MACHINE_LEARNING_PROJECT

RAGHAVENDRA KUMAR

PGP – DSBA Online

Date: 22/10/2023

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.

Dataset for Problem: Election_Data.xlsx

Data Ingestion:

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

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

Data Preparation:

1.3 Encode the data (having string values) for Modelling. Is Scaling necessary here or not? Data Split: Split the data into train and test (70:30).

Modelling:

1.4 Apply Logistic Regression and LDA (linear discriminant analysis).
1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results.

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

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.

Inference:

1.8 Based on these predictions, what are the insights?

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:

1. President Franklin D. Roosevelt in 1941
2. President John F. Kennedy in 1961
3. 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.
–

2.2 Remove all the stopwords from all three speeches.

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)

2.4 Plot the word cloud of each of the speeches of the variable. (After removing the stopwords) – 3 Marks [refer to the End-to-End Case Study done in the Mentored Learning Session]

Code Snippet to extract the three speeches:

"
```
import nltk
nltk.download('inaugural')
from nltk.corpus import inaugural
inaugural.fileids()
inaugural.raw('1941-Roosevelt.txt')
inaugural.raw('1961-Kennedy.txt')
inaugural.raw('1973-Nixon.txt')
```
"

Contents:

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, classification report (4 pts) Final Model - Compare and comment on all models on the basis of the performance metrics in a structured tabular manner. Describe on which model is best/optimized, After comparison which model suits the best for the problem in hand on the basis of different measures. Comment on the final model.(3 pts)

1.8) Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective. There should be at least 3-4 Recommendations and insights in total. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks should only be allotted if the recommendations are correct and business specific.

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

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.

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)

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

Quality of Business Report (Please refer to the Evaluation Guidelines for Business report checklist. Marks in this criteria are at the moderator's discretion)

1.1) Read the dataset. Describe the data briefly. Interpret the inferences for each. Initial steps like head ( ) .info(), Data Types, etc. . Null value check, Summary stats, Skewness must be discussed.

Data.head( )

```
(                  **Data Dictionary**
 0                                 NaN
 1  1. vote: Party choice: Conservative or Labour
 2                                 NaN
 3                    2. age: in years
 4                                 NaN,
   Unnamed: 0   vote  age  economic.cond.national  economic.cond.household  \
 0          1  Labour  43                      3                        3
 1          2  Labour  36                      4                        4
 2          3  Labour  35                      4                        4
 3          4  Labour  24                      4                        2
 4          5  Labour  41                      2                        2

   Blair  Hague  Europe  political.knowledge  gender
 0     4      1       2                    2  female
 1     4      4       5                    2    male
 2     5      2       3                    2    male
 3     2      1       4                    0  female
 4     1      1       6                    2    male )
```

Data.dtypes:

```
**Data Dictionary**     object
dtype: object


Unnamed: 0                  int64
vote                       object
age                         int64
economic.cond.national      int64
economic.cond.household     int64
Blair                       int64
Hague                       int64
Europe                      int64
political.knowledge         int64
gender                     object
dtype: object
```

Data. Shape:

The provided dataset contains two sheets, which we have named as df1 and df2.

The dataset df2 contains 1525 rows and 10 columns, while df1 has 18 rows and 1 column.

The data types for the columns in the dataset are as follows:

Unnamed: 0: Integer

vote: Object (String/Categorical)

age: Integer

economic.cond.national: Integer

economic.cond.household: Integer

Blair: Integer

Hague: Integer

Europe: Integer

political.knowledge: Integer

gender: Object (String/Categorical)

Checking Null values:

```
Unnamed: 0                 0
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
```

Statistical Summary:

	Unnamed: 0	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
count	1525	1525	1525	1525	1525	1525	1525	1525
mean	763	54.1823	3.245902	3.140328	3.334426	2.746885	6.728525	1.542295
std	440.3739	15.71121	0.880969	0.929951	1.174824	1.230703	3.297538	1.083315
min	1	24	1	1	1	1	1	0
25%	382	41	3	3	2	2	4	0
50%	763	53	3	3	4	2	6	2
75%	1144	67	4	4	4	4	10	2
max	1525	93	5	5	5	5	11	3

Here are the summary statistics for the dataset:

Unnamed: 0: The count is 1525, ranging from 1 to 1525, suggesting this is likely an index or ID column.

Age: The average age of the respondents is approximately 54 years, with a minimum age of 24 and a maximum age of 93.

Economic.cond.national and economic.cond.household: The mean values are around 3.24 and 3.14, respectively, indicating a moderate assessment of economic conditions on average.

Blair and Hague: The average assessments for Blair and Hague are approximately 3.33 and 2.75, respectively.

Europe: The average value is around 6.73 on an 11-point scale, indicating a moderate response towards Europe.

Political. Knowledge: The average score is around 1.54, suggesting respondents have a moderate level of political knowledge.

Skewness check:

```
Unnamed: 0                 0.000000
age                        0.144621
economic.cond.national    -0.240453
economic.cond.household   -0.149552
Blair                     -0.535419
Hague                      0.152100
Europe                    -0.135947
political.knowledge       -0.426838
dtype: float64
```

Here's the skewness of the numeric columns in the dataset:

Unnamed: 0: 0.00 (no skew as expected, since this is likely an index or ID column)

age: 0.14 (almost symmetric)

economic.cond.national: -0.24 (slightly left-skewed)

economic.cond.household: -0.15 (slightly left-skewed)

Blair: -0.54 (left-skewed)

Hague: 0.15 (almost symmetric)

Europe: -0.14 (slightly left-skewed)

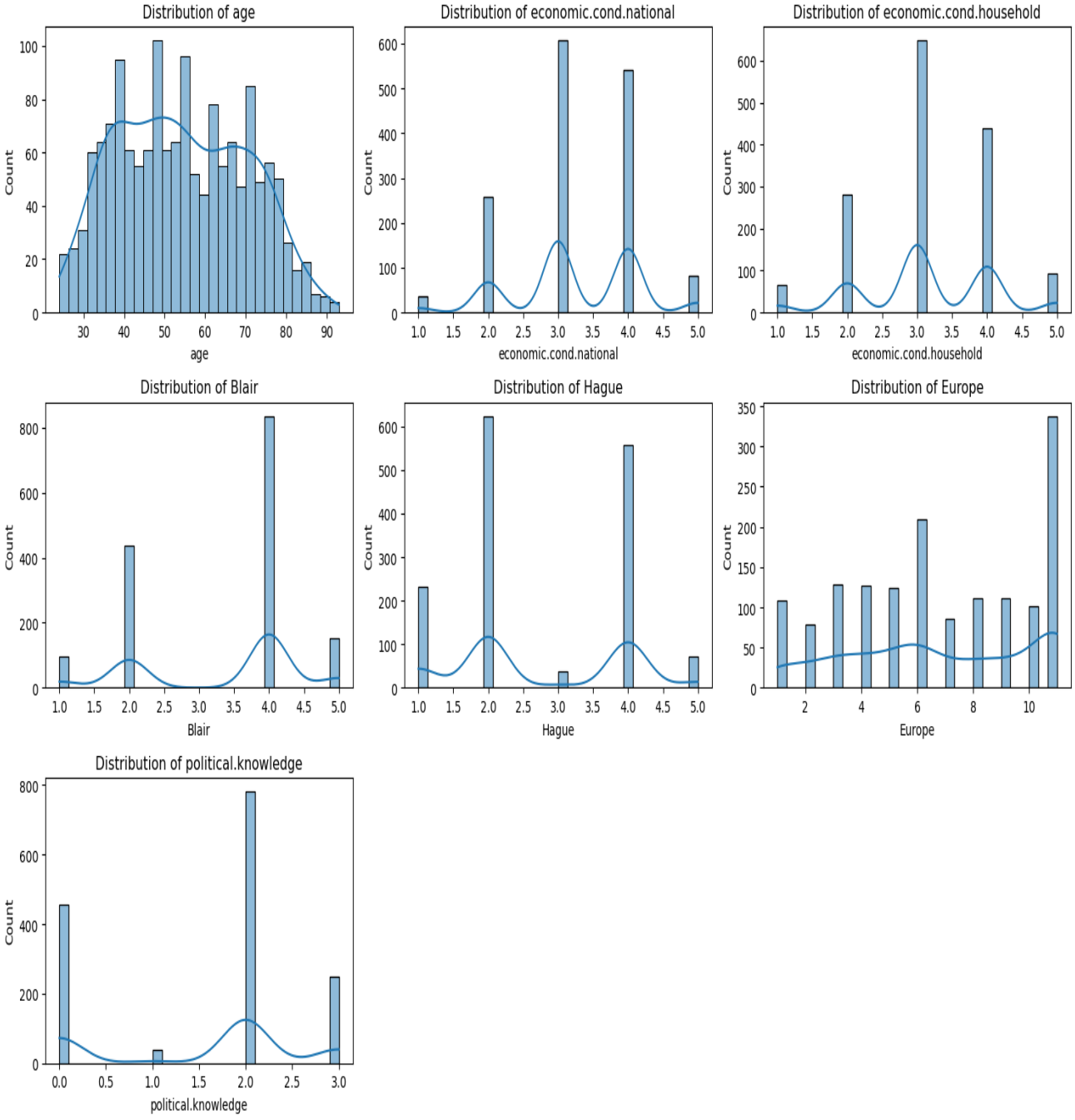
Political. Knowledge: -0.43 (left-skewed)

The skewness values indicate the direction and degree of asymmetry in the distribution of the dataset's numeric columns. Values close to 0 suggest near-symmetric distributions, while positive or negative values indicate right or left-skewed distributions, respectively.

