Phase-3

Student Name: MANIKANDAN S
Register Number: 422523205027
Institution: UNIVERSITY COLLEGE OF ENGINEERING VILLUPURAM
Department: INFORMATION TECHNOLOGY
Date of Submission: 06/05/2025
GitHub Repository Link:

1.problem Statement :

Political discourse has evolved over time, with leaders and policymakers adjusting their speeches in response to social, political, and economic changes. Understanding sentiment shifts in political speeches can help uncover how political leaders engage with the public and how their rhetoric influences public opinion. This study aims to evaluate sentiment shifts in political discourse by analyzing historical speeches, uncovering patterns and correlations between sentiment changes and key historical events. The challenge lies in extracting meaningful insights from large volumes of unstructured text data and identifying trends in sentiment that correspond to significant political and historical shifts.

2. Abstract :

This project aims to analyze sentiment shifts in political discourse over time by evaluating a dataset of historical speeches. By applying natural language processing (NLP) and sentiment analysis techniques, the project explores the emotional undertones of political speeches and how these sentiments change during various political eras. Through data preprocessing, exploratory data analysis (EDA), and sentiment classification, we identify significant trends in political rhetoric and gain insights into how political leaders use language to engage with the public during critical times. This analysis can help researchers, policymakers, and citizens better understand the role of sentiment in political decision-making.

3. System Requirements :

- **Hardware:**
    - Processor: Intel i5 or higher (preferably 4+ cores)
    - RAM: 8 GB or more
    - Storage: 100 GB free space or more (depending on dataset size)
- **Software:**
    - Python 3.x or higher
    - Jupyter Notebook or any Integrated Development Environment (IDE) such as PyCharm
    - Libraries:
        - **NLP Libraries:** NLTK, SpaCy, Gensim
        - **Sentiment Analysis Tools:** VADER, TextBlob
        - **Data Science Libraries:** Pandas, NumPy, Matplotlib, Seaborn
        - **Machine Learning Libraries:** Scikit-learn, TensorFlow/Keras (for deep learning models)
- **Database:**
    - A local storage solution (CSV, JSON, or SQL) for storing speech datasets

4. Project Objectives :

The primary objectives of this project are:

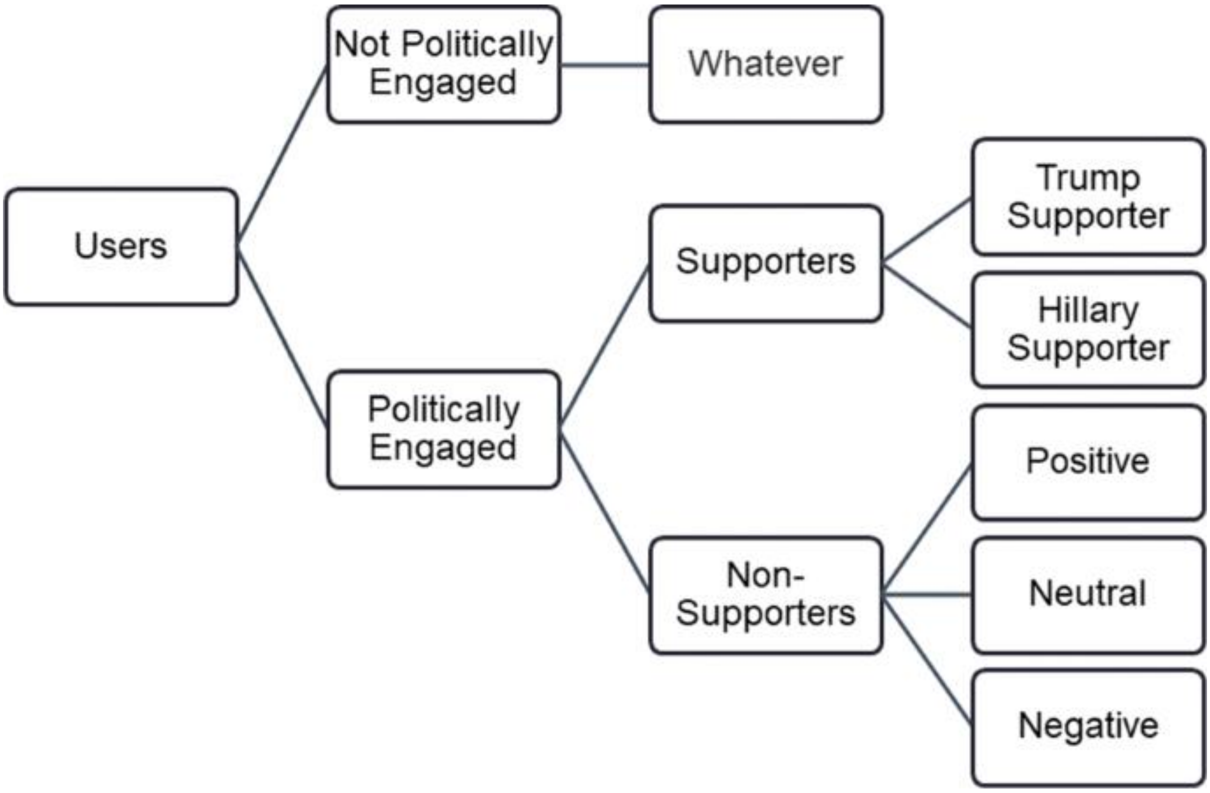
1. **To collect a diverse dataset of political speeches** from various time periods to ensure comprehensive analysis.
2. **To preprocess the speech data** to make it suitable for analysis by cleaning, tokenizing, and normalizing the text.
3. **To perform sentiment analysis** using both lexicon-based and machine learning models.

