Machine Learning Project :

-Prapthi Pandian

Table of Contents

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.

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

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

Shape of the dataset:

```
No. of rows:  1525
No. of columns:  10
```

Data info-

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 10 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   Unnamed: 0               1525 non-null   int64 
 1   vote                     1525 non-null   object
 2   age                      1525 non-null   int64 
 3   economic.cond.national   1525 non-null   int64 
 4   economic.cond.household  1525 non-null   int64 
 5   Blair                    1525 non-null   int64 
 6   Hague                    1525 non-null   int64 
 7   Europe                   1525 non-null   int64 
 8   political.knowledge      1525 non-null   int64 
 9   gender                   1525 non-null   object
dtypes: int64(8), object(2)
memory usage: 119.3+ KB
```

There are 10 variables in the dataset. 2 categorical and 8 numeric variables of int datatype.

```
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
```

There seems to be **no null values** in the dataset.

We have removed the “Unnamed: 0” column from the dataset as it represents the index of the data and is of no value for our analysis.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   vote                     1525 non-null   object
 1   age                      1525 non-null   int64 
 2   economic.cond.national   1525 non-null   int64 
 3   economic.cond.household  1525 non-null   int64 
 4   Blair                    1525 non-null   int64 
 5   Hague                    1525 non-null   int64 
 6   Europe                   1525 non-null   int64 
 7   political.knowledge      1525 non-null   int64 
 8   gender                   1525 non-null   object
dtypes: int64(7), object(2)
memory usage: 107.4+ KB
```

Printing the categorical and Numerical columns in the dataset-

```
Categorical columns: ['vote', 'gender']
Numeric columns: ['age', 'economic.cond.national', 'economic.cond.household', 'Blair', 'Hague', 'Europe', 'political.knowledg
e']
```

Descriptive statistics of Numerical columns in DataFrame-

	count	mean	std	min	25%	50%	75%	max
age	1525.0	54.182295	15.711209	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	3.245902	0.880969	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	3.140328	0.929951	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	3.334426	1.174824	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	2.746885	1.230703	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	6.728525	3.297538	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	1.542295	1.083315	0.0	0.0	2.0	2.0	3.0

- Age ranges from 24 to 93.
- Average age is around 54 with a standard deviation of approximately 15.7
- The distribution seems somewhat symmetric, with the median (50th percentile) at 53.
- economic.cond.national & economic.cond.household variables represent assessments of economic conditions, ranging from 1 to 5.
- The mean for both is slightly above 3, indicating moderate conditions.
- Blair & Hague represent assessments of political leaders, ranging from 1 to 5.
- Europe represents an 11-point scale measuring attitudes toward European integration with an average score around 6.7 and standard deviation around 3.3.
- political.knowledge represents knowledge of parties positions on European integration on a scale of 0 to 3.
- The mean is approximately 1.54, indicating a moderate level of knowledge on an average.

	count	unique	top	freq
vote	1525	2	Labour	1063
gender	1525	2	female	812

- 'Labour' is the dominant category, 1063 out of 1525.
- There are more females in the dataset, 812 out of 1525.

Skewness-

It is a statistical measure that describes the distribution of data points in the dataset.

```
Skewness of variables:
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
```

- vote, age, Hague, gender are positively skewed
- economic.cond.national, economic.cond.household, Blair, Europe, political.knowledge are negatively skewed.