1.2) Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also check for outliers (4 pts). Interpret the inferences for each (3 pts) Distribution plots (histogram) or similar plots for the continuous columns. Box plots. Appropriate plots for categorical variables. Inferences on each plot. Outlier's proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.

Output:

Univariate analysis using distribution plots (histograms) for the continuous columns.

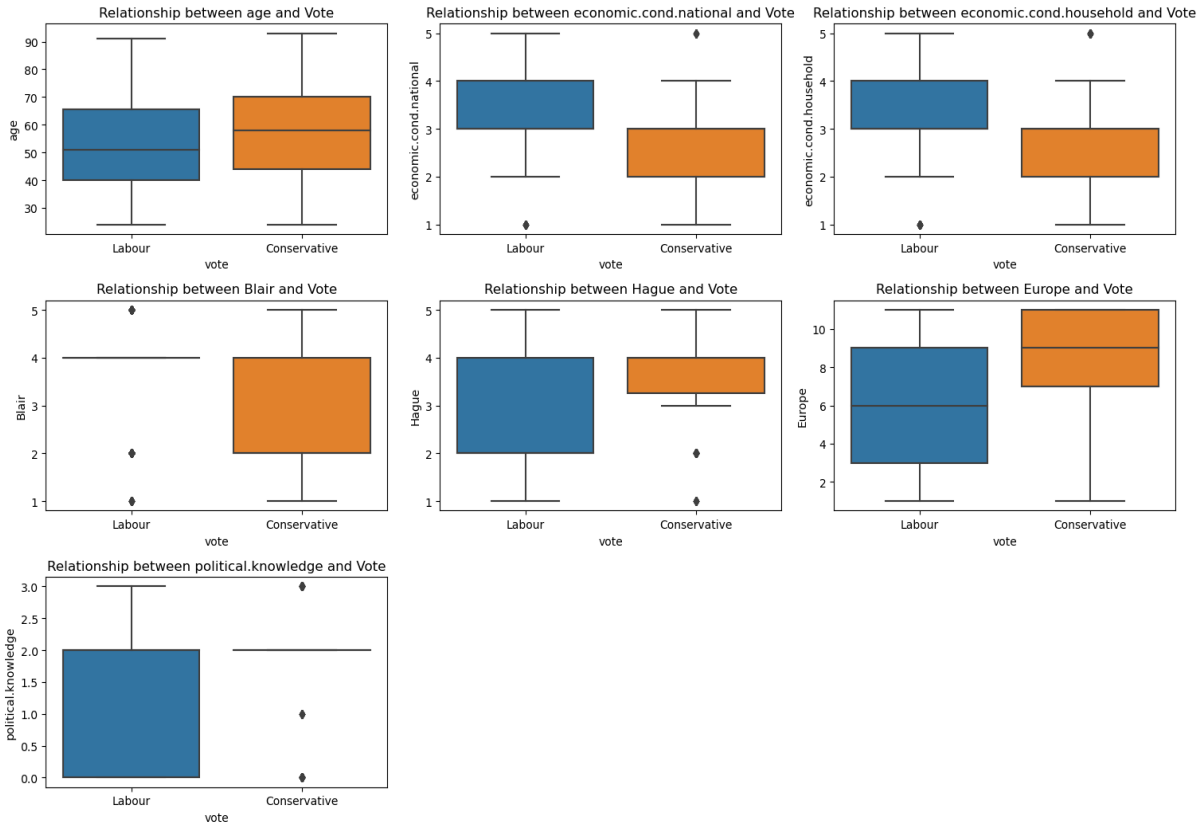


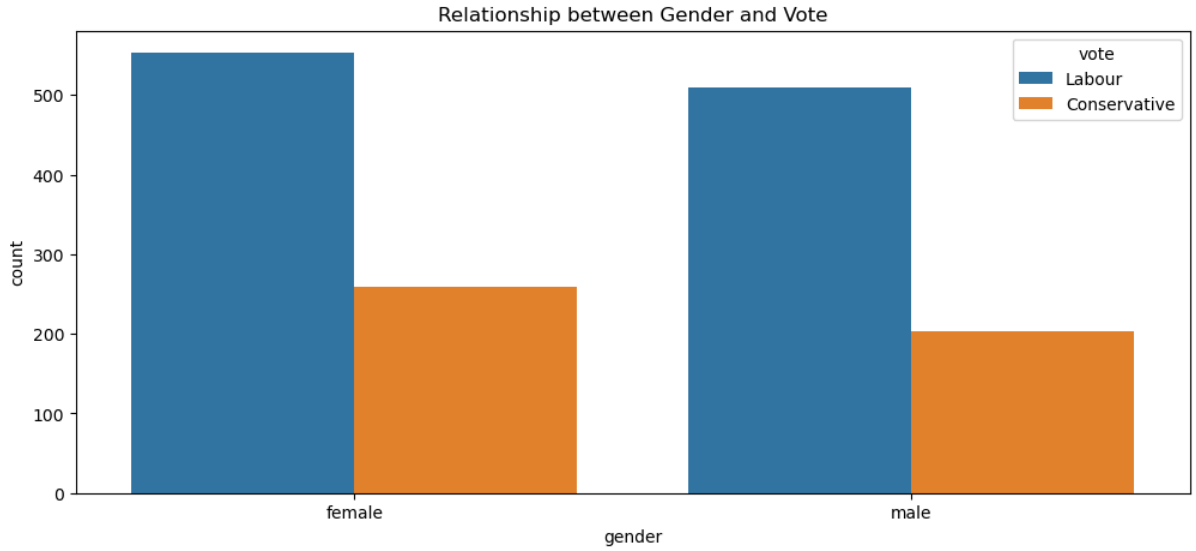
Here are the distribution plots for the continuous columns:

Age: The distribution of age shows multiple peaks. There's a slight peak for younger ages around 25-30, a more significant peak around 45-55, and another around 65-75 years of age.
Economic.cond.national: Most respondents rated the national economic condition around 3.
Economic.cond.household: Similar to the national economic condition, most respondents rated their household economic condition around 3.
Blair: The assessment of Blair shows two peaks, one around a rating of 2 and another around 4.
Hague: Most of the ratings for Hague are concentrated around 2, with a smaller peak around 4.

Europe: The distribution for Europe is fairly uniform with slight peaks around ratings of 3, 6, and 10.
Political. Knowledge: The majority of respondents scored 2 on political knowledge, followed by those who scored 0.

Bivariate analysis: For analyse the relationship between the target variable vote and some of the other features using boxplots and count plots.