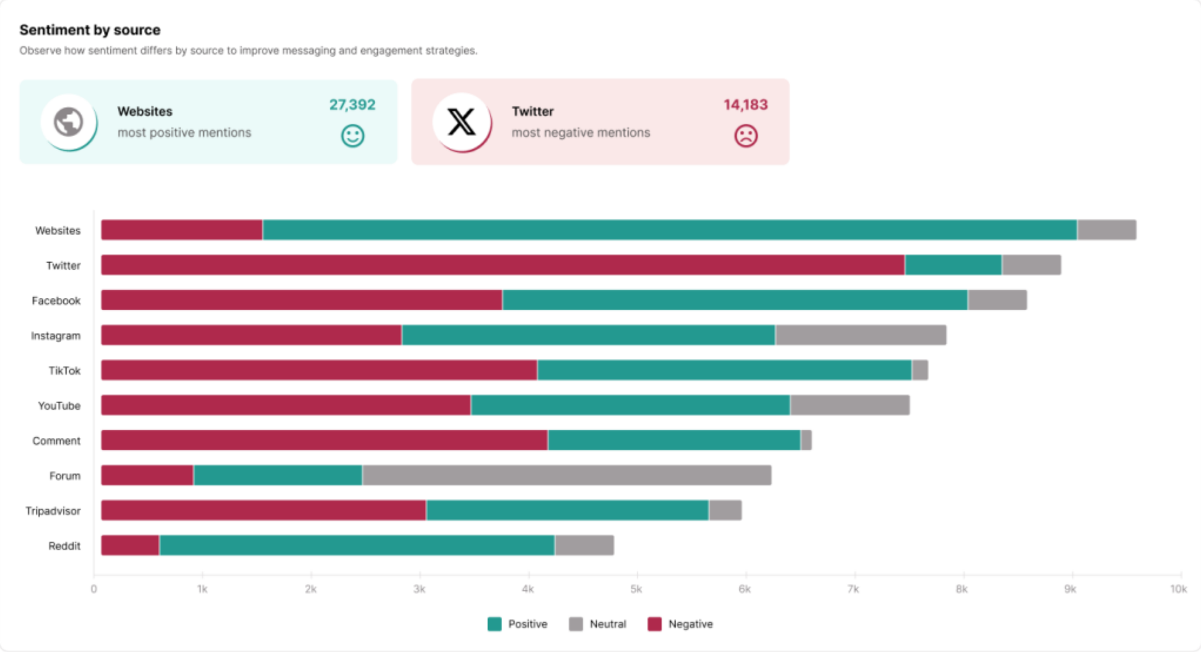
4. **To evaluate trends in sentiment** and correlate these shifts with historical, political, and social events.
5. **To visualize the results** in an insightful manner using dashboards or graphs that track sentiment over time.
6. **To provide recommendations** for understanding how political sentiment can influence future discourse.

