MACHINE LEARNING

TARUN BHATT

Contents

List of Figures:

Problem 1:

List of Tables:

Problem 1

Problem 1:

You are hired by one of the leading news channels CNBE who wants to analyze recent elections. This survey was conducted on 1525 voters with 9 variables. You have to build a model, to predict which party a voter will vote for on the basis of the given information, to create an exit poll that will help in predicting overall win and seats covered by a particular party.

Table 1: Data Description

Column Name	Description	Data Type
Vote	Party choice: Conservative or Labour	object
age	in years	int
economic.cond.national	Assessment of current national economic conditions, 1 to 5.	int
economic.cond.household	Assessment of current household economic conditions, 1 to 5.	int
Blair	Assessment of the Labour leader, 1 to 5.	int
Hague	Assessment of the Conservative leader, 1 to 5.	int
Europe:	an 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Eurosceptic' sentiment.	int
political.knowledge	Knowledge of parties' positions on European integration, 0 to 3.	int
Gender	female or male.	Object
Unnamed: 0		int

1.1 Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it.

Table 2: Head of dataset

	Unnamed: 0	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	1	Labour	43	3	3	4	1	2	2	female
1	2	Labour	36	4	4	4	4	5	2	male
2	3	Labour	35	4	4	5	2	3	2	male
3	4	Labour	24	4	2	2	1	4	0	female
4	5	Labour	41	2	2	1	1	6	2	male

Table 3: Tail of dataset

	Unnamed: 0	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
1520	1521	Conservative	67	5	3	2	4	11	3	male
1521	1522	Conservative	73	2	2	4	4	8	2	male
1522	1523	Labour	37	3	3	5	4	2	2	male
1523	1524	Conservative	61	3	3	1	4	11	2	male
1524	1525	Conservative	74	2	3	2	4	11	0	female

Checking for Null values:

Table 4: Null Value Counts

vote	0
age	0
economic.cond.national	0
economic.cond.household	0
Blair	0
Hague	0
Europe	0
political.knowledge	0
gender	0

dtype: int64

There are no Null values.

Checking for duplicate values:

Table 5: Duplicate Rows

Duplicate Rows:

	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
67	Labour	35	4	4	5	2	3	2	male
626	Labour	39	3	4	4	2	5	2	male
870	Labour	38	2	4	2	2	4	3	male
983	Conservative	74	4	3	2	4	8	2	female
1154	Conservative	53	3	4	2	2	6	0	female
1236	Labour	36	3	3	2	2	6	2	female
1244	Labour	29	4	4	4	2	2	2	female
1438	Labour	40	4	3	4	2	2	2	male

Count of Duplicate Rows: 8

Observation:

- There are 1525 Rows and 10 columns in the dataset.
- Removing the Unnamed: 0 column from the dataset as it is ineffectual.
- The count of Duplicate rows in the dataset is 8.
- Dropping the duplicates.
- Shape of Data Frame after dropping duplicates: (1517, 9)
- There are no Null values present in the dataset.

Table 6: Skewness for Numeric Columns

age	0.139800
economic.cond.national	-0.238474
economic.cond.household	-0.144148
Blair	-0.539514
Hague	0.146191
Europe	-0.141891
political.knowledge	-0.422928

dtype: float64

In summary:

- age has a slight right skew, indicating more voters are younger.
- economic.cond.national and economic.cond.household have slight left skews, showing a positive outlook on economic conditions.
- Blair has a significant left skew, indicating more positive ratings for Blair.
- Hague has a slight right skew, indicating more positive ratings for Hague.
- Europe has a slight left skew, indicating more positive opinions on Europe.
- political.knowledge has a significant left skew, suggesting lower levels of political knowledge among voters.

Vote Summary:

Labour	1063
Conservative	462

"Labour" appears more frequently than "Conservative", indicating that there are more instances of individuals who voted for the "Labour" party compared to the "Conservative" party.

Inference:

The dataset appears to be well-prepared for analysis, with no missing values. It includes information about respondents demographic characteristics, economic perceptions, assessments of political leaders, attitudes toward European integration, political knowledge, and gender. This dataset is suitable for various types of analysis, including predictive modeling to understand voting behaviour and potentially predict party choices in elections. Further analysis and modeling can provide valuable insights into the factors that influence voters decisions in political elections.

1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.

Exploratory Data Analysis:

Null Value Counts:

Table 7: Null value check

vote	0
age	0
economic.cond.national	0
economic.cond.household	0
Blair	0
Hague	0
Europe	0
political.knowledge	0
gender	0

There are no Null values in the dataset.

Data types of each column:

Table 8: Datatype for each column

vote	object
age	int64
economic.cond.national	int64
economic.cond.household	int64
Blair	int64
Hague	int64
Europe	int64
political.knowledge	int64
gender	object

Shape of the Dataset: (After removing duplicates)
(1517, 9)

Table 9: Data summary

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
count	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000
mean	54.241266	3.245221	3.137772	3.335531	2.749506	6.740277	1.540541
std	15.701741	0.881792	0.931069	1.174772	1.232479	3.299043	1.084417
min	24.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
25%	41.000000	3.000000	3.000000	2.000000	2.000000	4.000000	0.000000
50%	53.000000	3.000000	3.000000	4.000000	2.000000	6.000000	2.000000
75%	67.000000	4.000000	4.000000	4.000000	4.000000	10.000000	2.000000
max	93.000000	5.000000	5.000000	5.000000	5.000000	11.000000	3.000000

Univariate analysis:

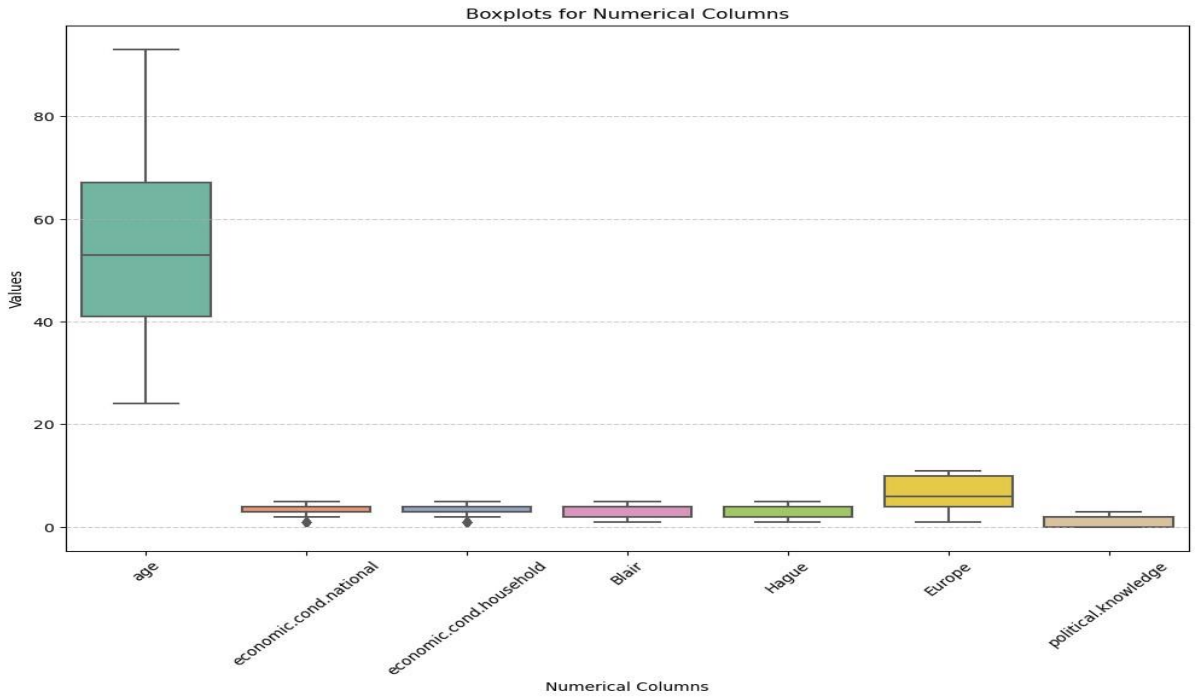
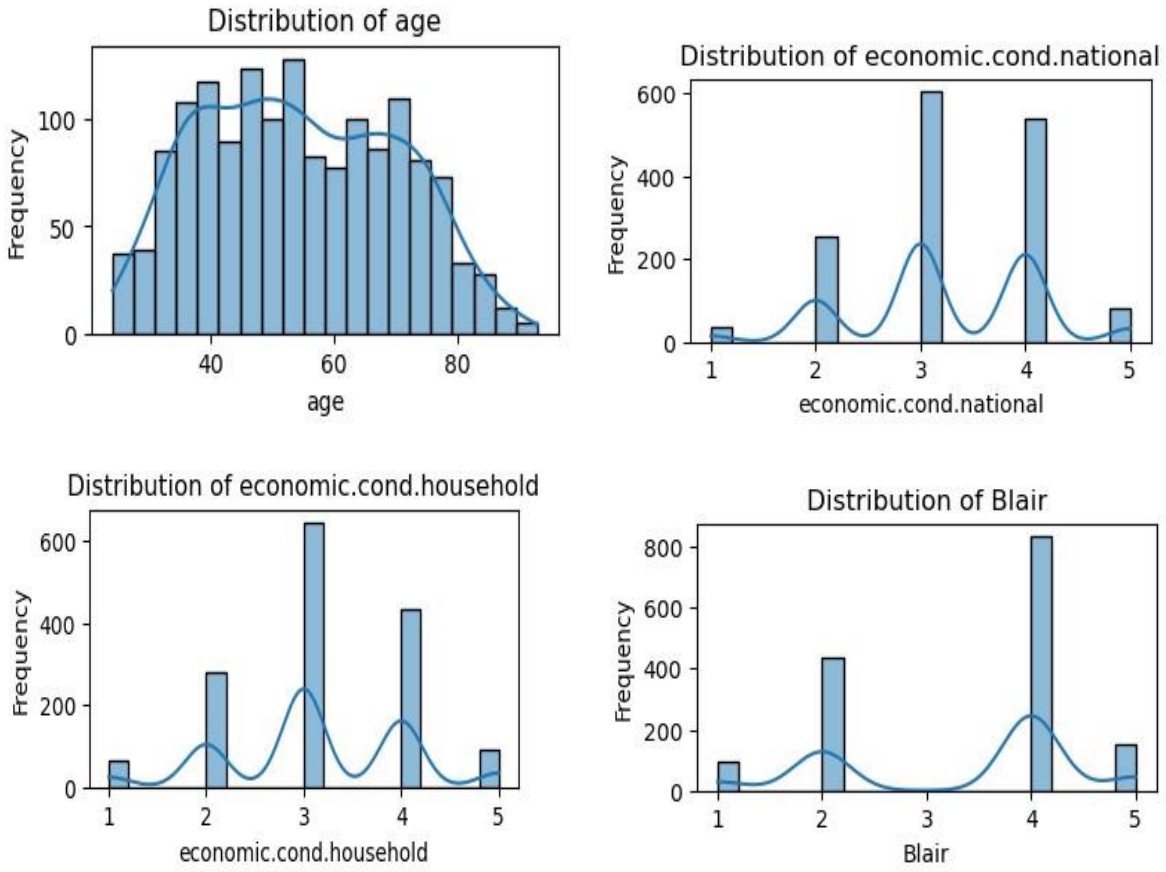