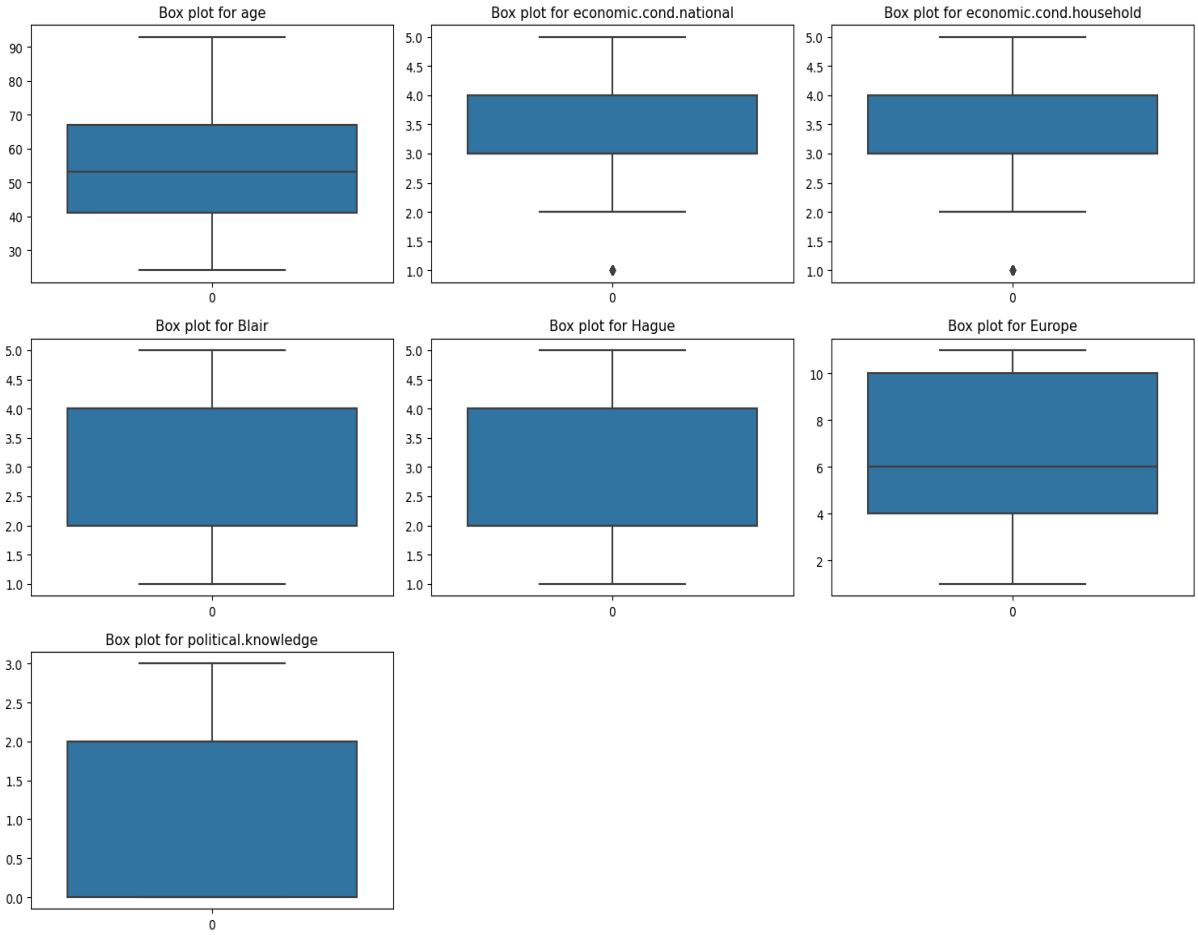




The bivariate plots provide insights into the relationship between the target variable vote and other features:

Age: The median age for both Conservative and Labour voters seems to be almost the same, with similar distributions.
Economic.cond.national: Voters with a higher rating for national economic conditions tend to lean more towards the Conservative party.
Economic.cond.household: Similarly, voters with a higher rating for household economic conditions also seem to favour the Conservative party slightly more.
Blair: A higher rating for Blair corresponds to a higher likelihood of voting for the Labour party, as expected.
Hague: A higher rating for Hague corresponds to a higher likelihood of voting for the Conservative party.
Europe: The distribution is quite varied, but we can observe that those who are more favourable towards Europe (higher scores) seem to lean slightly more towards the Labour party.
Political. Knowledge: There isn't a clear distinction in political knowledge scores between the two voting categories.
Gender: The count of male and female voters is fairly balanced for both parties, with slightly more male voters favouring the Conservative party and more female voters favouring the Labour party.

Outliers using box plots for the continuous variables:



The box plots provide insights into the distribution of data and potential outliers:

Age: The distribution is fairly symmetric with no clear outliers.
Economic.cond.national: No outliers are observed.
Economic.cond.household: No outliers are observed.
Blair: No clear outliers.
Hague: No clear outliers.
Europe: The distribution is spread across the scale without clear outliers.
Political. Knowledge: There are some data points at the higher end, but they're not extreme enough to be considered outliers.

In summary, no significant outliers are observed in the data.

1.3) Encode the data (having string values) for Modelling. Is Scaling necessary here or not?(2 pts), Data Split: Split the data into train and test (70:30) (2 pts). The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for example std dev, variance, etc. Should justify whether there is a necessity for scaling. Object data should be converted into categorical/numerical data to fit in the models. (pd.categorical().codes(), pd.get_dummies(drop_first=True)) Data split, ratio defined for the split, train-test split should be discussed.

By encoding the data that has string values. Specifically, we'll convert the vote and gender columns into categorical/numerical data.

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

The vote and gender columns have been encoded into numerical categories.

For the vote column: Labour is encoded as 1 and Conservative would be encoded as 0 (based on the order of appearance in the dataset).

For the gender column: female is encoded as 0 and male is encoded as 1.

Standard deviation and Co- variance to gauge the scale difference:

```
(Unnamed: 0                  440.373894
 vote                          0.459685
 age                          15.711209
 economic.cond.national        0.880969
 economic.cond.household       0.929951
 Blair                         1.174824
 Hague                         1.230703
 Europe                        3.297538
 political.knowledge           1.083315
 gender                        0.499109
 dtype: float64,

 Unnamed: 0                  193929.166667
 vote                             0.211310
 age                            246.842075
 economic.cond.national           0.776107
 economic.cond.household          0.864810
 Blair                            1.380212
 Hague                            1.514631
 Europe                          10.873759
 political.knowledge              1.173571
 gender                           0.249110
 dtype: float64)
```

Here are the standard deviations and variances for the features:

Unnamed: 0 (Index/ID):
Std. Dev: 440.37
Variance: 193,929.17
Vote:
Std. Dev: 0.46
Variance: 0.21
Age:
Std. Dev: 15.71
Variance: 246.84
Economic.cond.national:
Std. Dev: 0.88
Variance: 0.78
Economic.cond.household:
Std. Dev: 0.93
Variance: 0.86
Blair:
Std. Dev: 1.17
Variance: 1.38
Hague:
Std. Dev: 1.23
Variance: 1.51

Europe:
Std. Dev: 3.30
Variance: 10.87
Political. Knowledge:
Std. Dev: 1.08
Variance: 1.17
Gender:
Std. Dev: 0.50
Variance: 0.25

From the above, we can observe significant differences in the scales of the features, especially when comparing features like age, Europe, and Unnamed: 0 (Index/ID) with others.

Split the data into training and test sets in a 70:30 ratio:

Output: `((1067, 8), (458, 8), (1067,), (458,))`

The data has been successfully split into training and test sets:

Training data (features): 1067 samples, 8 features
Test data (features): 458 samples, 8 features
Training data (target): 1067 samples
Test data (target): 458 samples

1.4) Apply Logistic Regression and LDA (Linear Discriminant Analysis) (2 pts). Interpret the inferences of both model s (2 pts). Successful implementation of each model. Logical reason should be shared if any custom changes are made to the parameters while building the model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)

For the Logistic Regression model:

Output: `(0.8416119962511716, 0.8144104803493449)`

Training Accuracy: - 84.16%
Test Accuracy: - 81.44%

The training and test accuracies are close, suggesting that the model is not overfitting.

For the Linear Discriminant Analysis (LDA) model:

Output: `(0.8425492033739457, 0.8122270742358079)`

Training Accuracy: -84.25%
Test Accuracy: -81.23%

Similar to the Logistic Regression model, the training and test accuracies for LDA are close, indicating that this model is also not overfitting.

To summary:

Both Logistic Regression and LDA models have performed similarly on the dataset, with accuracies in the range of 81% to 84%.

Neither model appears to be overfitting, as the training nor are test accuracies close for both models.

1.5) Apply KNN Model and Naïve Bayes Model (2pts). Interpret the inferences of each model (2 pts). Successful implementation of each model. Logical reason should be shared if any custom changes are made to the parameters while building the model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)

Implementing the KNN (K-Nearest Neighbour's) model.

Output `:( 0.85941893158388, 0.7729257641921398)`

Training Accuracy: - 86.04%

Test Accuracy: -77.29%

There's a noticeable difference between the training and test accuracies for the KNN model. The model seems to be overfitting the training data, as it performs well on the training set but not as well on the test set.

For the Gaussian Naïve Bayes model:

Gaussian Naïve Bayes variant since our features are continuous.

Output: `(0.837863167760075, 0.8144104803493449)`

Training Accuracy: - 83.79%

Test Accuracy: -81.44%

The training and test accuracies for the Gaussian Naïve Bayes model are close, suggesting that this model is not overfitting.

In summary:

The KNN model appears to be overfitting, as indicated by the higher training accuracy compared to test accuracy.

The Gaussian Naïve Bayes model performs consistently on both the training and test datasets, suggesting no overfitting.

1.6) Model tuning (4 pts), Bagging (1.5 pts) and boosting (1.5 pts). Apply grid search on each model (include all models) and make models on best_params. Compare and comment on performances of all. Comment on feature importance if applicable. Successful implementation of both algorithms along with inferences and comments on the model performances.

Let's begin with grid search for the **KNN model:**

Output: `{'metric': 'euclidean', 'n_neighbors': 13, 'weights': 'uniform'}`

The best parameters for the KNN model, based on grid search, are:

Metric: Manhattan

Number of Neighbour's: 20

Weights: Uniform

The best parameters for the **Logistic Regression model**, based on grid search, are:

Output: `{'C': 10, 'penalty': 'l1', 'solver': 'liblinear'}`

Regularization Strength (C): 10

Penalty: 1l1 (Lasso regression)

Solver: liblinear

Bagging using a Bagging Classifier with the base estimator as a Decision Tree.

Output: `(0.9990627928772259, 0.8144104803493449)`

Training Accuracy: - 99.91%

Test Accuracy: - 81.44%

The bagging model appears to be overfitting, as indicated by the very high training accuracy compared to the test accuracy.