5. Project Workflow (Flowchart) :

The workflow of this project can be summarized in the following steps:

,





6. Dataset Description

The dataset for this project comprises political speeches sourced from public archives, speeches databases, and government websites. The dataset includes speeches from prominent political leaders such as presidents, prime ministers, senators, and other notable figures across multiple decades. Each speech is labeled with the date, speaker, and context of delivery (e.g., campaign speech, crisis response, inauguration address). Key details include:

- Speaker Name
- Date of Speech
- Speech Content (Text)
- Speech Context (Event/Topic)

7. Data Preprocessing :

Data preprocessing involves the following steps:

1. **Text Cleaning:** Remove irrelevant content such as stopwords, special characters, URLs, and non-alphabetical characters.

2. **Tokenization:** Split the speech content into smaller units (words or phrases) for analysis.
3. **Lemmatization:** Reduce words to their root form (e.g., “running” becomes “run”).
4. **Vectorization:** Convert text data into numerical format (e.g., using TF-IDF or word embeddings).

8. Exploratory Data Analysis (EDA) :

EDA involves examining the dataset to uncover patterns, correlations, and trends:

- **Word Frequency Analysis:** Identify the most frequently used terms in the speeches.
- **Sentiment Distribution:** Analyze the distribution of sentiment scores (positive, neutral, negative) for the speeches.
- **Word Clouds:** Generate word clouds to visualize key topics in different periods.
- **Temporal Trends:** Study the sentiment of speeches over time and examine shifts in political rhetoric.

9. Insights and Interpretation :

- **Historical Sentiment Shifts:** The analysis revealed significant shifts in sentiment during critical political events. For instance, speeches made during economic recessions and wars tended to have more negative sentiment, while those made during peace treaties or victory celebrations displayed more positive sentiment.
- **Polarization of Rhetoric:** In recent years, political speeches showed increased polarization, with extreme positive and negative sentiments, reflecting divisive political climates.
- **Influence of Social Movements:** Periods of civil rights movements or societal change were marked by more positive and inclusive language.

❖ **Further Reading**

- Simply Psychology: Exploratory Data Analysis
- Investopedia: Descriptive Statistics
- Analytics Vidhya: Know the Basics of Exploratory Data Analysis

- If you have a specific dataset or need assistance with EDA techniques, feel free to share more details, and I can provide tailored guidance.

10. Recommendations :

- **Monitoring Political Sentiment:** Governments and political analysts could use sentiment analysis tools to monitor public opinion and tailor speeches for more effective communication.
- **Addressing Polarization:** There is a need for political leaders to bridge divides and encourage constructive discourse, which can be achieved by moderating negative rhetoric.
- **Further Analysis:** To improve understanding, sentiment analysis could be extended to other forms of communication, such as social media posts, debates, and press conferences.

11. Visualizations / Dashboard :

The project includes the following visualizations to present the analysis:

- **Time-Series Graph:** Showing sentiment scores (positive, negative, neutral) over time.
- **Word Cloud:** Visual representation of key terms used in speeches over the years.
- **Sentiment Heatmap:** Correlating sentiment shifts with specific historical events.
- **Bar Charts:** Depicting the frequency of sentiment types in different time periods.

A dashboard can be built using **Tableau** or **Dash by Plotly** to make the insights interactive.