Figure 1: Boxplot for numerical variable



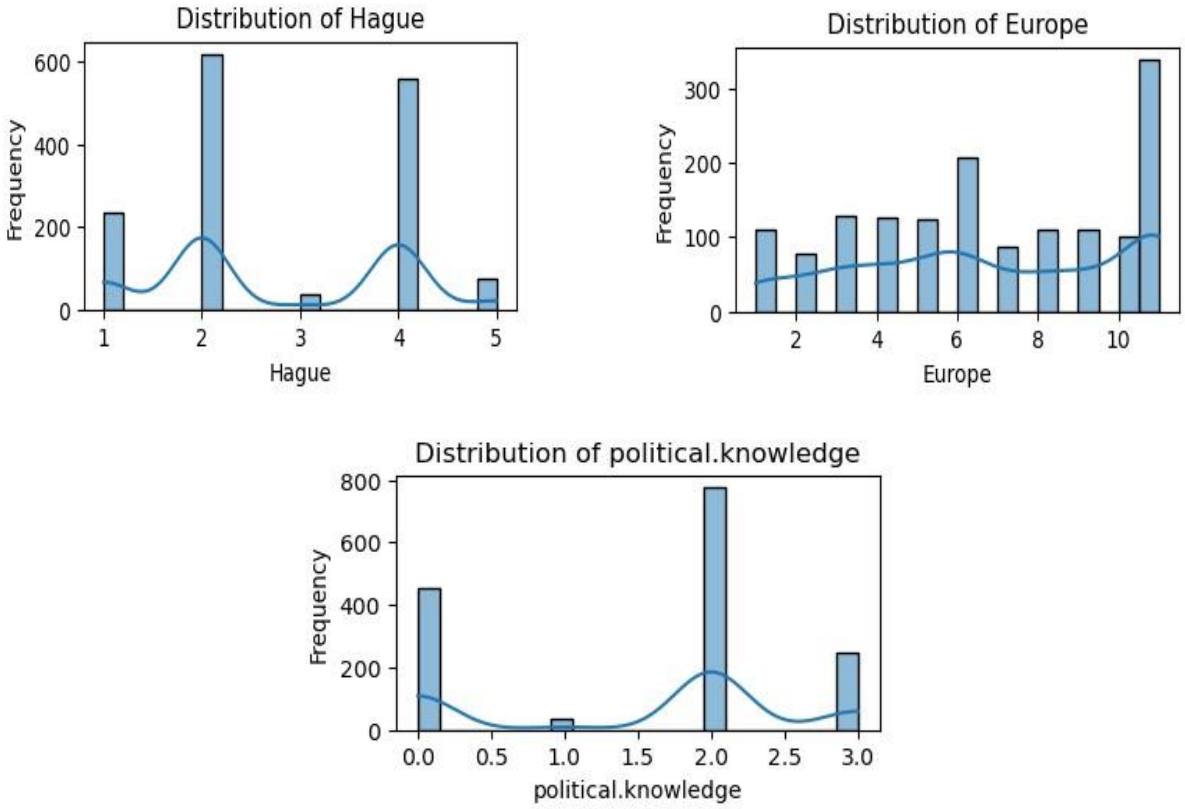
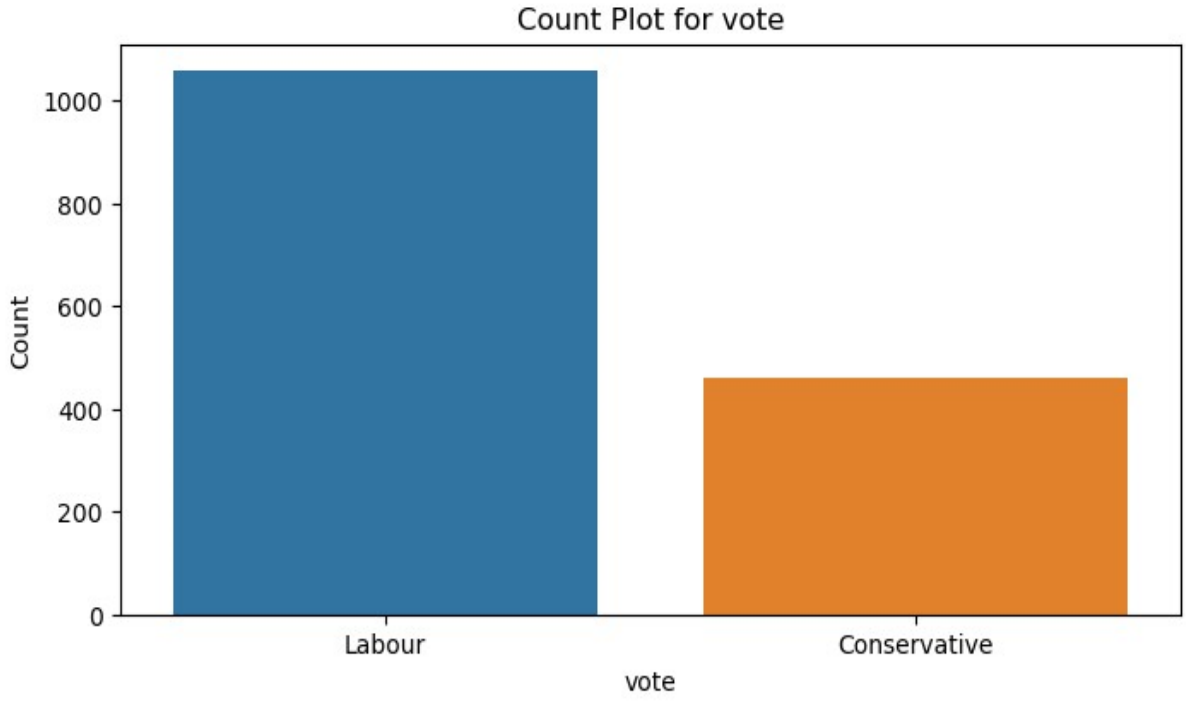


Figure 2: Hist plot for Numerical column

Count Plot:



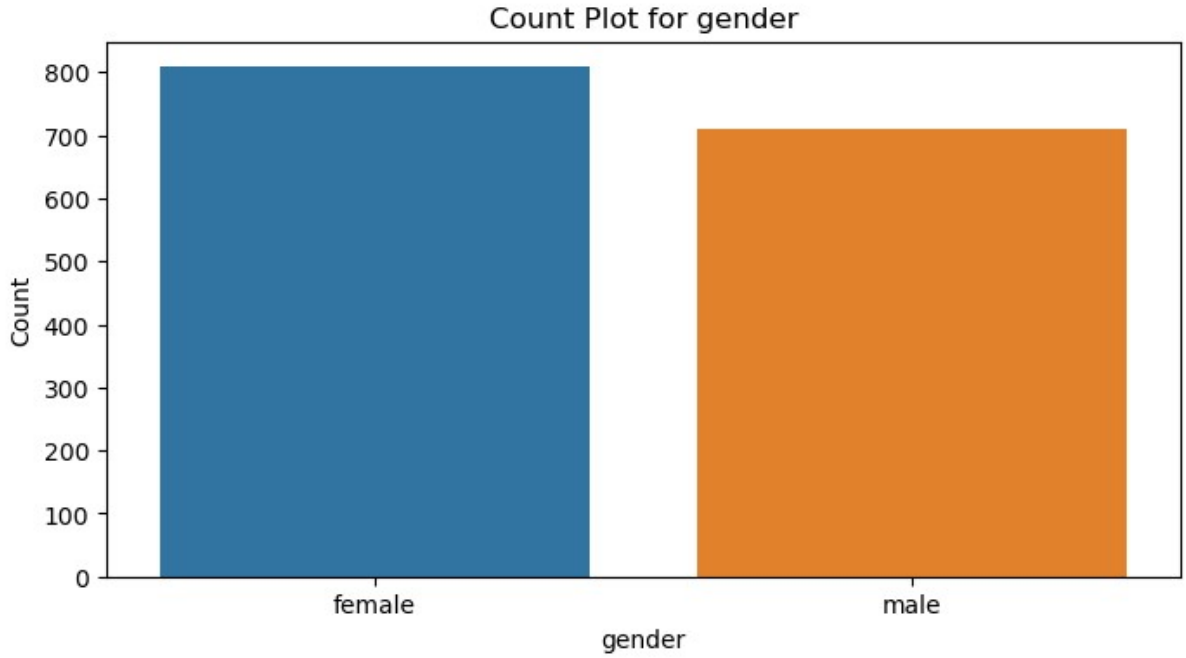
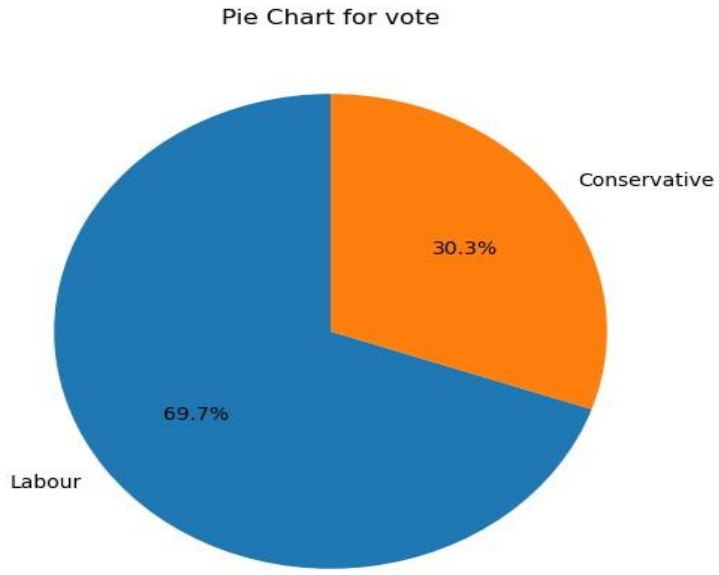


Figure 3: Count Plot for categorical variable- vote and gender

Pie Chart:



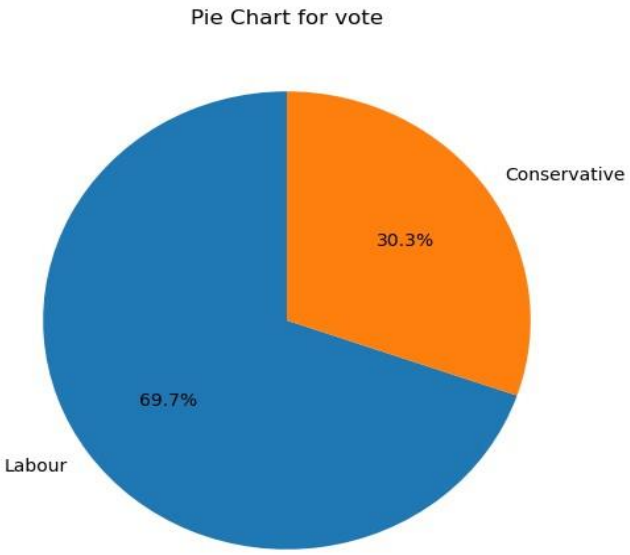


Figure 4: Pie chart for vote and gender column

Bivariate Analysis:

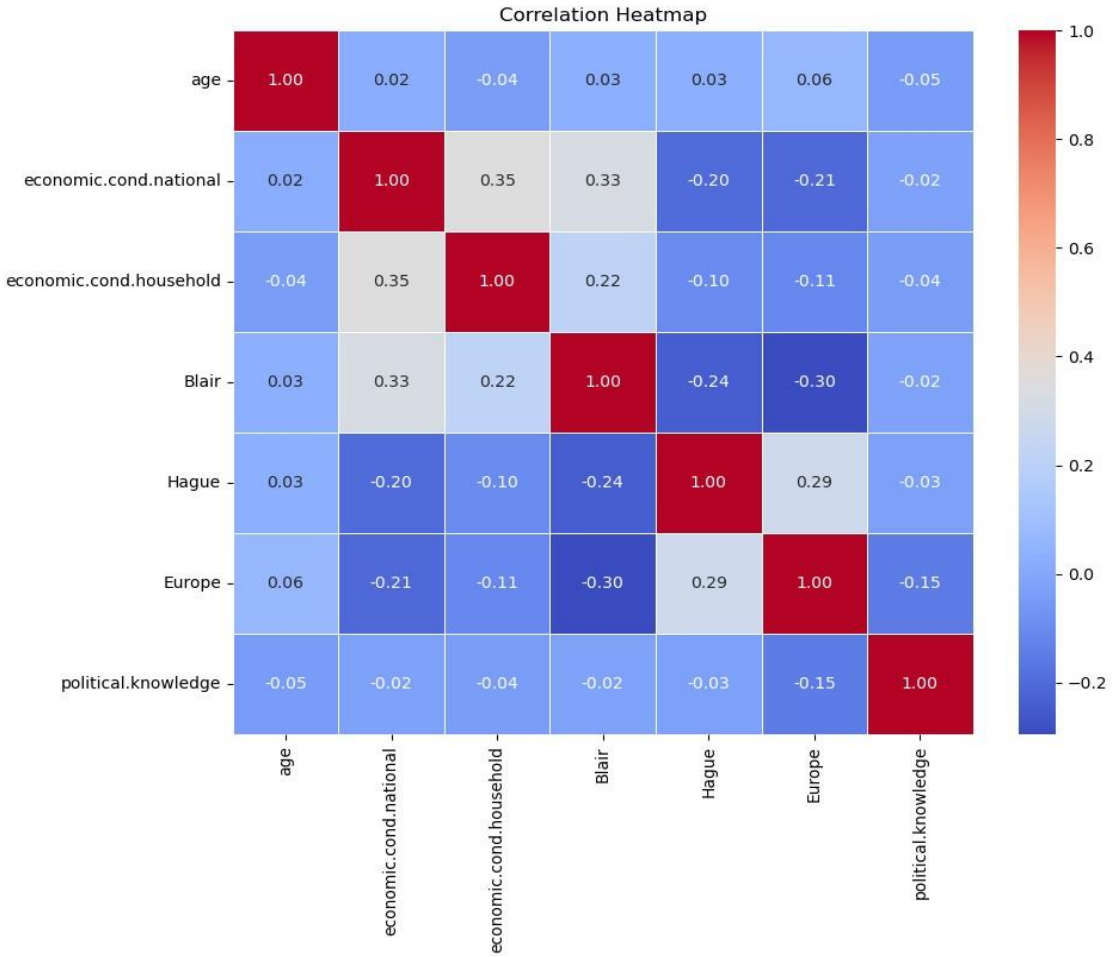


Figure 5: Correlation Heatmap

Observation:

- From the correlation matrix we concluded that there is less correlation in the dataset. There are some variables which are moderately positively and negatively correlated.
- ‘Blair’ with 'economic.cond.national' and ‘economic.cond.household’, ‘Europe’ with ‘Hague’, ‘economic.cond.national’ with ‘economic.cond.household’ have moderate positive correlation. have moderate positive correlation.
- 'Hague' with 'economic.cond.national' and 'Blair', 'Europe' with 'economic.cond.national' and 'Blair' have moderate negative correlation.

Figure 6: Pair plot

Observation:

From the above plot it can be seen that Blair, political.knowledge and Europe are slightly left skewed and all other variables are almost normally distributed.