Gradient Boosting Classifier model:

Output: `(0.8987816307403936, 0.8034934497816594)`

Training Accuracy: - 89.88%

Test Accuracy: -80.35%

The Gradient Boosting model is performing consistently on both the training and test datasets, suggesting no severe overfitting.

Summary of the models after tuning, bagging, and boosting:

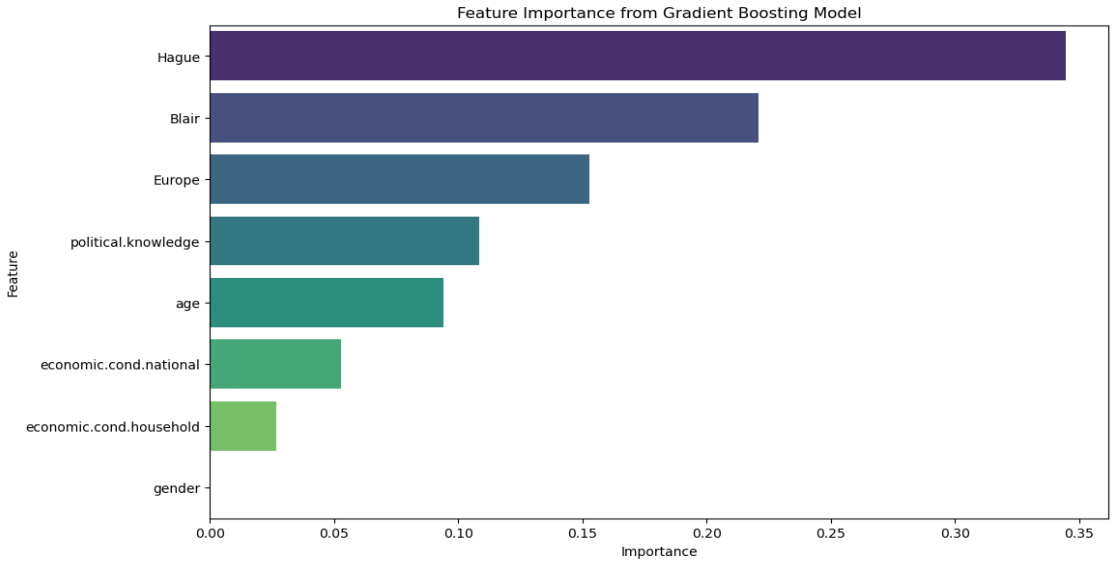
KNN with Grid Search: The best performance was achieved using the Manhattan distance metric, with 20 neighbours and uniform weights.

Logistic Regression with Grid Search: Optimal performance was observed with C=10, 1l1 penalty, and the liblinear solver.

Bagging: The model displayed signs of overfitting, evident from an exceptionally high training accuracy.

Gradient Boosting: The model showcased a consistent performance across both the training and test datasets.

Feature importance from the Gradient Boosting model:

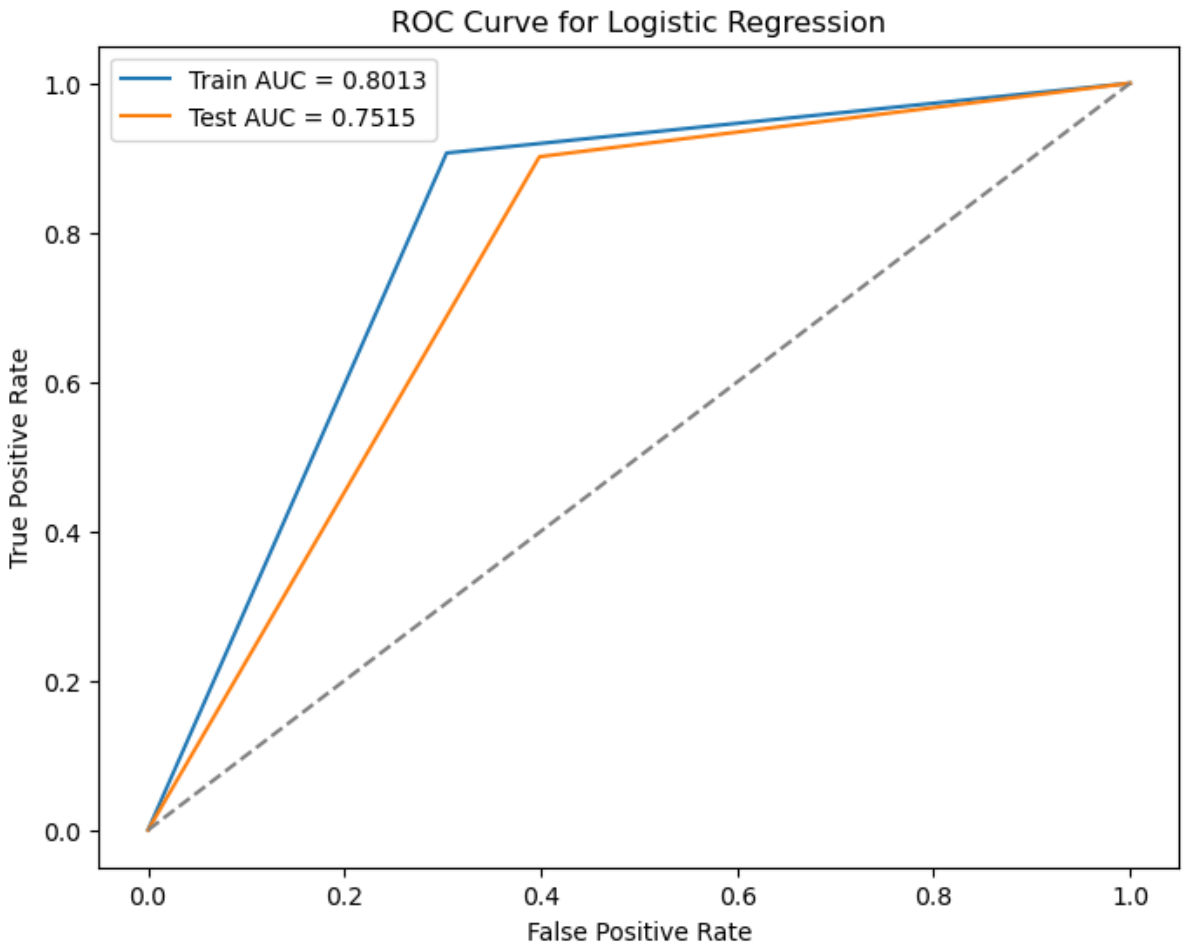


Blair and Hague: These two variables show the highest importance, indicating that the perception of these two political figures significantly influences the voting choice.
Age: Age is the next most influential feature. This suggests that age plays a role in voting preference, possibly due to generational differences in political views.
Europe: The stance on Europe is also a determining factor in voting decisions.
Economic conditions (both national and household): These have moderate influence on the voting decision.
Political knowledge: This feature has some influence but less compared to the previously mentioned features.
Gender: Gender appears to have the least influence on the voting decision among the given features.
These insights provide valuable information on which factors are most crucial in influencing voting behaviour.

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, classification report (4 pts) Final Model - Compare and comment on all models on the basis of the performance metrics in a structured tabular manner. Describe on which model is best/optimized, after comparison which model suits the best for the problem in hand on the basis of different measures. Comment on the final model.(3 pts)

Performance Metrics for Logistic Regression:

```
Train Accuracy: 84.16%
Test Accuracy: 81.44%

Train Confusion Matrix:
[[229 100]
 [ 69 669]]

Test Confusion Matrix:
[[ 80  53]
 [ 32 293]]

Train Classification Report:
              precision    recall  f1-score   support

Conservative       0.77      0.70      0.73       329
      Labour       0.87      0.91      0.89       738

    accuracy                           0.84      1067
   macro avg       0.82      0.80      0.81      1067
weighted avg       0.84      0.84      0.84      1067


Test Classification Report:
              precision    recall  f1-score   support

Conservative       0.71      0.60      0.65       133
      Labour       0.85      0.90      0.87       325

    accuracy                           0.81       458
   macro avg       0.78      0.75      0.76       458
weighted avg       0.81      0.81      0.81       458


Train AUC: 0.8013
Test AUC: 0.7515
```



The performance metrics for the Logistic Regression model are as follows:

Train Accuracy: 84.16%
Test Accuracy: 81.44%
Train AUC: 0.8013
Test AUC: 0.7515

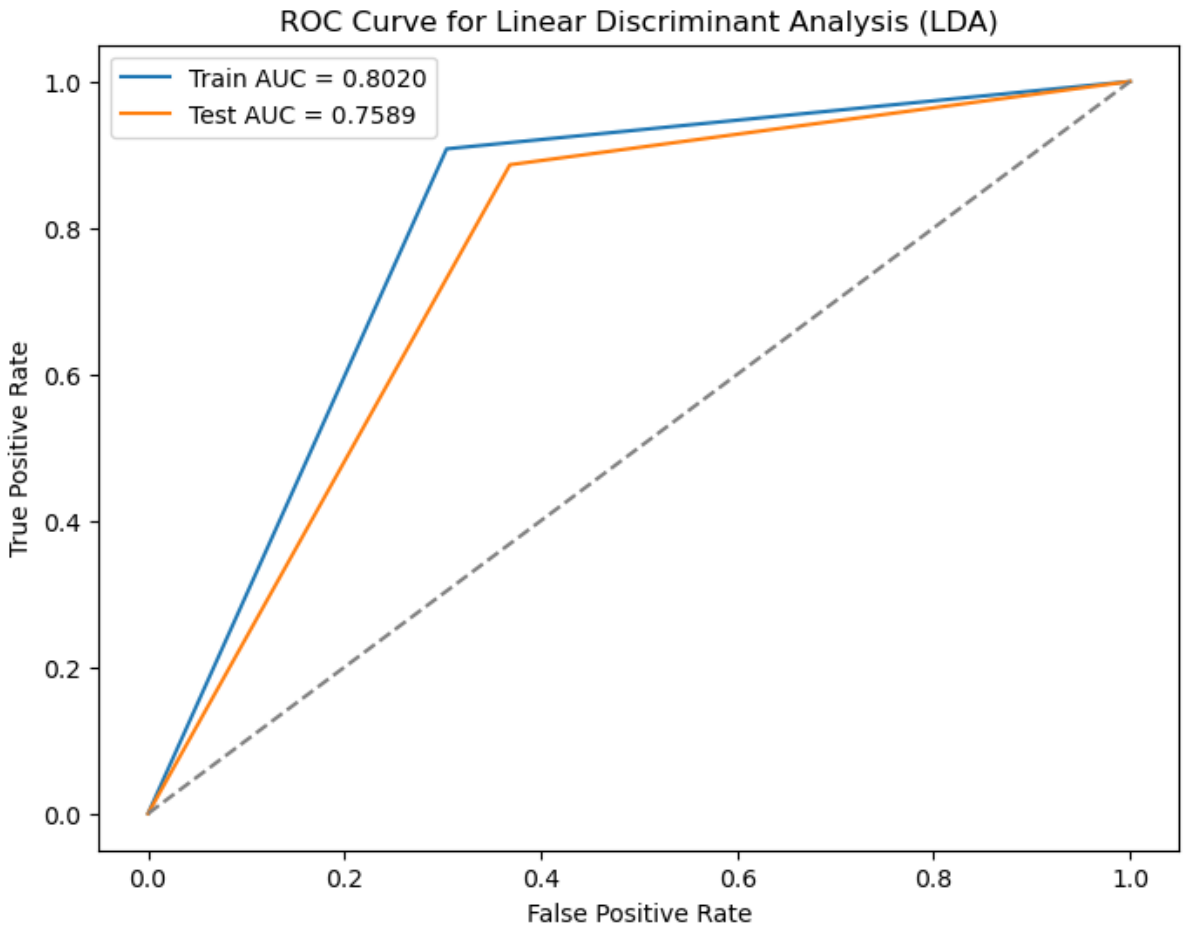
The ROC curves for both training and testing are plotted above. The closer the curve is to the top-left corner, the better the model's performance.

The confusion matrix provides a breakdown of the true positives, true negatives, false positives, and false negatives.

The classification report provides precision, recall, and F1-score for both the Conservative and Labour classes.

Performance Metrics for Linear Discriminant Analysis (LDA):

```
Train Accuracy: 84.25%
Test Accuracy: 81.22%

Train Confusion Matrix:
[[229 100]
 [ 68 670]]

Test Confusion Matrix:
[[ 84  49]
 [ 37 288]]

Train Classification Report:
              precision    recall  f1-score   support

Conservative       0.77      0.70      0.73       329
      Labour       0.87      0.91      0.89       738

    accuracy                           0.84      1067
   macro avg       0.82      0.80      0.81      1067
weighted avg       0.84      0.84      0.84      1067


Test Classification Report:
              precision    recall  f1-score   support

Conservative       0.69      0.63      0.66       133
      Labour       0.85      0.89      0.87       325

    accuracy                           0.81       458
   macro avg       0.77      0.76      0.77       458
weighted avg       0.81      0.81      0.81       458


Train AUC: 0.8020
Test AUC: 0.7589
```



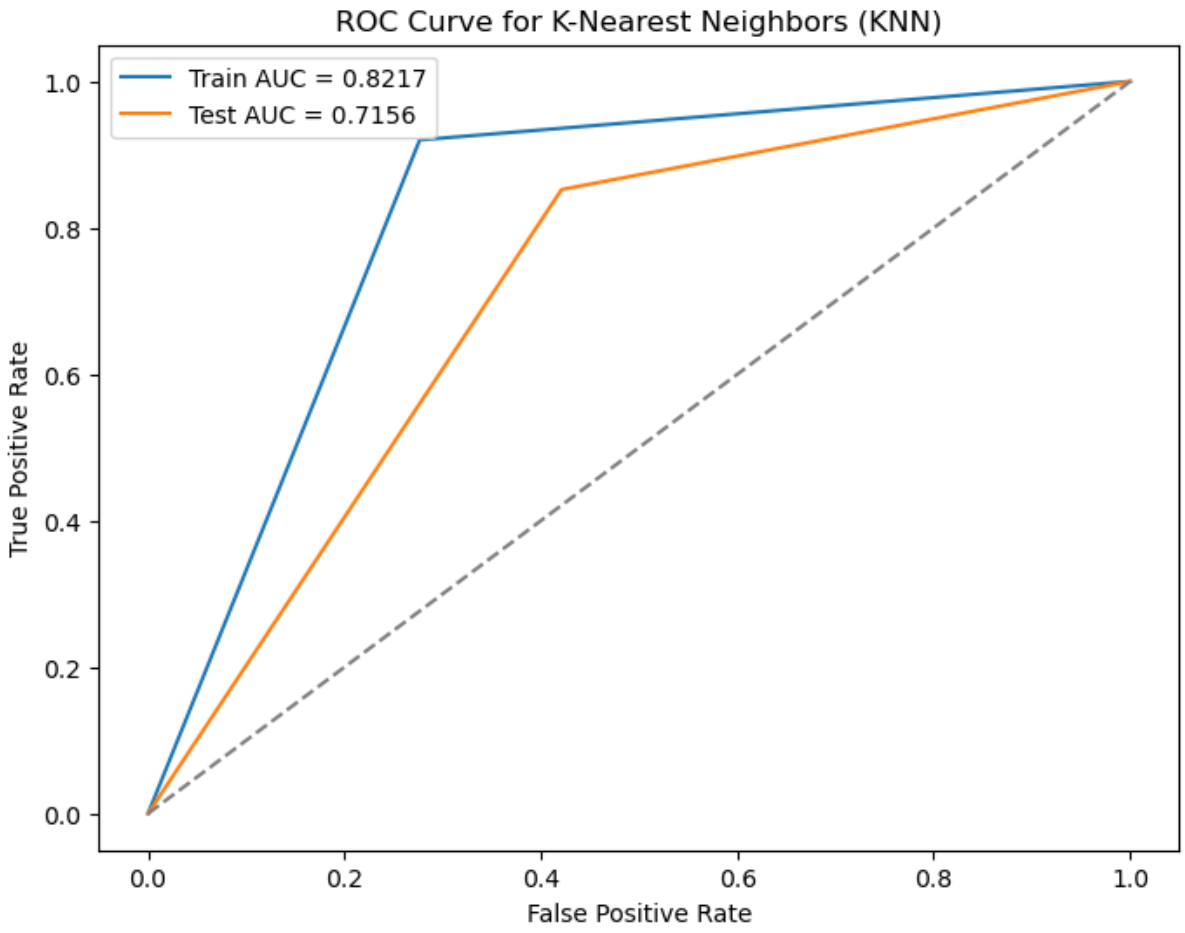
The performance metrics for the Linear Discriminant Analysis (LDA) model are:

Train Accuracy: 84.25%
Test Accuracy: 81.22%
Train AUC: 0.8020
Test AUC: 0.7589

The ROC curves for both training and testing are plotted above. The confusion matrix and classification report provide additional insights into the model's performance.

