda: htt...
14  5h0rtie      @JoeBiden @richardmarx Trouble is trump voted ...
15 shut_up_stewart  @JoeBiden I honestly cannot comprehend how a c...

Sentiment Polarity Expression Label
3          0.050000          positive
5          0.033333          Neutral
13         -0.050000          negative
14         -0.200000          Neutral
15          0.200000          Neutral

Trump : Sentiment(polarity=0.15, subjectivity=0.3125)
Biden : Sentiment(polarity=0.6, subjectivity=0.9)

      user                                     text \
2771 MickyLafferty  @realDonaldTrump Yeah, you're definitely afrai...
2773 connor182010  @realDonaldTrump Oh is little baby scared to b...
2782 Crystal60151335  @realDonaldTrump Absolutely!
2785 SpencerRossy  @realDonaldTrump I rarely get involved with fo...
2787 bjklinz      @realDonaldTrump I'm sorry, Donald. No. #POTUS

Sentiment Polarity Expression Label
2771         -0.6000          negative
2773         -0.1875          negative
2782          0.2500          positive
2785          0.2250          positive
2787         -0.5000          negative
```

	user	text \
2532	Gypsy0112	@JoeBiden The only you can do to calm the wat...
2534	virgil_merchant	@JoeBiden fire is the last thing our country n...
2535	meryn1977	@JoeBiden You'll just try to calm those waters...
2538	LeslyeHale	@JoeBiden Trump wants our children back at sch...
2539	rerickre	@JoeBiden ... and I know, because it's much co...

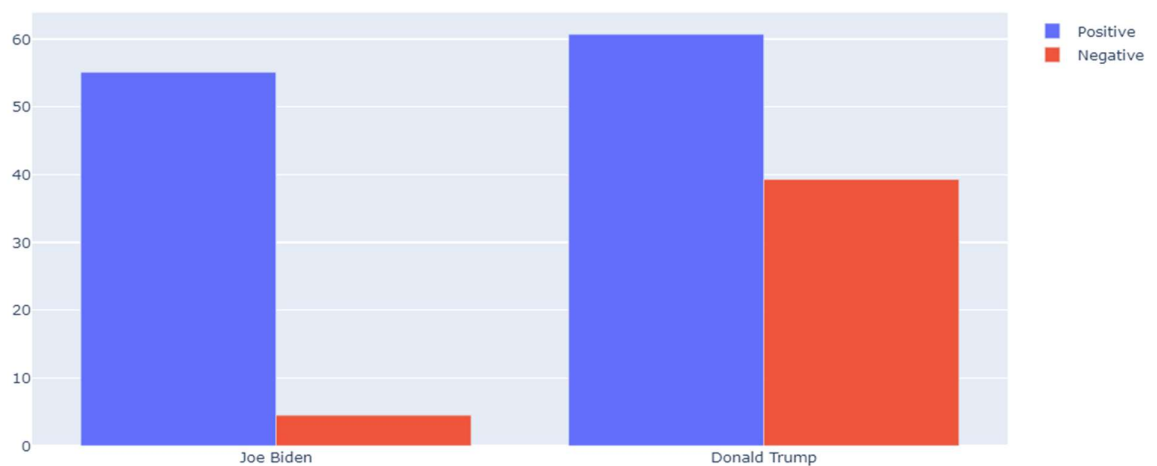
	Sentiment	Polarity	Expression	Label
2532	0.150000		positive	
2534	0.142857		positive	
2535	0.150000		positive	
2538	0.100000		positive	
2539	0.200000		positive	

	user	text \
2771	MickyLafferty	@realDonaldTrump Yeah, you're definitely afrai...
2773	connor182010	@realDonaldTrump Oh is little baby scared to b...
2782	Crystal60151335	@realDonaldTrump Absolutely!
2785	SpencerRossy	@realDonaldTrump I rarely get involved with fo...
2787	bjklinz	@realDonaldTrump I'm sorry, Donald. No. #POTUS

	Sentiment	Polarity	Expression	Label
2771	-0.6000		negative	
2773	-0.1875		negative	
2782	0.2500		positive	
2785	0.2250		positive	
2787	-0.5000		negative	

(1000, 4)  
(1000, 4)

	user	text	Sentiment	Polarity
Expression Label				
negative	449	449		449
positive	551	551		551
	user	text	Sentiment	Polarity
Expression Label				
negative	393	393		393
positive	607	607		607



## **Conclusion**

In conclusion, this project aims to predict election results by analyzing tweets using NLP techniques and machine learning algorithms. By harnessing the power of NLP and sentiment analysis, we can gain valuable insights into public sentiment and utilize that information for predicting election outcomes. The successful implementation of this project can contribute to the field of political analysis and assist in making informed decisions based on real-time social media data.