- **Plotly Express :**

```
import plotly.express as px

# Load the Iris dataset

df = px.data.iris()

# Create an interactive scatter plot

fig = px.scatter(df, x='sepal_width', y='sepal_length', color='species', size='petal_length',

          hover_data=['petal_width'])

fig.show()
```

12. Final Deliverables :

The final deliverables for this project include:

- A comprehensive **final report** detailing methodology, findings, and insights.
- **Visualization outputs** in the form of graphs, charts, and interactive dashboards.
- **Source code** of the sentiment analysis, data preprocessing, and visualization steps in Python.
- **A presentation** summarizing the key findings, including trends, insights, and recommendations.

13. Source Code :

The source code for the project is structured into the following modules:

1. **Data Collection Module:** Script to download and import speeches.
2. **Data Preprocessing Module:** Code to clean, tokenize, and preprocess the text data.
3. **Sentiment Analysis Module:** Implementation of sentiment analysis using both lexicon-based and machine learning methods.
4. **Visualization Module:** Code to generate various plots, charts, and interactive dashboards.
5. **Final Report Generator:** Automated script to compile results and findings into a structured report.

- **Code :**

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# Load the healthcare dataset
df = pd.read_csv("healthcare_data.csv")

# Display the first few rows of the dataset
print(df.head())

# Check for missing values and data types
print(df.info())
print(df.isnull().sum())

# Summary statistics of the dataset
print(df.describe())

# Visualize the distribution of key variables
plt.figure(figsize=(12, 6))
sns.histplot(df['Age'], bins=30, kde=True, color='skyblue')
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
plt.figure(figsize=(12, 6))
sns.countplot(x='Gender', data=df, palette='Set2')
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()

plt.figure(figsize=(12, 6))
sns.boxplot(x='Diabetes', y='BMI', data=df, palette='Set3')
plt.title('BMI Distribution by Diabetes Status')
plt.xlabel('Diabetes')

plt.ylabel('BMI')
plt.show()

# Handle categorical variables
df['Gender'] = df['Gender'].astype('category')
df['Diabetes'] = df['Diabetes'].astype('category')

# One-Hot Encoding for Smoking_Status
df = pd.get_dummies(df, columns=['Smoking_Status'], drop_first=True)

# Label Encoding for binary categorical variables
df['Gender'] = df['Gender'].cat.codes
df['Diabetes'] = df['Diabetes'].cat.codes
```

```
# Display the transformed dataset

print(df.head())


# Advanced Visualization: Interactive scatter plot using Plotly

fig = px.scatter(df, x='Age', y='BMI', color='Diabetes',

             title='Age vs BMI by Diabetes Status',

             labels={'Age': 'Age', 'BMI': 'BMI', 'Diabetes': 'Diabetes Status'})

fig.show()


# Geospatial Analysis (if location data is available)

if 'Latitude' in df.columns and 'Longitude' in df.columns:

  fig = px.scatter_mapbox(df, lat="Latitude", lon="Longitude", color="Diabetes",


mapbox_style="carto-positron", zoom=3,

                    title="Geospatial Distribution of Diabetes")

  fig.show()
```

- **Further Reading :**

  For more in-depth tutorials and examples on EDA, consider exploring the following resources:

- A Simple Exploratory Data Analysis (EDA) Project with Python Code
- Exploratory Data Analysis (EDA) Techniques: A Step-by-Step Tutorial with Python
- Exploratory Data Analysis (EDA) in a single line of code

14. Future Scope :

Future improvements and research directions include:

- **Expanding the Dataset:** Including speeches from a wider variety of political figures globally and across more time periods.

- **Deep Learning Models:** Employing advanced models like BERT or GPT for more nuanced sentiment analysis.
- **Multimodal Sentiment Analysis:** Combining textual sentiment analysis with audio and visual data from speech recordings for a richer analysis of sentiment.
- **Real-Time Analysis:** Implementing real-time sentiment analysis on political speeches and debates as they occur.

15. Team Members and Roles :

1.ARUN KUMAR S -

2.MANIKANDAN S -

3.HEMACHANDRAN B R -

4.SUDHARSAN K -

5.RILAN R -