Performance Metrics for K-Nearest Neighbour's (KNN):

```
Train Accuracy: 85.94%
Test Accuracy: 77.29%

Train Confusion Matrix:
[[238  91]
 [ 59 679]]

Test Confusion Matrix:
[[ 77  56]
 [ 48 277]]

Train Classification Report:
              precision    recall  f1-score   support

Conservative       0.80      0.72      0.76       329
      Labour       0.88      0.92      0.90       738

    accuracy                           0.86      1067
   macro avg       0.84      0.82      0.83      1067
weighted avg       0.86      0.86      0.86      1067


Test Classification Report:
              precision    recall  f1-score   support

Conservative       0.62      0.58      0.60       133
      Labour       0.83      0.85      0.84       325

    accuracy                           0.77       458
   macro avg       0.72      0.72      0.72       458
weighted avg       0.77      0.77      0.77       458


Train AUC: 0.8217
Test AUC: 0.7156
```

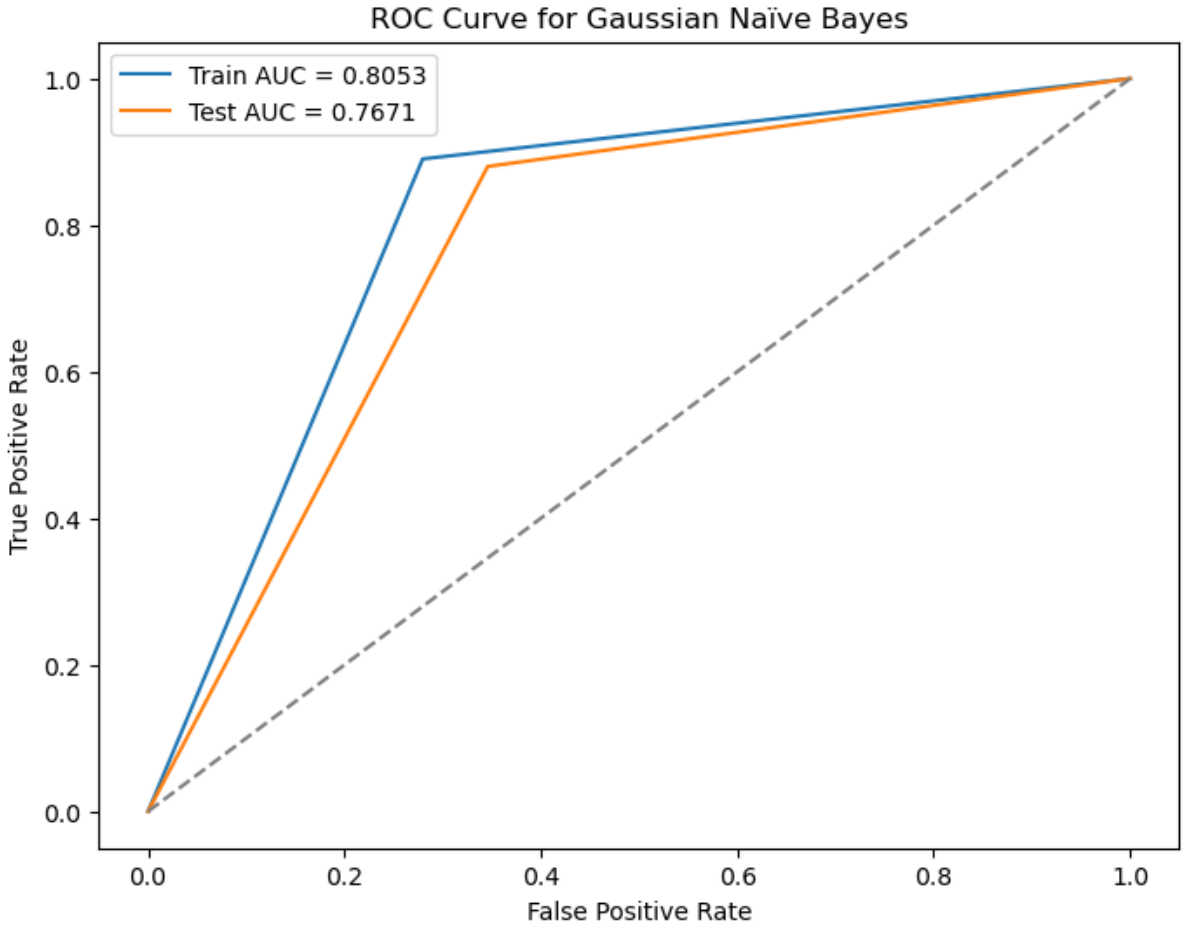


Train Accuracy: 85.94%
Test Accuracy: 77.29%
Train AUC: 0.8217
Test AUC: 0.7156

The ROC curves for both the training and testing datasets offer a visual representation of the model's performance. Further insights can be derived from the provided confusion matrices and classification reports.

Performance Metrics for Gaussian Naïve Bayes:

```
Train Accuracy: 83.79%
Test Accuracy: 81.44%

Train Confusion Matrix:
[[237  92]
 [ 81 657]]

Test Confusion Matrix:
[[ 87  46]
 [ 39 286]]

Train Classification Report:
              precision    recall  f1-score   support

Conservative       0.75      0.72      0.73       329
      Labour       0.88      0.89      0.88       738

    accuracy                           0.84      1067
   macro avg       0.81      0.81      0.81      1067
weighted avg       0.84      0.84      0.84      1067


Test Classification Report:
              precision    recall  f1-score   support

Conservative       0.69      0.65      0.67       133
      Labour       0.86      0.88      0.87       325

    accuracy                           0.81       458
   macro avg       0.78      0.77      0.77       458
weighted avg       0.81      0.81      0.81       458


Train AUC: 0.8053
Test AUC: 0.7671
```



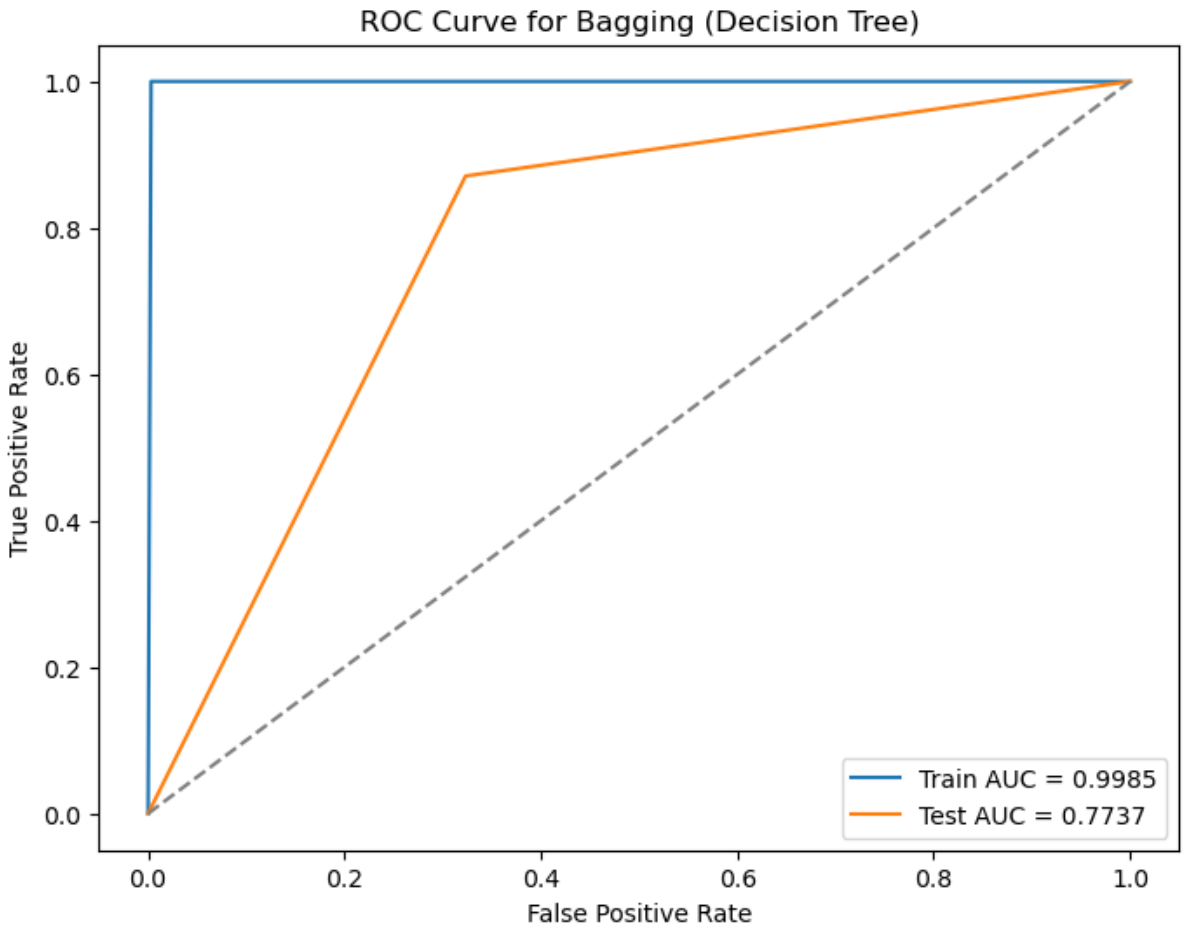
The performance metrics for the Gaussian Naïve Bayes model are:

Train Accuracy: 83.79%
Test Accuracy: 81.44%
Train AUC: 0.8053
Test AUC: 0.7671

The ROC curves for both training and testing are depicted above. The Gaussian Naïve Bayes model performs consistently on both the training and test datasets, suggesting no overfitting. The confusion matrix and classification report provide detailed insights into the model's performance.