Duplicate values-

```
Number of duplicate rows = 8
```

	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

Post dropping the duplicate values,

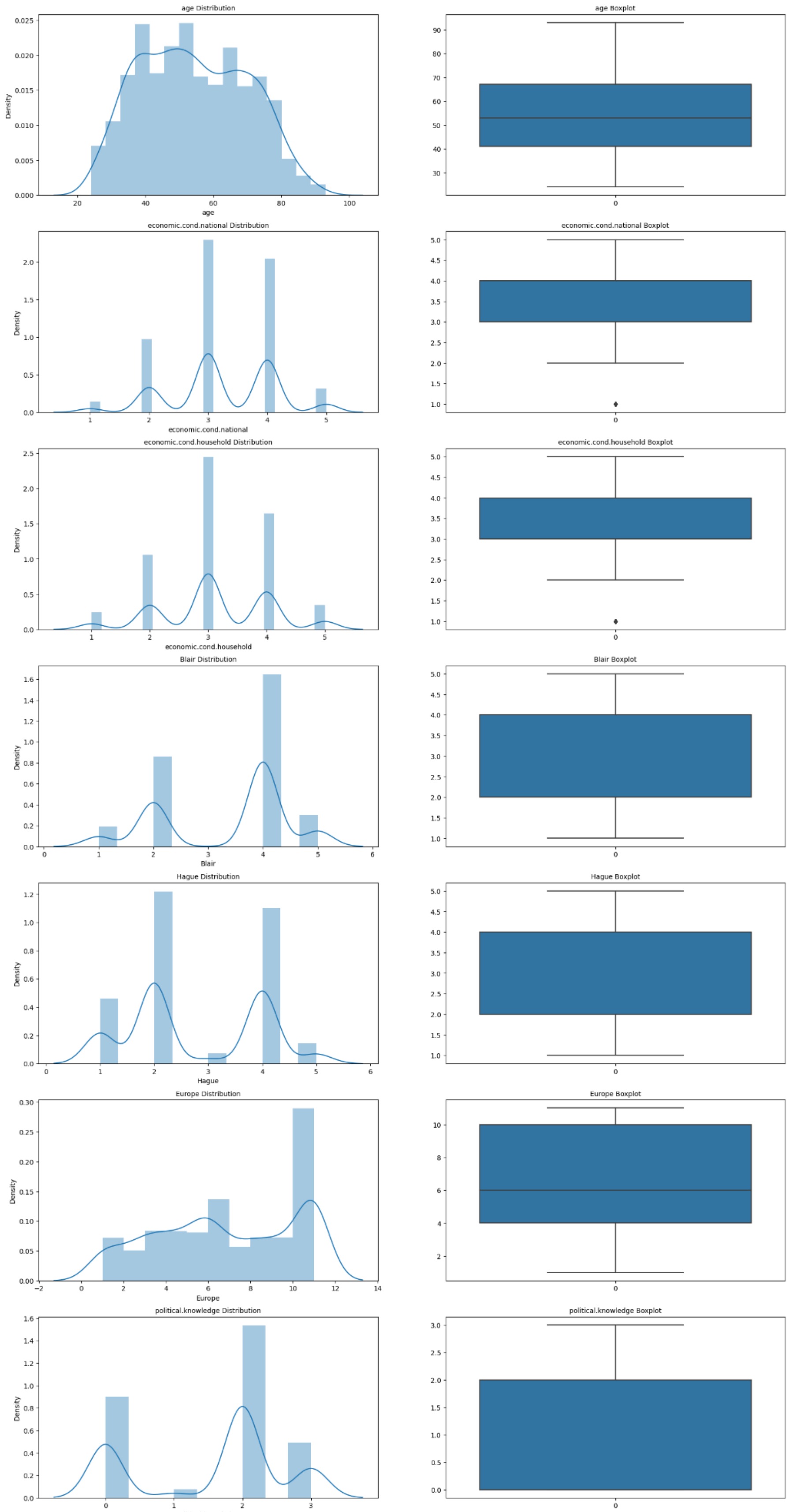
```
Number of duplicate rows = 0
```

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

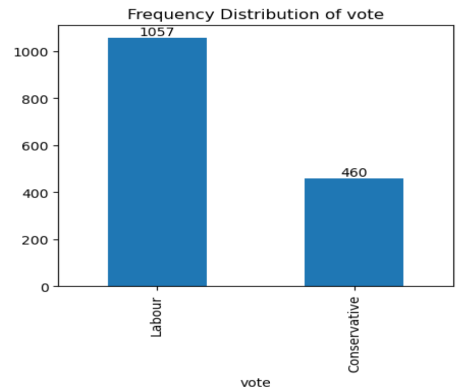
Printing unique values-

```
VOTE :  2
vote
Conservative     460
Labour          1057
Name: count, dtype: int64


GENDER :  2
gender
male      709
female    808
Name: count, dtype: int64
```

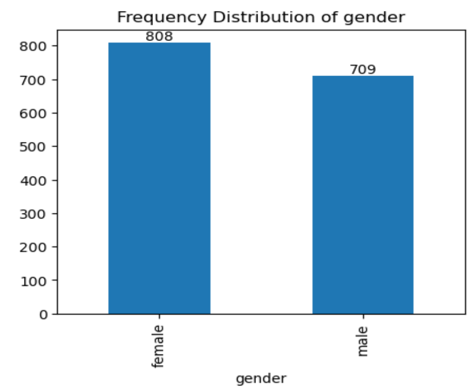
1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.


age Distribution
Density
age
age Boxplot
economic.cond.national Distribution
Density
economic.cond.national
economic.cond.national Boxplot
economic.cond.household Distribution
Density
economic.cond.household
economic.cond.household Boxplot
Blair Distribution
Density
Blair
Blair Boxplot
Hague Distribution
Density
Hague
Hague Boxplot
Europe Distribution
Density
Europe
Europe Boxplot
political.knowledge Distribution
Density
political.knowledge
political.knowledge Boxplot

```
Details of vote
----------------------------------------------------------------
vote
Labour          1057
Conservative     460
Name: count, dtype: int64
```

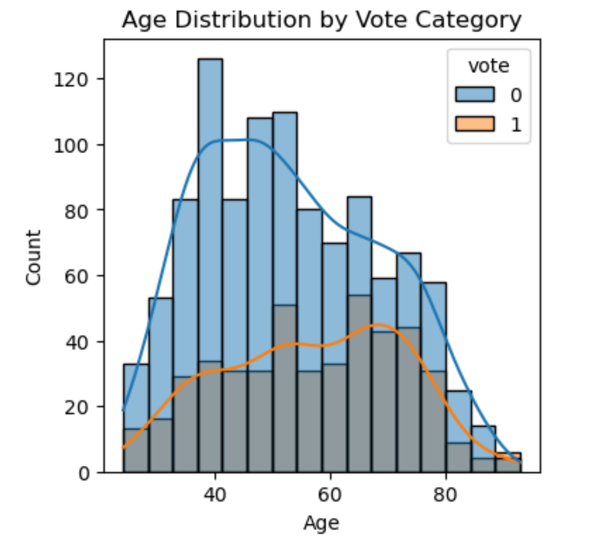



```
Details of gender
----------------------------------------------------------------
gender
female    808
male      709
Name: count, dtype: int64
```

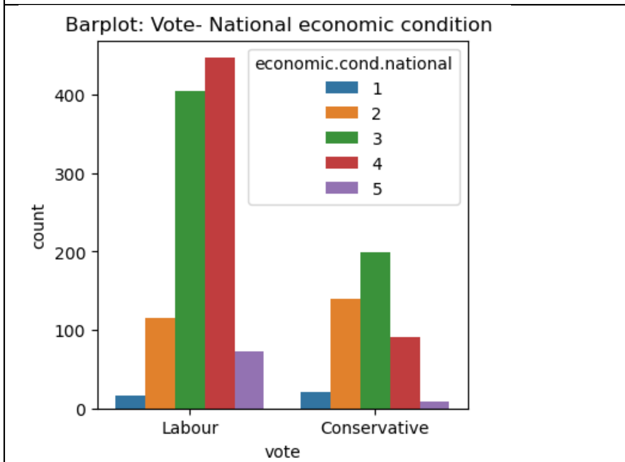


- There are more number of votes for Labour party
- And based on the gender, the male proportion has casted less number of votes in comparison to the females.