Outlier detection and treatment:

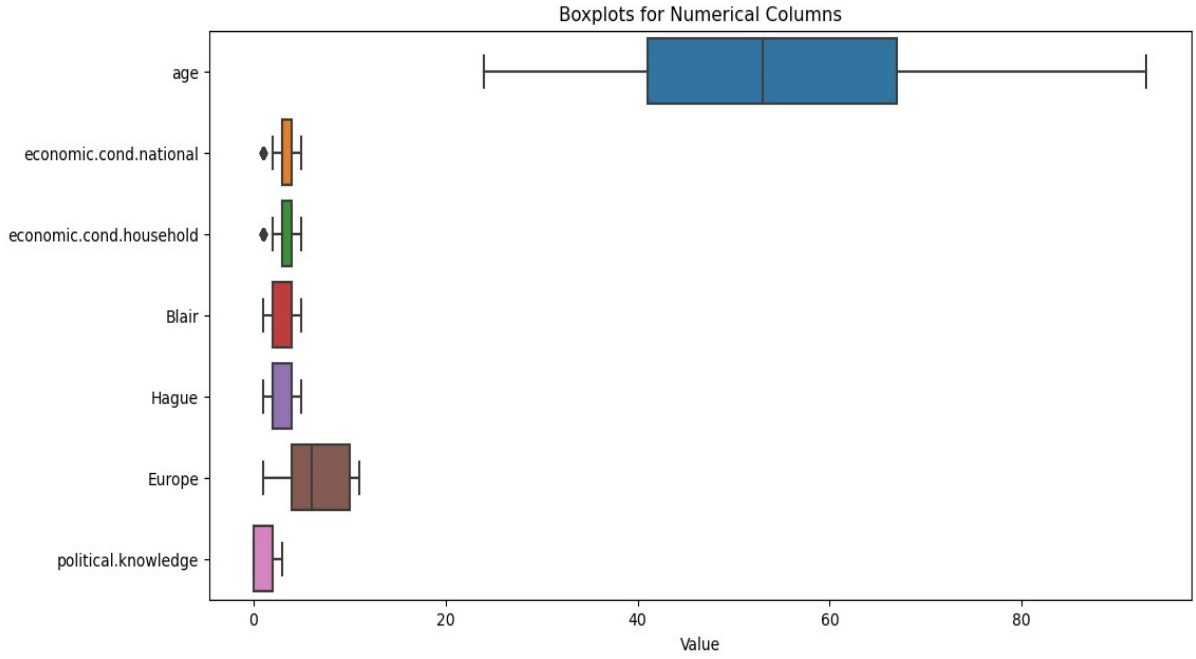


Figure 7: Boxplot

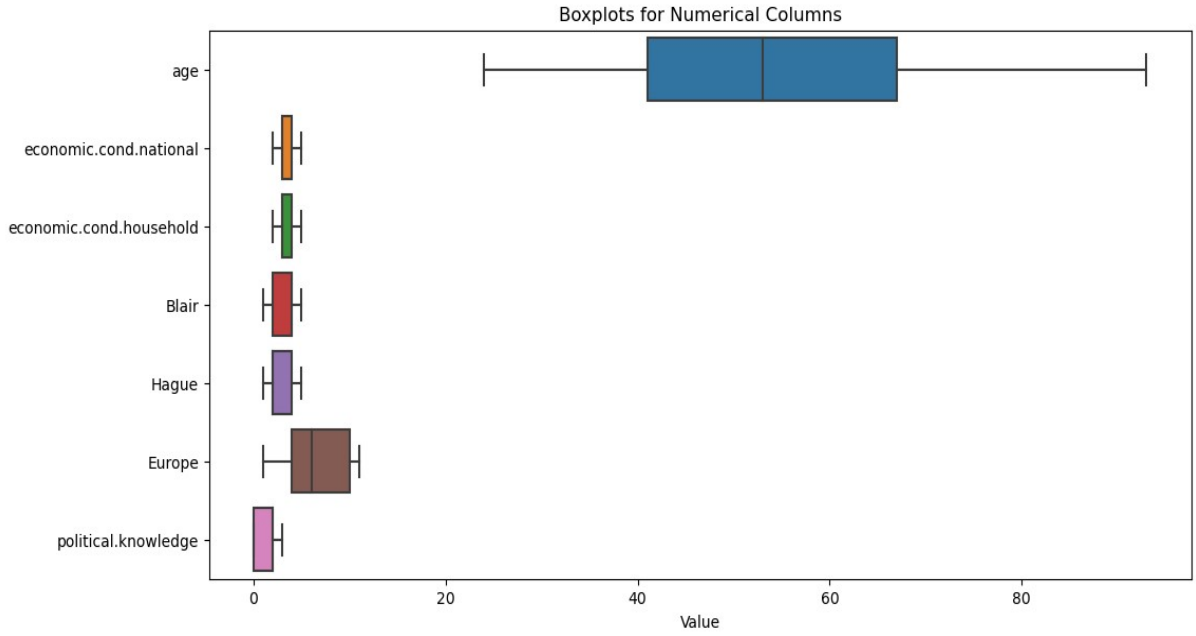


Figure 8: Boxplot post outlier trratment

1.3. Encode the data (having string values) for Modelling. Data Split: Split the data into test and train (30:70). Apply Linear regression. Performance Metrics: Check the performance of Predictions on Train and Test sets using R-square, RMSE.

Encoding:

One- hot encoding is performed in the dataset.

Train-test split:

Data Pre-processing Defining Target Variable and Features:

- Our target variable, denoted as 'y,' represents the outcome we want to predict. In this project, 'y' corresponds to the 'vote' column in our dataset. The 'vote' column is indicative of the political party a person voted for in an election.
- Features 'X' are the input variables or attributes that we will utilize to make predictions. For this project, we have considered all the columns in our dataset except for 'vote' as our features.

Test Size and Random State:

Training set: 70% of our dataset.

Testing set: 30% of our dataset

Shape of Train- test split:

Shape of X_train: (1061, 8)

Shape of y_train: (1061,)

Shape of X_test: (456, 8)

Shape of y_test: (456,)

Scaling of dataset:

Min- max scaling is performed.

Scaling was performed to standardize the numerical features, ensuring that the machine learning models can effectively learn from the data without being biased or sensitive to the scale of individual features. This pre-processing step contributes to improved model performance and interpretability.

1.4 Apply Logistic Regression and LDA (linear discriminant analysis).

Logistic Regression Model:

Logistic Regression Accuracy: 0.8333333333333334

Table 10: Logistic Regression Classification Report

	precision	recall	f1-score	support
Conservative	0.74	0.60	0.66	125
Labour	0.86	0.92	0.89	331
accuracy			0.83	456
macro avg	0.80	0.76	0.78	456
weighted avg	0.83	0.83	0.83	456

linear Discriminant Model:

LDA Accuracy: 0.831140350877193

Table 11: LDA classification report

LDA Classification Report:

	precision	recall	f1-score	support
Conservative	0.72	0.62	0.67	125
Labour	0.86	0.91	0.89	331
accuracy			0.83	456
macro avg	0.79	0.77	0.78	456
weighted avg	0.83	0.83	0.83	456

Observation and validness:

Both the Logistic Regression and Linear Discriminant Analysis (LDA) models performed similarly on the classification task.

- Logistic Regression: Achieved an accuracy of approximately 83.33%. It showed good precision, recall, and F1-score for class 1, indicating it's effective at correctly identifying one of the classes. It's a straightforward and interpretable model.
- Linear Discriminant Analysis (LDA): Achieved an accuracy of approximately 83.11%, very close to Logistic Regression. LDA also demonstrated good precision, recall, and F1-score for class 1. It's a linear classification technique that aims to maximize class separability.

Both models provide reasonable classification results for the given dataset, and the choice between them may depend on factors like ease of interpretation and specific project requirements. Further fine-tuning or exploring other models may lead to even better performance if necessary.

Train and Test accuracy for both models:

Logistic Regression Training Accuracy: 0.8350612629594723

Logistic Regression Testing Accuracy: 0.8333333333333334

LDA Training Accuracy: 0.8331762488218661

LDA Testing Accuracy: 0.831140350877193

Validness of the models:

Both the Logistic Regression and LDA models appear to be valid as they exhibit consistent performance on both the training and testing datasets.

There is no clear evidence of overfitting or underfitting, as the training and testing accuracies are similar and relatively high (around 83%).

These models seem to generalize well to unseen data, making them valid for predicting party choices in the election dataset

1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results.

KNN Accuracy: 0.8245614035087719

Table 12: KNN Classification Report

KNN Classification Report:

	precision	recall	f1-score	support
Conservative	0.69	0.65	0.67	125
Labour	0.87	0.89	0.88	331
accuracy			0.82	456
macro avg	0.78	0.77	0.78	456
weighted avg	0.82	0.82	0.82	456

Naïve Bayes Accuracy: 0.8442982456140351

Table 13: Naïve Bayes Classification Report

Naïve Bayes Classification Report:

	precision	recall	f1-score	support
Conservative	0.73	0.69	0.71	125
Labour	0.88	0.90	0.89	331
accuracy			0.84	456
macro avg	0.81	0.80	0.80	456
weighted avg	0.84	0.84	0.84	456

- K-Nearest Neighbors (KNN) Training Accuracy: 0.31950989632422244
  K-Nearest Neighbors (KNN) Testing Accuracy: 0.8245614035087719

- Naïve Bayes Training Accuracy: 0.6842601319509897
  Naïve Bayes Testing Accuracy: 0.7258771929824561

- The K-Nearest Neighbors (KNN) model shows signs of overfitting, with a large gap between its high training accuracy and lower testing accuracy. This suggests that the KNN model may be too complex for the data and struggles to generalize to unseen samples.
- The Naïve Bayes model has a smaller gap between training and testing accuracies compared to KNN, indicating less overfitting. However, it still exhibits some degree of overfitting, as the training accuracy is notably higher than the testing accuracy.

1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting.