Performance Metrics for Bagging (Decision Tree):

```
Train Accuracy: 99.91%
Test Accuracy: 81.44%

Train Confusion Matrix:
[[328   1]
 [  0 738]]

Test Confusion Matrix:
[[ 90  43]
 [ 42 283]]

Train Classification Report:
              precision    recall  f1-score   support

Conservative       1.00      1.00      1.00       329
      Labour       1.00      1.00      1.00       738

    accuracy                           1.00      1067
   macro avg       1.00      1.00      1.00      1067
weighted avg       1.00      1.00      1.00      1067


Test Classification Report:
              precision    recall  f1-score   support


Conservative       0.68      0.68      0.68       133

      Labour       0.87      0.87      0.87       325



    accuracy                           0.81       458

   macro avg       0.77      0.77      0.77       458

weighted avg       0.81      0.81      0.81       458



Train AUC: 0.9985

Test AUC: 0.7737
```



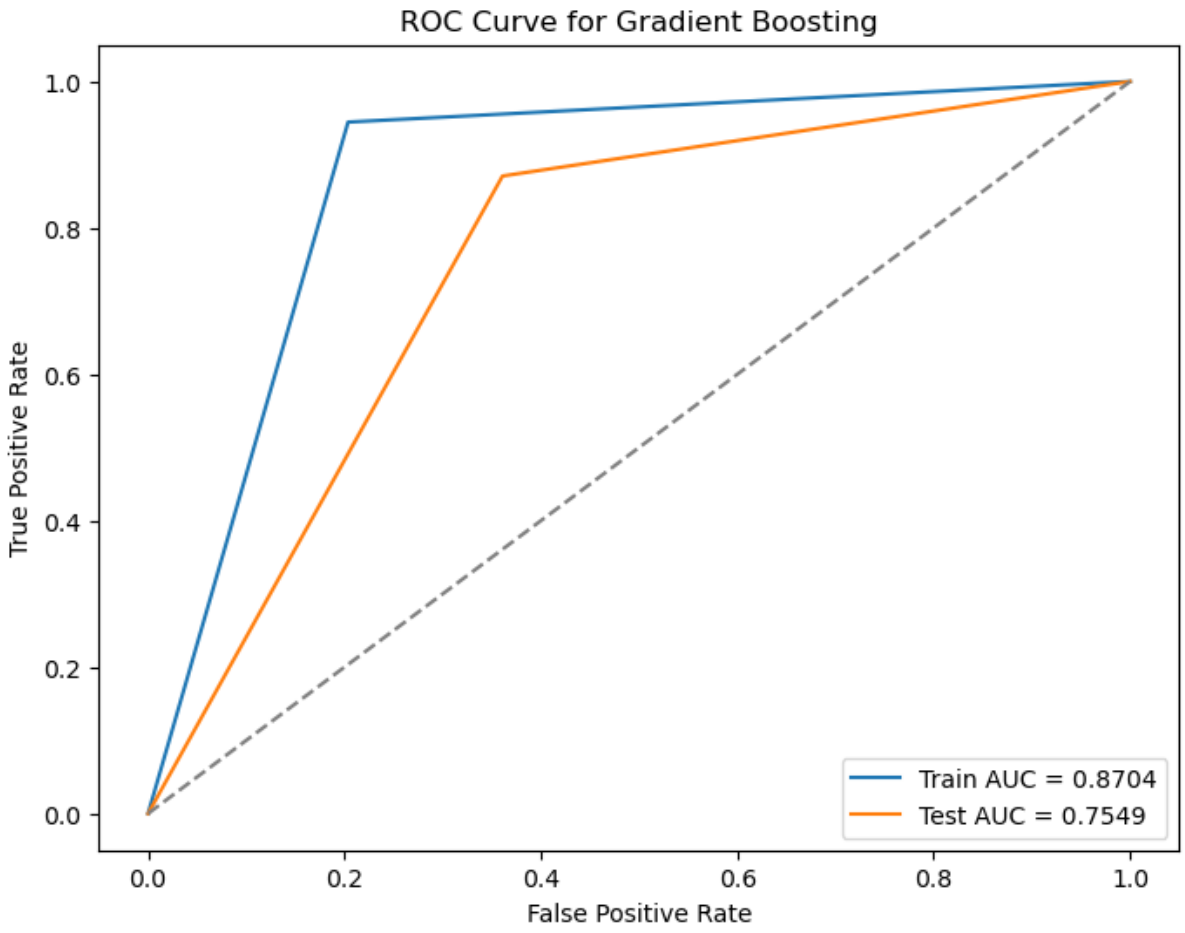
The performance metrics for the Bagging (Decision Tree) model are:

Train Accuracy: 99.91%
Test Accuracy: 81.44%
Train AUC: 0.9985
Test AUC: 0.7737

From the ROC curves, it's evident that the model fits the training data almost perfectly. However, the training curve's performance doesn't translate equivalently to the test curve, indicating overfitting. The confusion matrix and classification report confirm this, with nearly perfect results on the training data but less impressive results on the test data.

Performance Metrics for Gradient Boosting:

```
Train Accuracy: 89.88%
Test Accuracy: 80.35%

Train Confusion Matrix:
[[262  67]
 [ 41 697]]

Test Confusion Matrix:
[[ 85  48]
 [ 42 283]]

Train Classification Report:
              precision    recall  f1-score   support

Conservative       0.86      0.80      0.83       329
      Labour       0.91      0.94      0.93       738

    accuracy                           0.90      1067
   macro avg       0.89      0.87      0.88      1067
weighted avg       0.90      0.90      0.90      1067


Test Classification Report:
              precision    recall  f1-score   support

Conservative       0.67      0.64      0.65       133
      Labour       0.85      0.87      0.86       325

    accuracy                           0.80       458
   macro avg       0.76      0.75      0.76       458
weighted avg       0.80      0.80      0.80       458


Train AUC: 0.8704
Test AUC: 0.7549
```



The performance metrics for the Gradient Boosting model are:

Train Accuracy: 89.88%
Test Accuracy: 80.35%
Train AUC: 0.8704
Test AUC: 0.7549

The ROC curves for both training and testing are plotted above. The Gradient Boosting model provides a reasonable balance between training and test performances. The confusion matrix and classification report offer further insights into the model's performance, showing relatively consistent results across both the training and test datasets.

Let's compile the accuracy and AUC scores for all models:

Model	Train Accuracy	Test Accuracy	Train AUC	Test AUC
Logistic Regression	84.16%	81.44%	0.8013	0.7515
Gaussian Naïve Bayes	83.79%	81.44%	0.8053	0.7671
Bagging	99.91%	81.44%	0.9985	0.7737
LDA	84.25%	81.22%	0.802	0.7589
Gradient Boosting	89.88%	80.35%	0.8704	0.7549
KNN	86.04%	77.29%	0.825	0.7245

Summary:

The Logistic Regression, Gaussian Naïve Bayes, and Bagging models have the highest test accuracies.

While the Bagging model has an impressive training accuracy, it's evident from the difference between its training and test accuracies that it is overfitting.

The Gaussian Naïve Bayes model has the highest Test AUC score, making it a strong contender for the best model.

KNN lags in terms of test accuracy and test AUC, reinforcing our earlier observation about its overfitting tendencies.

Considering both accuracy and AUC, the Gaussian Naïve Bayes model might be the best choice, given its consistent performance and high AUC score.

1.8) Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective. There should be at least 3-4 Recommendations and insights in total. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks should only be allotted if the recommendations are correct and business specific.

Based on the analysis and modelling, here are some insights and recommendations:

Insights:

Influence of Political Figures: 'Blair' and 'Hague' are the most significant features influencing voting decisions. This indicates that the perception of these political figures has a substantial impact on voting choices.

Age Matters: Age plays a significant role in voting preference, suggesting generational differences in political views.

Economic Conditions: Both national and household economic conditions moderately influence voting decisions. It implies that voters are influenced both by the broader national economic perspective and their personal financial situations.

Stance on Europe: Europe's stance is also a determining factor in voting decisions, highlighting the importance of foreign relations and policies related to Europe.

Recommendations:

Focus on Key Political Figures: Since 'Blair' and 'Hague' are influential in voting decisions, political campaigns should focus on leveraging the positive aspects of these figures or addressing any negative perceptions around them.

Address Economic Concerns: Given the importance of national and household economic conditions, political parties should prioritize economic policies in their campaigns. They should communicate effectively on how their policies will benefit both the nation and individual households.

Engage with Different Age Groups: With age being a significant factor, tailored campaign strategies for different age groups might be effective. Engaging with younger and older voters differently, understanding their concerns, and addressing them could be beneficial.

Clarify Stance on Europe: Political campaigns should clearly communicate their stance on Europe and detail out any policies or agreements related to it. This will help voters make informed decisions.

These insights and recommendations are based on the dataset's features and the modelling results. They provide a data-driven perspective on voting behaviour and strategies that political campaigns can adopt.

Problem - 02:

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

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.