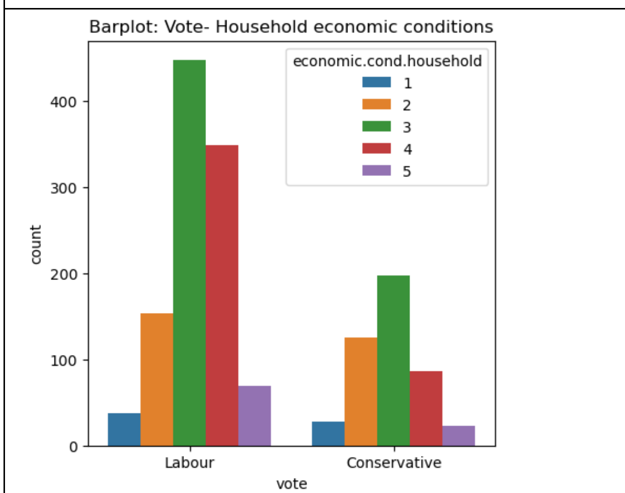
Bivariate & Multivariate Analysis-

<table>
<tr>
<td>
</td>
<td>

- Majority of people casting votes are aged between 35-55.

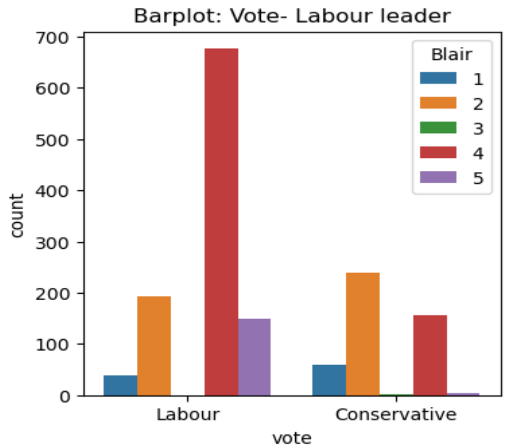
</td>
</tr>
<tr>
<td>
</td>
<td>

- Participants who have voted for the Labour party has rated the current national economic condition to a score of 4 followed by 3 which means it is quite good.
- Participants who have voted for the Conservative party has rated the current national economic condition to a score of 3 followed by 2 which means it is moderate.

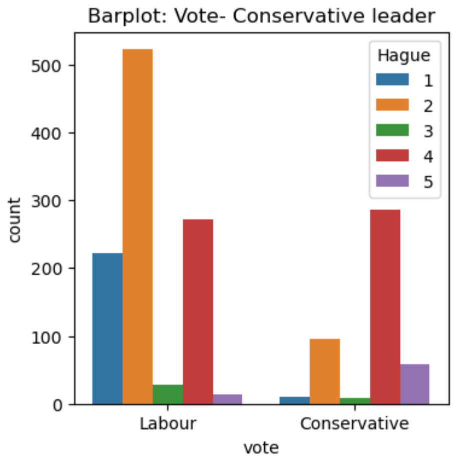
</td>
</tr>
<tr>
<td>
</td>
<td>

- Participants who have voted for the Labour party has rated the current household economic condition to a major score of 3 followed by 4 which means it is quite good.
- Participants who have voted for the Conservative party has rated the current household economic condition to a major score of 3 followed by 2 which means it is moderate.

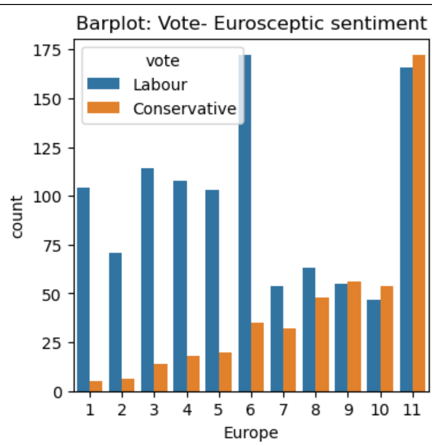
</td>
</tr>
</table>



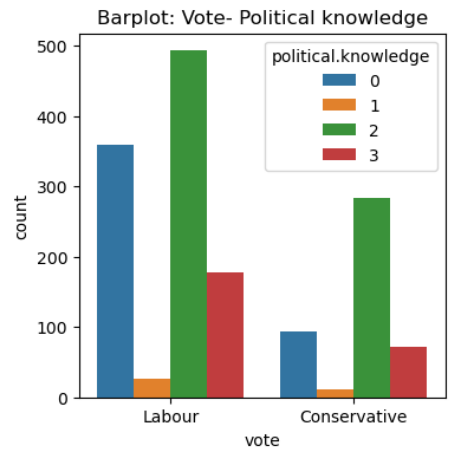
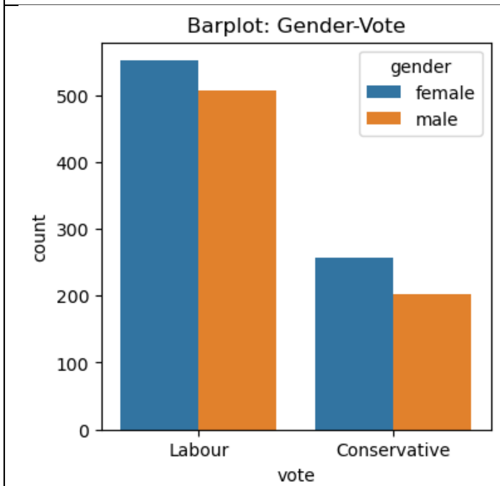
- Participants who have voted for the Labour party has rated the Labour leader with a major score of 4.
- Participants who have voted for the Conservative party has rated the Labour leader with a major score of 2 followed by 4 which reflects their difference in opinion.