Performed Model Tuning:
Best Logistic Regression Accuracy: 0.831140350877193

Best Logistic Regression Classification Report:

Table 14:Best Logistic Regression Classification Report

Best Logistic Regression Classification Report:

	precision	recall	f1-score	support
Conservative	0.74	0.59	0.66	125
Labour	0.86	0.92	0.89	331
accuracy			0.83	456
macro avg	0.80	0.76	0.77	456
weighted avg	0.82	0.83	0.82	456

Best K-Nearest Neighbors (KNN) Accuracy: 0.8267543859649122

Table 15: Best k-Nearest Neighbors Classification Report

Best K-Nearest Neighbors (KNN) Classification Report:

	precision	recall	f1-score	support
Conservative	0.69	0.66	0.68	125
Labour	0.88	0.89	0.88	331
accuracy			0.83	456
macro avg	0.78	0.78	0.78	456
weighted avg	0.82	0.83	0.83	456

In our analysis, we have compared two models, Logistic Regression and K-Nearest Neighbors (KNN), for classifying political party preferences.

The best Logistic Regression model achieved an accuracy of approximately 83.11%, with slightly better performance in recognizing the "Labour" class.

The best KNN model achieved an accuracy of around 82.68% and performed well in terms of F1-score for both classes.

Both models offer reasonably good predictive performance. The choice between them may depend on other considerations such as computational efficiency and interpretability.

Bagging:

Random Forest Accuracy: 0.8355263157894737

Table 16: Random Forest Classification Report

	precision	recall	f1-score	support
Conservative	0.74	0.62	0.68	125
Labour	0.87	0.92	0.89	331
accuracy			0.84	456
macro avg	0.80	0.77	0.78	456
weighted avg	0.83	0.84	0.83	456

The Random Forest model achieved an accuracy of 83.55% and demonstrated good precision, recall, and F1-score for both Conservative and Labour classes. This makes it a strong candidate for solving the classification problem in the business context.

AdaBoost Accuracy: 0.8289473684210527

Table 17: AdaBoost Classification Report

	precision	recall	f1-score	support
Conservative	0.70	0.65	0.68	125
Labour	0.87	0.90	0.88	331
accuracy			0.83	456
macro avg	0.79	0.77	0.78	456
weighted avg	0.83	0.83	0.83	456

Best Hyperparameters for Random Forest:

{'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 100}

Best Hyperparameters for AdaBoost:

{'base_estimator': DecisionTreeClassifier(max_depth=2), 'learning_rate': 0.1, 'n_estimators': 50}

Observation:

Overall, AdaBoost achieved an accuracy of 82.89% on the test data. It shows promising results in terms of precision, recall, and F1-score for both Conservative and Labour classes. This boosted model is another viable option for your classification task.

Here's a brief comparison on the performances of the models:

Logistic Regression:
Accuracy: 83.33% Good balance between precision and recall for both classes. Suitable for a quick, interpretable model.

Linear Discriminant Analysis (LDA):
Accuracy: 83.11% Similar to Logistic Regression in performance. Useful when assumptions of linearity hold.

K-Nearest Neighbors (KNN):
Accuracy: 82.46% Lower accuracy compared to Logistic Regression and LDA. Sensitive to the choice of the number of neighbors.

Naïve Bayes:
Accuracy: 84.43% Good accuracy and balanced precision and recall. Performs well for text-based classification tasks.

Random Forest (Bagging):
Accuracy: 83.55% Balanced performance with good accuracy. Handles non-linearity and feature importance.

AdaBoost (Boosting):
Accuracy: 82.89% Slightly lower accuracy than Random Forest. Benefits from combining weak learners and improving their performance.

Overall, Naïve Bayes and Random Forest stand out as strong performers with accuracy exceeding 84%. Logistic Regression and LDA offer a good balance between simplicity and performance. KNN lags slightly behind due to sensitivity to hyperparameters. AdaBoost, while competitive, is not the highest-performing model in this scenario.

1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized.

Model: Logistic Regression

Train Accuracy: 0.83
Test Accuracy: 0.83

Train Confusion Matrix:

```
[[228 107]
 [ 69 657]]
```

Test Confusion Matrix:

```
[[ 73  52]
 [ 25 306]]
```

Table 18: Logistic Regression Train Classification Report

Train Classification Report:	precision	recall	f1-score	support
Conservative	0.77	0.68	0.72	335
Labour	0.86	0.90	0.88	726
accuracy			0.83	1061
macro avg	0.81	0.79	0.80	1061
weighted avg	0.83	0.83	0.83	1061

Table 19: Logistic Regression Test Classification Report

Test Classification Report:	precision	recall	f1-score	support
Conservative	0.74	0.58	0.65	125
Labour	0.85	0.92	0.89	331
accuracy			0.83	456
macro avg	0.80	0.75	0.77	456
weighted avg	0.82	0.83	0.82	456

Model: LDA

Train Accuracy: 0.83
Test Accuracy: 0.83

Train Confusion Matrix:

```
[[236  99]
 [ 78 648]]
```

Test Confusion Matrix:

```
[[ 78  47]
 [ 30 301]]
```

Table 20: LDA Train Classification Report

Train Classification Report:	precision	recall	f1-score	support
Conservative	0.75	0.70	0.73	335
Labour	0.87	0.89	0.88	726
accuracy			0.83	1061
macro avg	0.81	0.80	0.80	1061
weighted avg	0.83	0.83	0.83	1061

Table 21: LDA Test Classification Report

Test Classification Report:

	precision	recall	f1-score	support
Conservative	0.72	0.62	0.67	125
Labour	0.86	0.91	0.89	331
accuracy			0.83	456
macro avg	0.79	0.77	0.78	456
weighted avg	0.83	0.83	0.83	456

Model: KNN

Train Accuracy: 0.87
Test Accuracy: 0.83

Train Confusion Matrix:

```
[[261  74]
 [ 66 660]]
```

Test Confusion Matrix:

```
[[ 83  42]
 [ 37 294]]
```

Table 22: KNN Train Classification Report

Train Classification Report:

	precision	recall	f1-score	support
Conservative	0.80	0.78	0.79	335
Labour	0.90	0.91	0.90	726
accuracy			0.87	1061
macro avg	0.85	0.84	0.85	1061
weighted avg	0.87	0.87	0.87	1061

Table 23: KNN Test Classification Report

Test Classification Report:

	precision	recall	f1-score	support
Conservative	0.69	0.66	0.68	125
Labour	0.88	0.89	0.88	331
accuracy			0.83	456
macro avg	0.78	0.78	0.78	456
weighted avg	0.82	0.83	0.83	456

Model: Naïve Bayes

Train Accuracy: 0.83
Test Accuracy: 0.84

Train Confusion Matrix:

```
[[234 101]
 [ 80 646]]
```

Test Confusion Matrix:

```
[[ 86  39]
 [ 32 299]]
```

Table 24: Model Train Classification Report

Train Classification Report:

	precision	recall	f1-score	support
Conservative	0.75	0.70	0.72	335
Labour	0.86	0.89	0.88	726
accuracy			0.83	1061
macro avg	0.81	0.79	0.80	1061
weighted avg	0.83	0.83	0.83	1061

Table 25: Model Test Classification Report

Test Classification Report:

	precision	recall	f1-score	support
Conservative	0.73	0.69	0.71	125
Labour	0.88	0.90	0.89	331
accuracy			0.84	456
macro avg	0.81	0.80	0.80	456
weighted avg	0.84	0.84	0.84	456

Model: Random Forest

Train Accuracy: 0.89
Test Accuracy: 0.83

Train Confusion Matrix:

```
[[258  77]
 [ 37 689]]
```

Test Confusion Matrix:

```
[[ 76  49]
 [ 27 304]]
```

Table 26: Random Forest Train Classification Report

Train Classification Report:

	precision	recall	f1-score	support
Conservative	0.87	0.77	0.82	335
Labour	0.90	0.95	0.92	726
accuracy			0.89	1061
macro avg	0.89	0.86	0.87	1061
weighted avg	0.89	0.89	0.89	1061

Table 27: Random Forest Test Classification Report

Test Classification Report:

	precision	recall	f1-score	support
Conservative	0.74	0.61	0.67	125
Labour	0.86	0.92	0.89	331
accuracy			0.83	456
macro avg	0.80	0.76	0.78	456
weighted avg	0.83	0.83	0.83	456

Model: AdaBoost

Train Accuracy: 0.86
Test Accuracy: 0.85

Train Confusion Matrix:

```
[[247  88]
 [ 63 663]]
```

Test Confusion Matrix:

```
[[ 85  40]
 [ 29 302]]
```

Table 28: AdaBoost Train Classification Report

Train Classification Report:

	precision	recall	f1-score	support
Conservative	0.80	0.74	0.77	335
Labour	0.88	0.91	0.90	726
accuracy			0.86	1061
macro avg	0.84	0.83	0.83	1061
weighted avg	0.86	0.86	0.86	1061

Table 29: AdaBoost Test Classification Report

Test Classification Report:

	precision	recall	f1-score	support
Conservative	0.75	0.68	0.71	125
Labour	0.88	0.91	0.90	331
accuracy			0.85	456
macro avg	0.81	0.80	0.80	456
weighted avg	0.85	0.85	0.85	456

In summary, all models performed well on both the train and test sets, with test accuracies ranging from 0.83 to 0.85. The Random Forest model achieved the highest train accuracy of 0.89, while AdaBoost achieved the highest test accuracy of 0.85.