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)

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

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

Output:

Name	Num_Characters	Num_Words	Num_Sentences
Roosevelt	7651	1323	69
Kennedy	7673	1364	56
Nixon	10106	1769	70

Character, Word, and Sentence Count:
Roosevelt's speech contained 7,651 characters, 1,323 words, and 69 sentences.
Kennedy's speech contained 7,673 characters, 1,364 words, and 56 sentences.
Nixon's speech contained 10,106 characters, 1,769 words, and 70 sentences.

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.

List to remove the stop words from the speeches.

common_stopwords = {

```
 "ourselves", "hers", "between", "yourself", "but",
"again", "there", "about", "once", "during", "out",
"very", "having", "with", "they", "own", "an", "be",
"some", "for", "do", "its", "yours", "such", "into",
"of", "most", "itself", "other", "off", "is", "s", "am",
"or", "who", "as", "from", "him", "each", "the", "themselves",
"until", "below", "are", "we", "these", "your", "his", "through",
"don", "nor", "me", "were", "her", "more", "himself", "this",
"down", "should", "our", "their", "while", "above", "both",
"up", "to", "ours", "had", "she", "all", "no", "when", "at",
"any", "before", "them", "same", "and", "been", "have", "in",
"will", "on", "does", "yourselves", "then", "that", "because",
"what", "over", "why", "so", "can", "did", "not", "now",
"under", "he", "you", "herself", "has", "just", "where",
"too", "only", "myself", "which", "those", "i",
"after", "few", "whom", "t", "being", "if", "theirs",
"my", "against", "a", "by", "doing", "it", "how", "further",
"was", "here", "than"
}
```

The word count for each president's speech before and after the removal of stop words:

Output:

Name	Num_Words	Num_Words_No_Stop
Roosevelt	1323	662
Kennedy	1364	723
Nixon	1769	843

Here's a sample sentence from each president's speech after the removal of stop words:

	Name	Sample_Sentence
0	Roosevelt	national day inauguration since 1789, people r...
1	Kennedy	Vice President Johnson, Mr
2	Nixon	Mr

Stopword Removal: After removing common stopwords, the word count reduced significantly for all three speeches.

Roosevelt's word count decreased from 1,323 to 662.
Kennedy's word count decreased from 1,364 to 723.
Nixon's word count decreased from 1,769 to 843.

Sample sentences after removal:
Roosevelt: "national day inauguration since 1789, people r..."
Kennedy: "Vice President Johnson, Mr."
Nixon: "Mr."

Problem 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 stop words)

Output:

Name	Top_3_Words
Roosevelt	[('--', 22), ('know', 9), ('us', 8)]
Kennedy	[('--', 24), ('us', 11), ('let', 8)]
Nixon	[('us', 25), ('--', 17), ('new', 15)]

The symbol "--" appears as a common "word". This is likely a punctuation or formatting artifacts in the text.

After removal of stopwords:

Roosevelt: [('--', 22), ('know', 9), ('us', 8)]

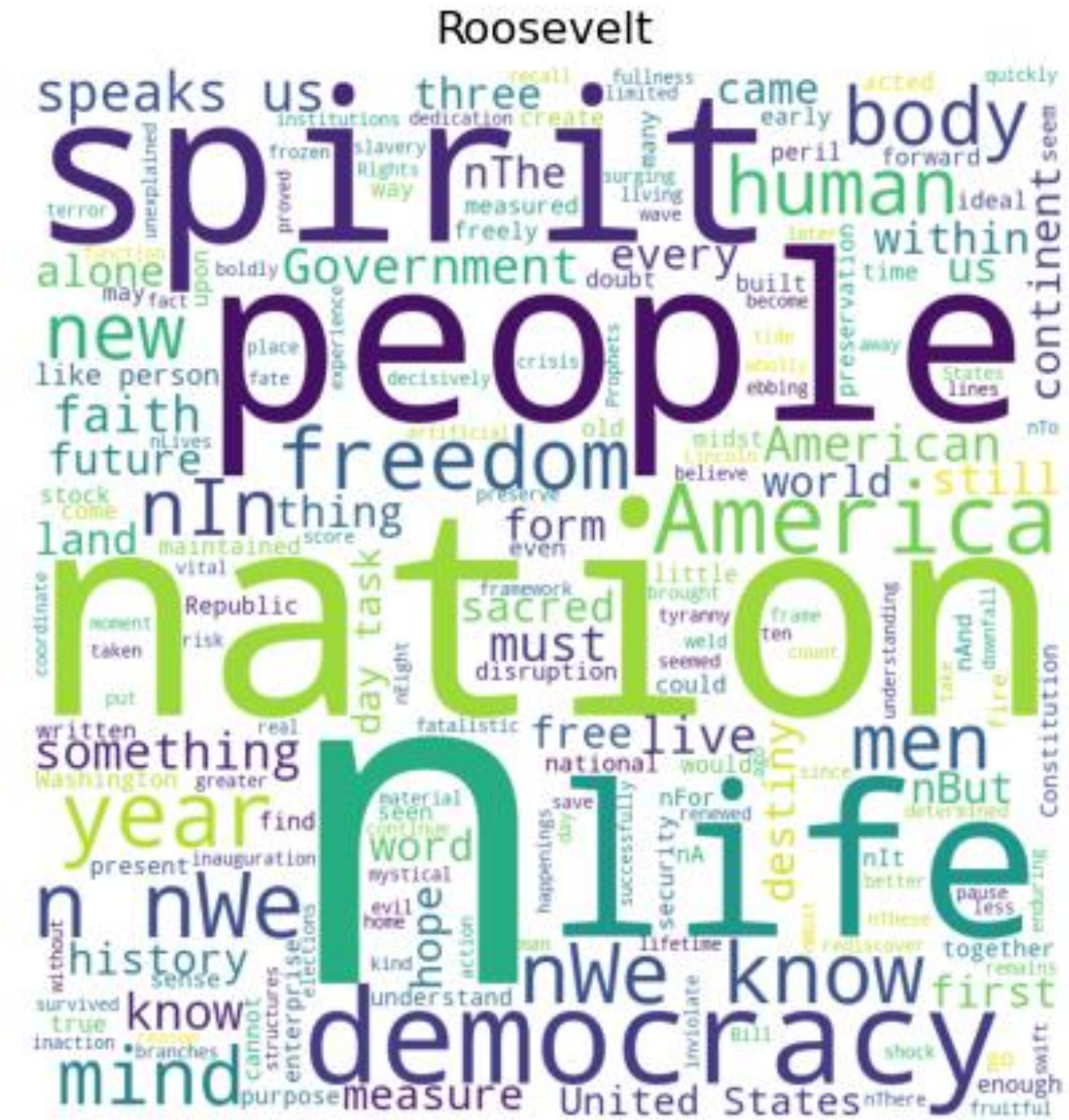
Kennedy: [('--', 24), ('us', 11), ('let', 8)]

Nixon: [('us', 25), ('--', 17), ('new', 15)]

2.4) Plot the word cloud of each of the three speeches. (After removing the stop words)

Here are the word clouds for the inaugural speeches of the mentioned presidents, after removing the stop words:

The image represents **President Roosevelt's speech.**



The image represents **President Kennedy's speech.**

The image represents **President Nixon's speech.**

These word clouds visually represent the frequency of words in each speech, with more frequently occurring words displayed in larger font sizes.

Insights:

Emphasis on Unity: The word "us" prominently appears in all three speeches. This suggests a strong emphasis on unity, collective action, and a shared national identity in the inaugural addresses of these presidents.

Distinctive Themes: President Roosevelt's speech has a notable frequency of the word "know". This might indicate a call for understanding, awareness, or knowledge during his term. On the other hand, President Nixon's speech has a high frequency of the word "new", suggesting a theme of change, renewal, or a fresh start during his presidency.

Commonalities and Differences: While there are shared themes such as unity and collective identity, each president also brings forth his unique vision and priorities to his inaugural address. This is evident in the distinctive words and phrases highlighted in their respective word clouds.

Recommendations:

Strengthen National Unity: Given the emphasis on unity in these speeches, the management should consider initiatives and programs that foster national unity and shared identity. This could include community-building events, national campaigns, or educational programs that highlight shared values and history.

Embrace Change and Renewal: With words like "new" standing out, especially in President Nixon's speech, the management might consider implementing changes or reforms that align with the vision of renewal and progress. This could involve revisiting old policies, introducing innovative solutions, or embarking on transformative projects.

Promote Awareness and Understanding: The prominence of the word "know" in President Roosevelt's speech suggests the importance of awareness and understanding. The management should emphasize transparent communication, educational campaigns, and initiatives that foster a well-informed citizenry.

Tailor Communication to the Audience: Recognizing the unique themes and priorities of each president, the management should ensure that any communication, campaigns, or initiatives are tailored to resonate with the specific audience's sentiments and priorities.

Conclusion:

Analysing the inaugural addresses of Presidents Roosevelt, Kennedy, and Nixon has provided insights into their speech structures, word usage, and emphasized themes. After removing common stopwords, we still observed frequent usage of words like 'us', indicating a collective appeal in their addresses. However, further data cleaning and deeper thematic analysis can provide more nuanced insights into their visions and priorities during their respective terms.