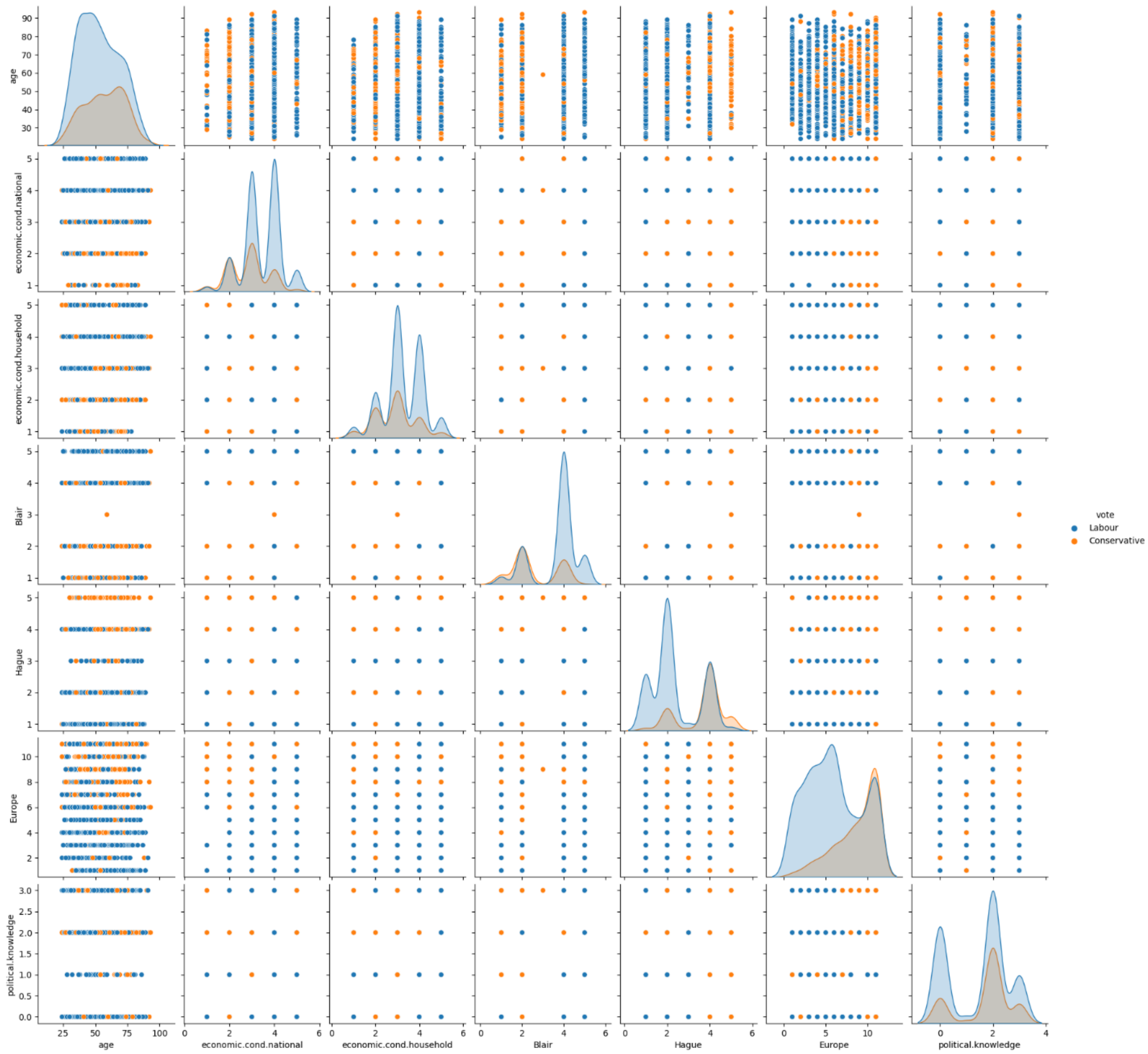


- Participants who have voted for the Labour party has rated the Conservative leader with a major score of 2 and the score of 5 the least.
- Participants who have voted for the Conservative party has rated the Conservative leader with a major score of 4.