ROC Curve:

1. **Logistic Regression**

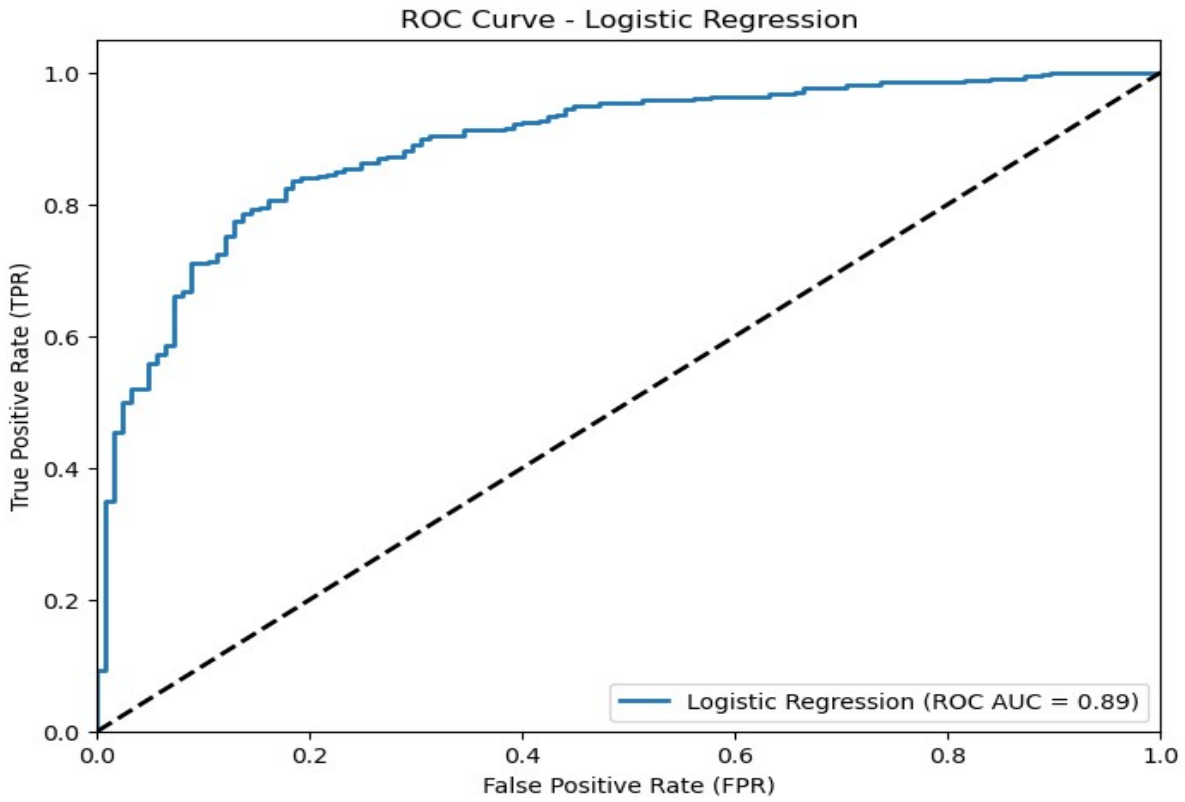


Figure 9: ROC Curve of logistic Regression

2. LDA:

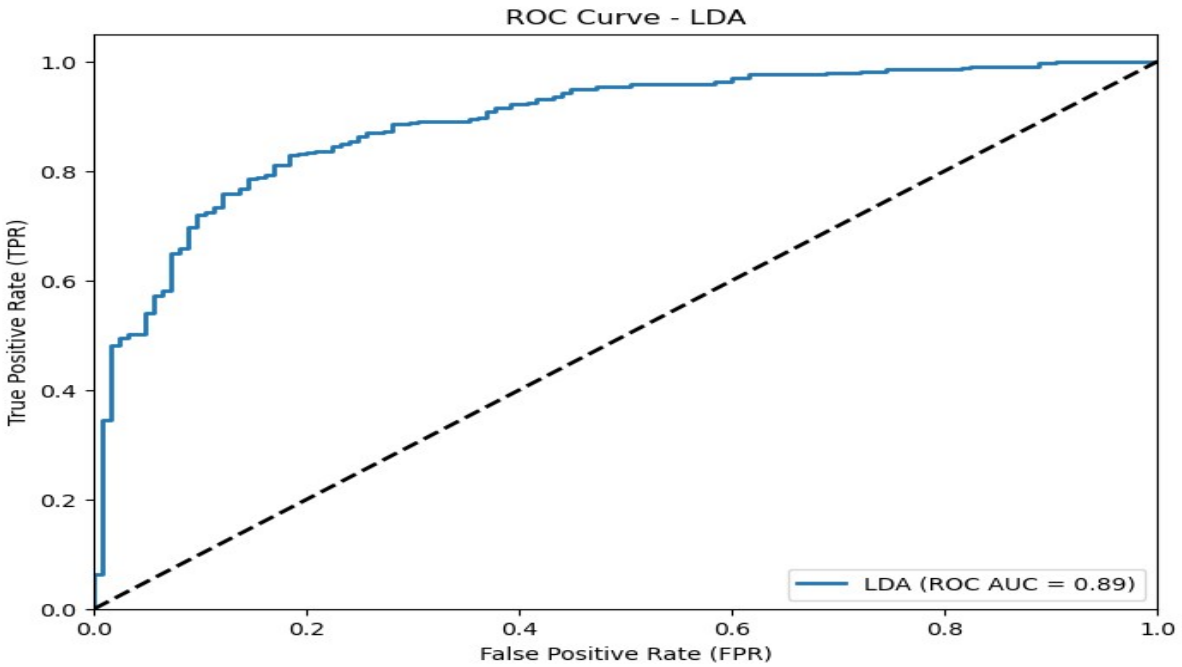


Figure 10: ROC Curve- LDA

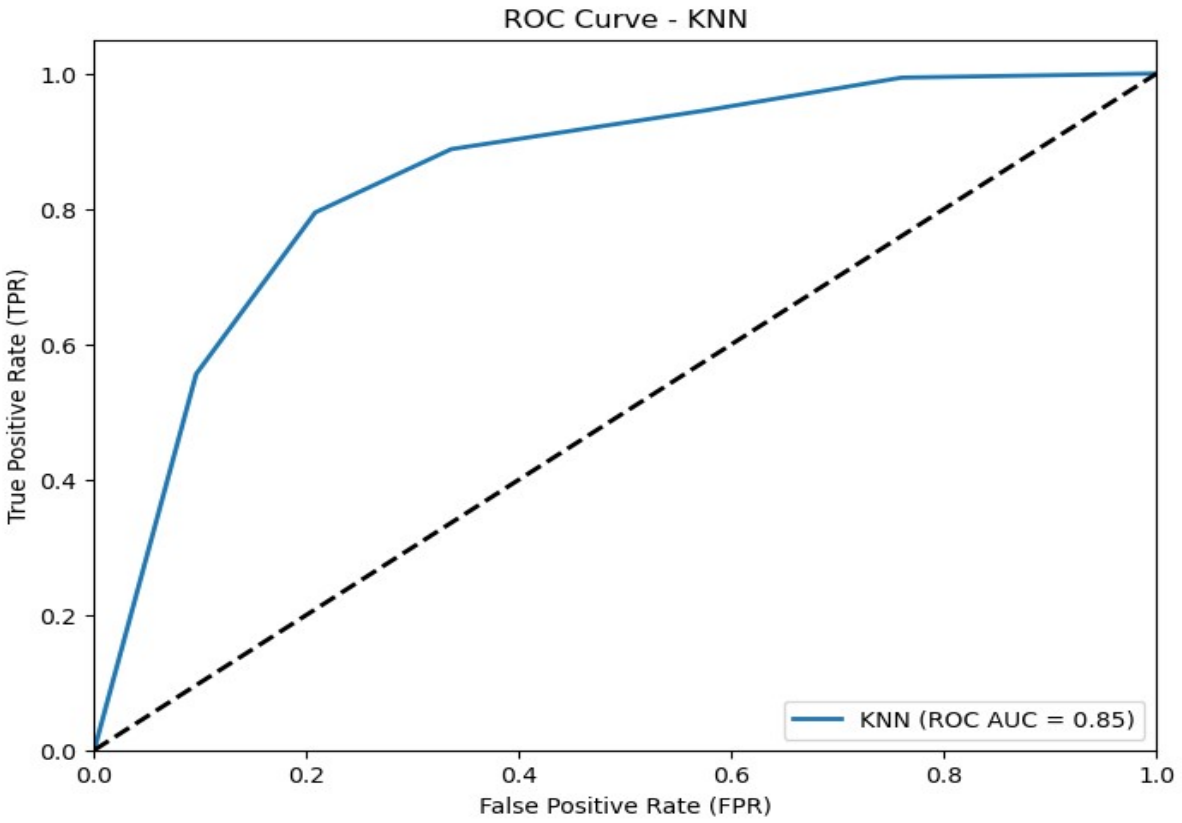


Figure 11: ROC Curve- KNN

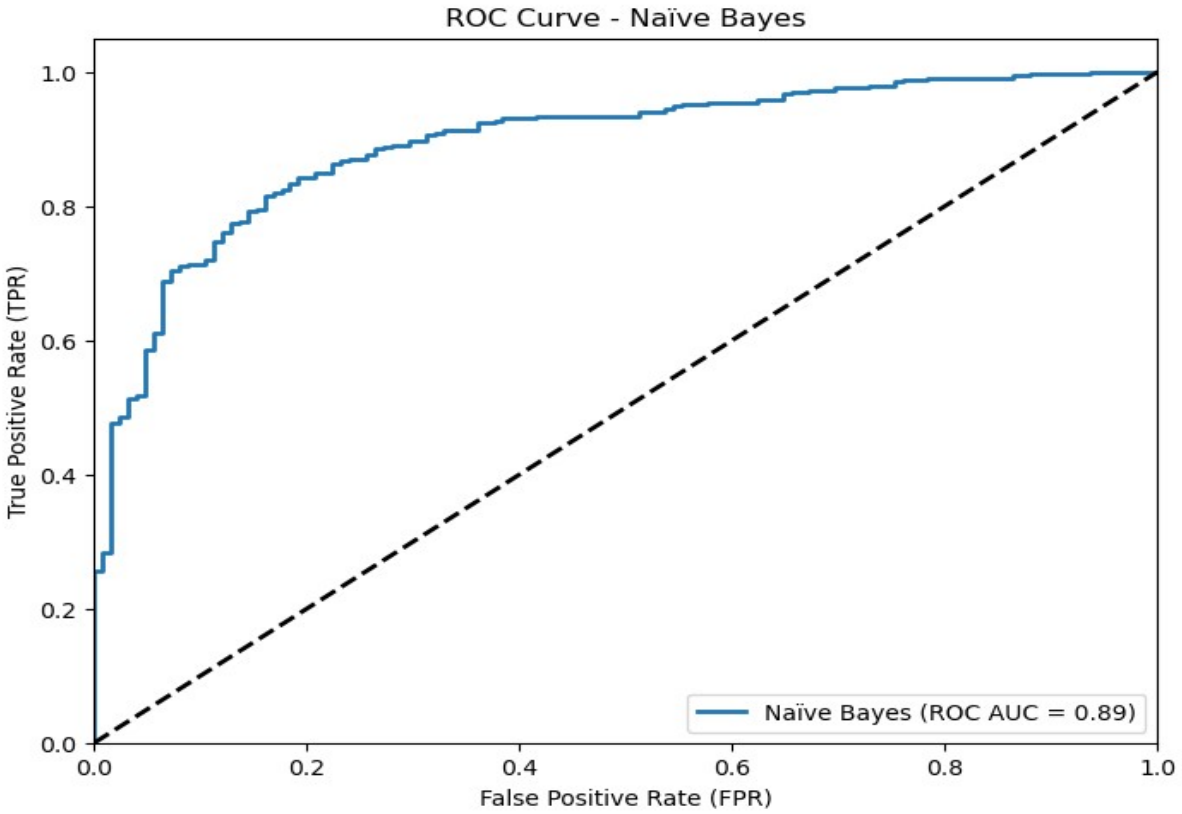


Figure 12: ROC Curve- Naïve Bayes