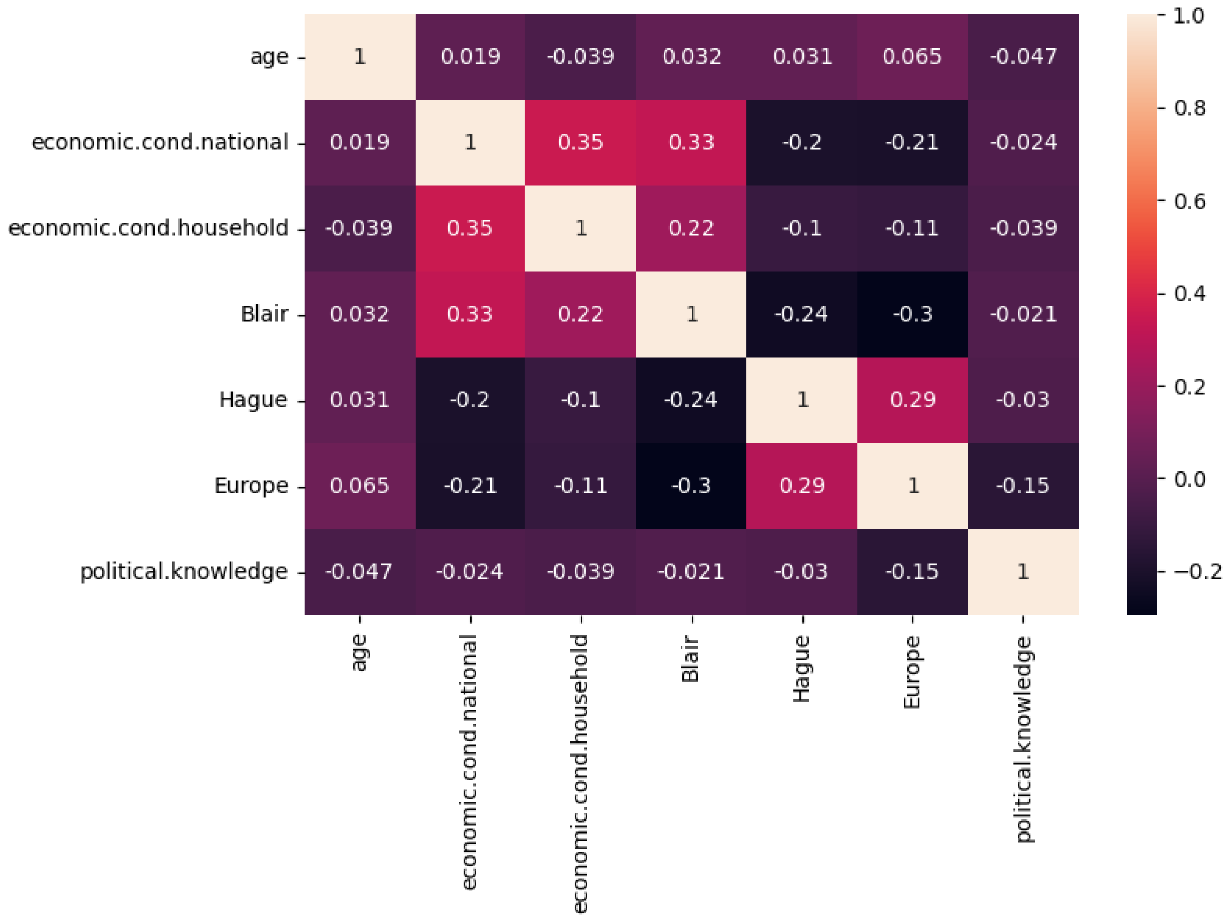


- Many members who've voted for the Conservative party have high 'Eurosceptic' sentiment. While those voted for Labour party have mixed attitudes toward European integration.

	• Irrespective of the parties the votes are casted to, the participants have a moderate level of knowledge on the parties' positions on European integration with a major score of 2.
	• Female votes are higher and many of them have voted for Labour party.

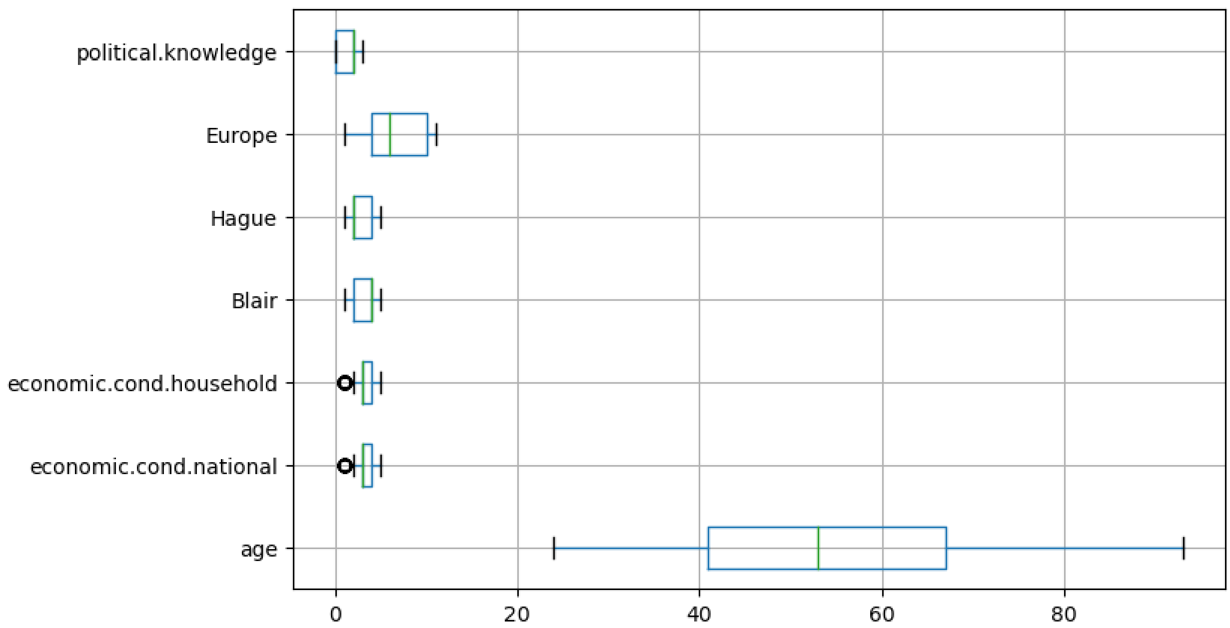


age
economic.cond.national
economic.cond.household
Blair
Hague
Europe
political.knowledge
vote
Labour
Conservative



- There is high correlation between economic.cond.national and economic.cond.household i.e. there is a strong relationship between how people view current national economic conditions and current household economic conditions.
- Also, assessment of the Labour leader (Blair) is correlated to assessment of current national economic and household conditions. Individuals opinions about the Labour leader are related to their views on the national and household economic conditions.
- Whereas, assessment of the Conservative leader (Hague) is highly correlated to the Eurosceptic' sentiment- There is a strong relation between perceptions of the Conservative leader and attitudes toward European integration. This suggests that individuals who hold Eurosceptic sentiments might also have opinions aligned with the assessment of the Conservative leader.

Outliers-



There are outliers present in “economic.cond.national” and “economic.cond.household’ variables.

Since these variables represent assessments on an expected scale of 1 to 5 and have meaningful interpretations, we are not treating them.

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).

Converting all objects to categorical codes and changing their datatype to int.

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

```
<class 'pandas.core.frame.DataFrame'>
Index: 1517 entries, 0 to 1524
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   vote                     1517 non-null   int64
 1   age                      1517 non-null   int64
 2   economic.cond.national   1517 non-null   int64
 3   economic.cond.household  1517 non-null   int64
 4   Blair                    1517 non-null   int64
 5   Hague                    1517 non-null   int64
 6   Europe                   1517 non-null   int64
 7   political.knowledge      1517 non-null   int64
 8   gender                   1517 non-null   int64
dtypes: int64(9)
memory usage: 150.8 KB
```

For the categorical variable that is nominal (gender), we have performed dummy variable encoding.

Sample data set post data encoding-

	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_2
0	0	43	3	3	4	1	2	2	False
1	0	36	4	4	4	4	5	2	True
2	0	35	4	4	5	2	3	2	True
3	0	24	4	2	2	1	4	0	False
4	0	41	2	2	1	1	6	2	True

We are not going to scale the data for Logistic Regression, LDA and other models. But, for KNN it is necessary to scale the data as it is distance-based algorithm. So, we will be scaling while building KNN model to have an equal weightage of all variables.

We have split the data into train and test sets in a 70:30 ratio. Here the target variable is “vote”

Class 0- Labour voters

Class 1- Conservative voters

Train value counts-

```
vote
0    0.71065
1    0.28935
Name: proportion, dtype: float64
```

Test value counts-

```
vote
0    0.664474
1    0.335526
Name: proportion, dtype: float64
```

```
Number of rows and columns of the training set for the independent variables: (1061, 8)
Number of rows and columns of the training set for the dependent variable: (1061,)
Number of rows and columns of the test set for the independent variables: (456, 8)
Number of rows and columns of the test set for the dependent variable: (456,)
```

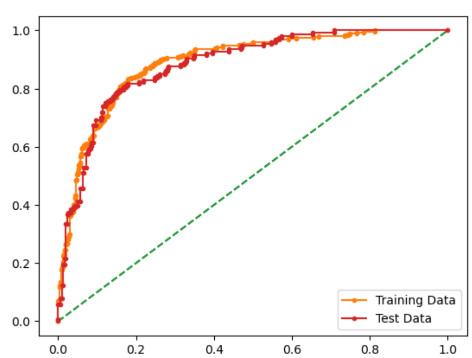
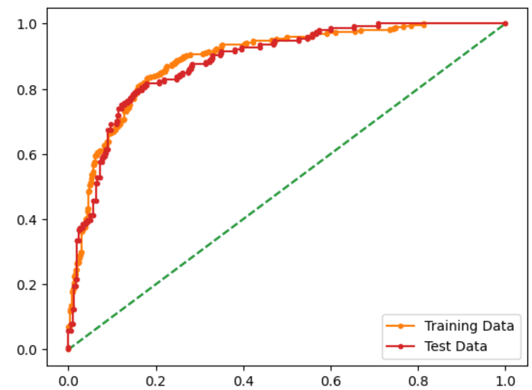
1.4 Apply Logistic Regression and LDA (linear discriminant analysis).

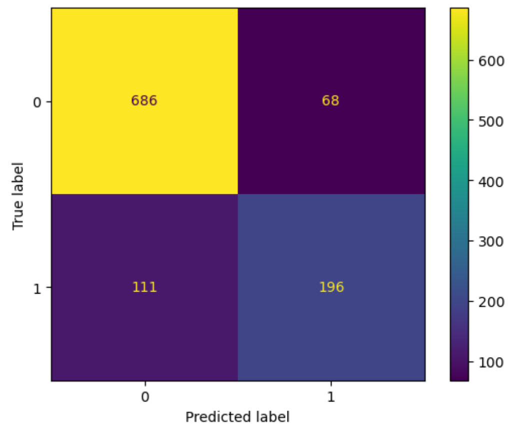
- **Accuracy-** How accurately the model classifies the data point. More the accuracy, lesser the false pr edictions.
- **Sensitivity/ Recall-** How many of actual True data points are identified as True data points by the m odel.
- **Precision-** Among the points identified as positive by the mode, how many are actual positives.
- **AUC score** represents degree/ measure of separability. i.e. how much the model is capable of distinguishing between classes. Value closer to 1 tells that there is good separability between the pr edicted classes and thus the model is good for prediction.
- **ROC Curve –** For visualizing the classifier performance. Steeper the ROC curve, stronger the model.
- **F1 score** helps to know if Type 1 / Type 2 error is high/ low on average.

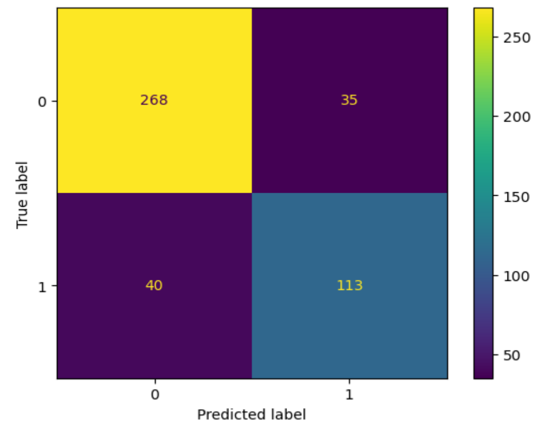
Logistic Regression Model-

```
LogisticRegression
LogisticRegression(max_iter=10000, n_jobs=2, penalty='none', solver='newton-cg',
                   verbose=True)
```