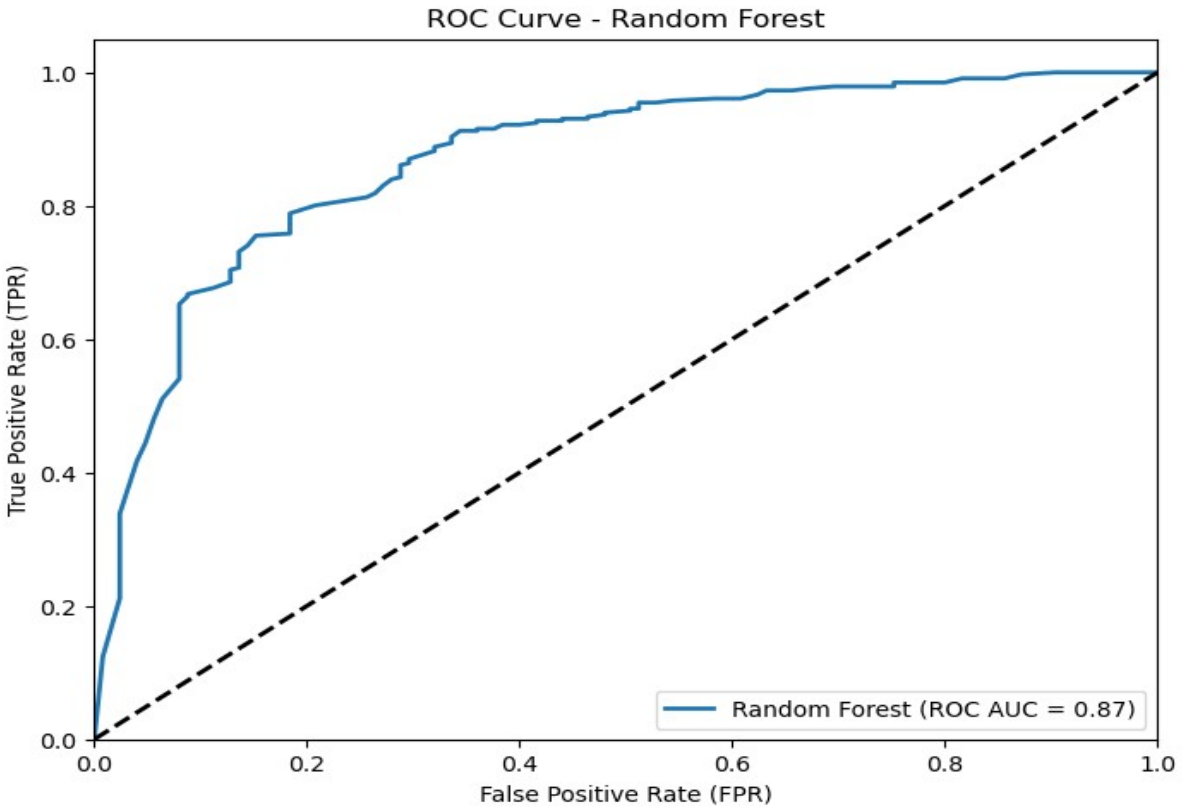


Figure 13: ROC Curve- Random Forest

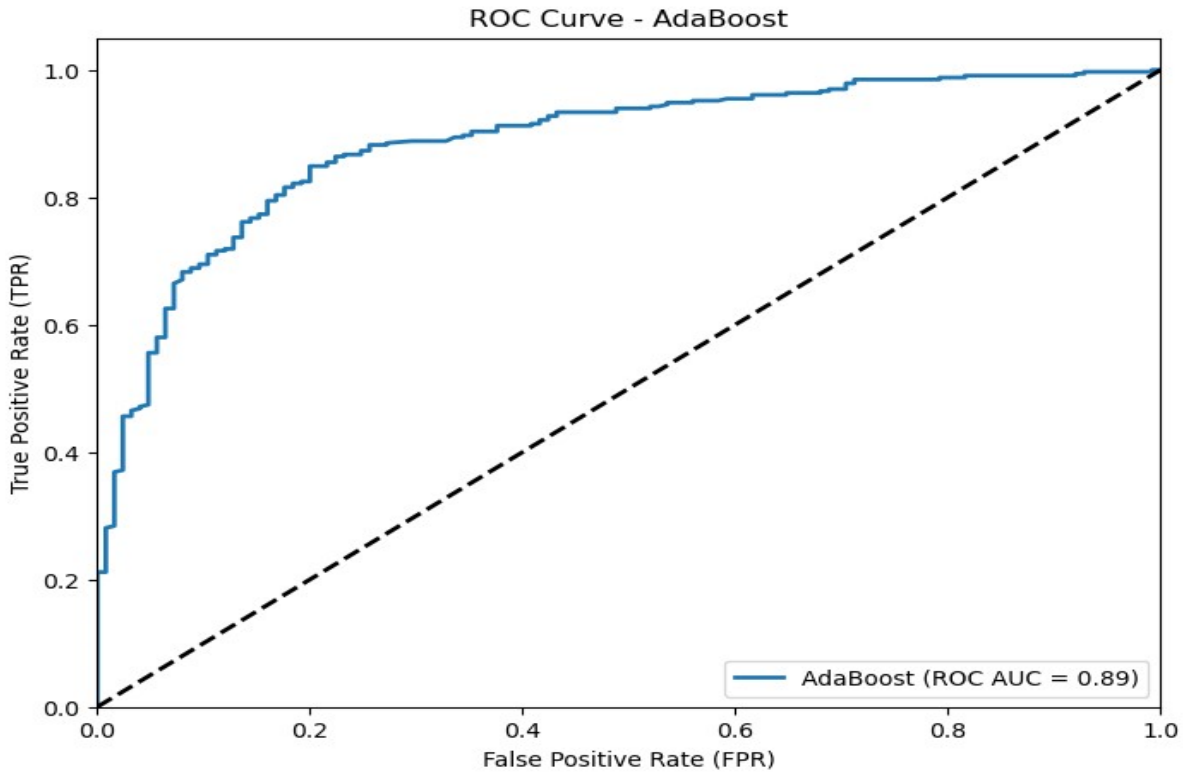


Figure 14: ROC Curve- AdaBoost

Logistic Regression ROC AUC Score: 0.89
LDA ROC AUC Score: 0.89
KNN ROC AUC Score: 0.85
Naïve Bayes ROC AUC Score: 0.89
Random Forest ROC AUC Score: 0.87
AdaBoost ROC AUC Score: 0.89