	Train dataset	Test dataset
Probability Prediction	array([1, 0, 0, ..., 0, 0, 0], dtype=int64)	array([0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0], dtype=int64)

<table>
<tr>
<td>Predicted class probabilities</td>
<td>
<table>
<tr><th></th><th>0</th><th>1</th></tr>
<tr><td>0</td><td>0.068175</td><td>0.931825</td></tr>
<tr><td>1</td><td>0.903016</td><td>0.096984</td></tr>
<tr><td>2</td><td>0.701584</td><td>0.298416</td></tr>
<tr><td>3</td><td>0.889790</td><td>0.110210</td></tr>
<tr><td>4</td><td>0.982777</td><td>0.017223</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
<tr><td>1056</td><td>0.954885</td><td>0.045115</td></tr>
<tr><td>1057</td><td>0.639824</td><td>0.360176</td></tr>
<tr><td>1058</td><td>0.744179</td><td>0.255821</td></tr>
<tr><td>1059</td><td>0.759462</td><td>0.240538</td></tr>
<tr><td>1060</td><td>0.975849</td><td>0.024151</td></tr>
</table>
</td>
<td>
<table>
<tr><th></th><th>0</th><th>1</th></tr>
<tr><td>0</td><td>0.575716</td><td>0.424284</td></tr>
<tr><td>1</td><td>0.851574</td><td>0.148426</td></tr>
<tr><td>2</td><td>0.992813</td><td>0.007187</td></tr>
<tr><td>3</td><td>0.163650</td><td>0.836350</td></tr>
<tr><td>4</td><td>0.931593</td><td>0.068407</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
<tr><td>451</td><td>0.957914</td><td>0.042086</td></tr>
<tr><td>452</td><td>0.413026</td><td>0.586974</td></tr>
<tr><td>453</td><td>0.959415</td><td>0.040585</td></tr>
<tr><td>454</td><td>0.933940</td><td>0.066060</td></tr>
<tr><td>455</td><td>0.959544</td><td>0.040456</td></tr>
</table>
456 rows × 2 columns
</td>
</tr>
<tr><td>Accuracy</td><td>0.8312912346842601</td><td>0.8355263157894737</td></tr>
<tr><td>AUC</td><td>0.890</td><td>0.883</td></tr>
<tr>
<td>ROC</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>Confusion Matrix</td>
<td>[[686 68]
 [111 196]]</td>
<td>[[268 35]
 [40 113]]</td>
</tr>
</table>





Classification Report

	precision	recall	f1-score	support
0	0.86	0.91	0.88	754
1	0.74	0.64	0.69	307
accuracy			0.83	1061
macro avg	0.80	0.77	0.79	1061
weighted avg	0.83	0.83	0.83	1061

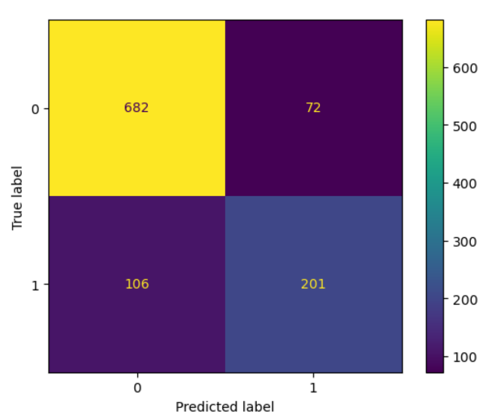
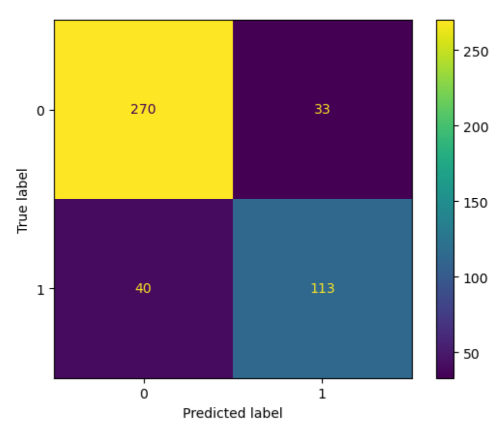
	precision	recall	f1-score	support
0	0.87	0.88	0.88	303
1	0.76	0.74	0.75	153
accuracy			0.84	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.84	0.83	456

Inference:

- The model performs well on both the training and test datasets.
- Accuracy is around 83-84%, indicating that it correctly predicts the party for approx. 83-84% of the voters in both the datasets.
- The model shows slightly better performance in predicting Labour voters (class 0) compared to Conservative voters (class 1) based on measures of precision, recall, and F1-scores for both classes.
- Overall, the model seems to generalize well on unseen data indicating a valid and reasonably fitting model.
- The tuned model will be represented in section 1.6

LDA Model-

```
LinearDiscriminantAnalysis
LinearDiscriminantAnalysis()
```

	Train dataset	Test dataset
Probability Prediction	array([1, 0, 0, ..., 0, 0, 0], dtype=int64)	array([0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0], dtype=int64)
Predicted Probabilities	array([[0.05924226, 0.94075774], [0.91577525, 0.08422475], [0.64423671, 0.35576329], [0.92839311, 0.07160689], [0.98495285, 0.01504715], [0.96699314, 0.03300686], [0.77283939, 0.22716061], [0.99700135, 0.00299865], [0.25640154, 0.74359846], [0.93676479, 0.06323521], [0.64333828, 0.35666172], [0.99554835, 0.00445165], [0.91233743, 0.08766257], [0.83372649, 0.16627351], [0.9276023 , 0.0723977], [0.51439571, 0.48560429], [0.40793647, 0.59206353], [0.18363411, 0.81636589], [0.48379779, 0.51620221], [0.85007954, 0.14992046]])	array([[0.52510043, 0.47489957], [0.85349146, 0.14650854], [0.9904768 , 0.0095232], [0.15607959, 0.84392041], [0.93014145, 0.06985855], [0.96201259, 0.03798741], [0.54351017, 0.45648983], [0.82379895, 0.17620105], [0.95277475, 0.04722525], [0.82764154, 0.17235846], [0.82436941, 0.17563059], [0.39715881, 0.60284119], [0.99132865, 0.00867135], [0.36635161, 0.63364839], [0.82820571, 0.17179429], [0.53886231, 0.46113769], [0.9230354 , 0.0769646], [0.95098558, 0.04901442], [0.07384744, 0.92615256], [0.74311711, 0.25688289]])
Confusion Matrix		