Table 30: Classification report of Logistic Regression

Classification Report for Logistic Regression:

	precision	recall	f1-score	support
Conservative	0.74	0.58	0.65	125
Labour	0.85	0.92	0.89	331
accuracy			0.83	456
macro avg	0.80	0.75	0.77	456
weighted avg	0.82	0.83	0.82	456

Table 31: Classification report of LDA

Classification Report for LDA:

	precision	recall	f1-score	support
Conservative	0.72	0.62	0.67	125
Labour	0.86	0.91	0.89	331
accuracy			0.83	456
macro avg	0.79	0.77	0.78	456
weighted avg	0.83	0.83	0.83	456

Table 32: Classification report of KNN

Classification Report for KNN:

	precision	recall	f1-score	support
Conservative	0.69	0.66	0.68	125
Labour	0.88	0.89	0.88	331
accuracy			0.83	456
macro avg	0.78	0.78	0.78	456
weighted avg	0.82	0.83	0.83	456

Table 33: Classification report of Naïve Bayes

Classification Report for Naïve Bayes:

	precision	recall	f1-score	support
Conservative	0.73	0.69	0.71	125
Labour	0.88	0.90	0.89	331
accuracy			0.84	456
macro avg	0.81	0.80	0.80	456
weighted avg	0.84	0.84	0.84	456

Table 34: Classification report of Random Forest

Classification Report for Random Forest:

	precision	recall	f1-score	support
Conservative	0.74	0.62	0.68	125
Labour	0.87	0.92	0.89	331
accuracy			0.84	456
macro avg	0.80	0.77	0.78	456
weighted avg	0.83	0.84	0.83	456

Table 35: Classification report of AdaBoost

Classification Report for AdaBoost:

	precision	recall	f1-score	support
Conservative	0.70	0.65	0.68	125
Labour	0.87	0.90	0.88	331
accuracy			0.83	456
macro avg	0.79	0.77	0.78	456
weighted avg	0.83	0.83	0.83	456

Table 36: Performance matrix table of each model

	Model	Train Accuracy	Test Accuracy	ROC AUC Score
0	Logistic Regression	0.83	0.83	0.89
1	LDA	0.83	0.83	0.89
3	Naïve Bayes	0.83	0.84	0.89
5	AdaBoost	0.86	0.85	0.89
4	Random Forest	0.89	0.83	0.87
2	KNN	0.87	0.83	0.85

Based on the comparison of performance metrics, AdaBoost appears to be the best-optimized model for the problem at hand. It achieves a high-test accuracy, a strong ROC AUC score, and demonstrates good generalization. It strikes a balance between accuracy and model complexity, making it a suitable choice as the final model for this classification problem.

1.8. Based on these predictions, what are the insights?

Insights and Recommendations:

Model Selection:
After comprehensive model evaluation, it is recommended to choose the AdaBoost classifier as the final model for predicting political party affiliation. AdaBoost offers a strong balance between accuracy and generalization performance, making it suitable for this classification problem.

Targeted Campaigns:
The model can be leveraged to create targeted political campaigns. Identify potential voters from the dataset and tailor campaign messages based on their predicted political affiliation. This approach can lead to more efficient resource allocation and a higher chance of winning over undecided voters.

Data Collection and Quality:
Continuously collect and update data on voter demographics, issues of concern, and political preferences to improve model accuracy. Ensure data quality and consistency to maintain model reliability and effectiveness.

Monitoring and Feedback:
Implement a system for monitoring the effectiveness of campaigns and gathering feedback from voters. Analyze the impact of campaign messages and adjust strategies accordingly for better engagement and results.

Competitor Analysis:
Conduct competitor analysis to understand the strategies of rival political parties. Use insights from the analysis to refine campaign messages and stay ahead in the political landscape.

Outreach Expansion:
Consider expanding outreach efforts to engage with a wider audience. Use the predictive model to identify areas or demographics where the party's message may resonate and invest resources accordingly.

Collaboration and Alliances:
Explore collaboration opportunities with like-minded political groups or parties to strengthen support. Data-driven insights can help identify potential allies and areas of common interest.

Voter Education:
Invest in voter education programs to inform voters about the party's stance on important issues. An informed electorate is more likely to make decisions aligned with the party's values.

Crisis Management:
Develop a crisis management strategy based on potential scenarios predicted by the model. Be prepared to respond effectively to changing political dynamics.

Feedback Loop:
Establish a feedback loop with voters to actively address their concerns and demonstrate responsiveness.
This can foster trust and loyalty among the party's supporters.

By implementing these recommendations and leveraging the predictive power of the AdaBoost model, the political party can enhance its campaign strategies, engage with voters more effectively, and increase its chances of success in upcoming elections.

Problem 2:

In this particular project, we are going to work on the inaugural corpora from the nltk in Python. We will be looking at the following speeches of the Presidents of the United States of America:

President Franklin D. Roosevelt in 1941
President John F. Kennedy in 1961
President Richard Nixon in 1973

(Hint: use .words(), .raw(), .sent() for extracting counts)

2.1 Find the number of characters, words and sentences for the mentioned documents. (Hint: use .words(), .raw(), .sent() for extracting counts)

Table 37: Number of characters, words and sentence count

S.No	Name	Character_Count	Word_Count	Sentence_Count
1	Roosevelt	7651	1453	32
2	Kennedy	7673	1494	27
3	Nixon	10106	1913	20

2.2 Remove all the stopwords from the three speeches. Show the word count before and after the removal of stopwords. Show a sample sentence after the removal of stopwords.

Before removing stopwords:

1. Speech column is converted to lower case.
2. Removal of special character.
3. Stemming is performed.
4. Stopwords are removed.

Table 38: word count of raw speech and after removal of stopwords

S.No	Name	Word_Count	Cleaned_Word_Count
1	Roosevelt	1453	663
2	Kennedy	1494	722
3	Nixon	1913	866

2.3 Which word occurs the most number of times in his inaugural address for each president? Mention the top three words. (after removing the stopwords)

Table 39: Most Common word in each speech

S.No	Name	Most_Common_Word
1	Roosevelt	(nation, 16)
2	Kennedy	(let, 11)
3	Nixon	(us, 26)

Table 40: Top Three word in each speech

S.No	Name	Top_Three_Words
1	Roosevelt	[(nation, 16), (thi, 12), (ha, 10)]
2	Kennedy	[(let, 11), (thi, 11), (us, 11)]
3	Nixon	[(us, 26), (america, 19), (respons, 16)]

2.4 Plot the word cloud of each of the three speeches. (after removing the stopwords)

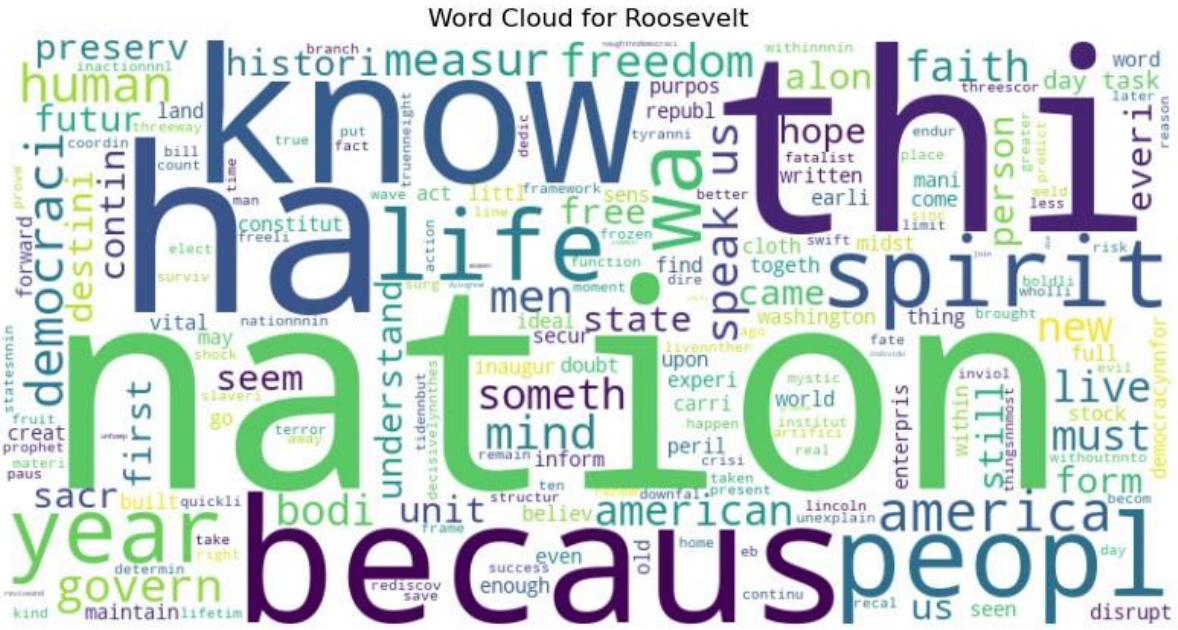


Figure 12: Word cloud for each of the three speech