<table>
<tr><td>Classification report</td><td>Classification Report of the training data:

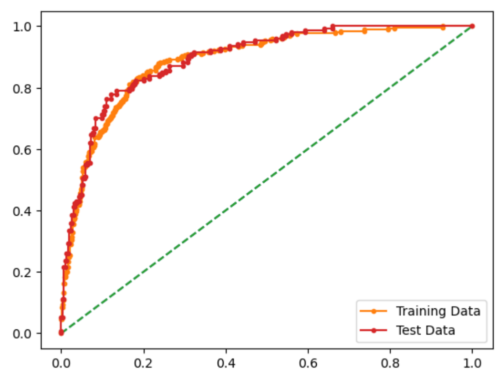
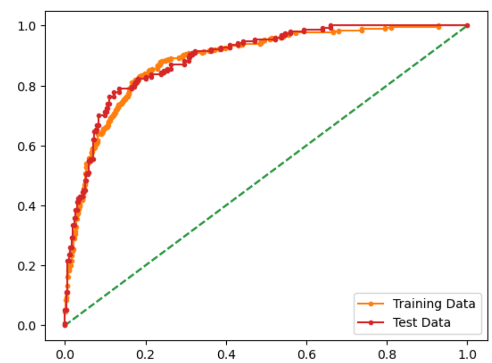
precision recall f1-score support

0 0.87 0.90 0.88 754
1 0.74 0.65 0.69 307

accuracy 0.83 1061
macro avg 0.80 0.78 0.79 1061
weighted avg 0.83 0.83 0.83 1061</td><td>Classification Report of the test data:

precision recall f1-score support

0 0.87 0.89 0.88 303
1 0.77 0.74 0.76 153

accuracy 0.84 456
macro avg 0.82 0.81 0.82 456
weighted avg 0.84 0.84 0.84 456</td></tr>
<tr><td>AUC</td><td>0.887</td><td>0.893</td></tr>
<tr><td>ROC</td><td>
</td><td>
</td></tr>
</table>

- The model's precision for both classes is relatively high, indicating a good ratio of correctly predicted classes.
- For class 0, the model captures around 90% (training) and 89% (test) correctly and for class 1, it's around 65% (training) and 74% (test).
- F1-score is higher for class 0 than class 1 in both training and test sets, indicating better performance in predicting class 0.
- Overall, the model appears valid, and is generalizing reasonably well to unseen data with no major biases towards either the training or test data making it a decent fit.

Intercept value-

This represents the estimated value of the response variable when all predictor variables are zero.

```
array([-3.526319])
```

Coefficients for LDF-

Coefficients represent each independent variables weight in Linear Discriminant Function.

```
array([[ 0.02348839, -0.42722063, -0.07376585, -0.76223865,  0.96233945,
         0.23141154,  0.50481148, -0.05819087]])
```

Rounded up coeff-

```
array([[ 0.02, -0.43, -0.07, -0.76,  0.96,  0.23,  0.5 , -0.06]])
```

```
Index(['age', 'economic.cond.national', 'economic.cond.household', 'Blair',
       'Hague', 'Europe', 'political.knowledge', 'gender_2'],
      dtype='object')
```

LDF for above model will be-

```
'\nLDF=(-3.526319)+ X1*0.02 + X2*(-0.43) + X3*(-0.07) + X4*(-0.76) + X5*0.96 + X6*(0.23) + X7*0.5 + X8*(-0.06)\n'
```

From the above equation, we can interpret the following-

- The coeff of X5 predictor is largest in magnitude thus it helps in discriminating the target the best
- The coeff of X4 predictor is smallest in magnitude thus it helps in discriminating the target the least.
- All the DS can be computed for each row using the above f(x) which will aid in classification

Comparison-

	Train Accuracy	Test Accuracy
LDA	0.834119	0.833333
Logistic Regression	0.835061	0.824561

Inference-

- Both models have similar accuracies (0.83 for the Logistic Regression, 0.84 for the LDA model) on the test data.
- LDA model shows slightly higher precision for both classes in the test data.
- LDA model exhibits better recall and F1 score for class 1.
- Hence, the LDA model appears slightly better overall for class 1 in the test data.

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

KNN Model-

Neighbors-based classification is a type of instance-based learning or non-generalizing learning. It does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point.

Generally, good KNN performance usually requires preprocessing of data to make all variables similarly scaled and centered

Now lets apply zscore on continuous columns and see the performance for KNN

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	-0.716161	-0.278185	-0.148020	0.565802	-1.419969	-1.437338	0.423832	-0.936736
1	-1.162118	0.856242	0.926367	0.565802	1.014951	-0.527684	0.423832	1.067536
2	-1.225827	0.856242	0.926367	1.417312	-0.608329	-1.134120	0.423832	1.067536
3	-1.926617	0.856242	-1.222408	-1.137217	-1.419969	-0.830902	-1.421084	-0.936736
4	-0.843577	-1.412613	-1.222408	-1.988727	-1.419969	-0.224465	0.423832	1.067536

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

<table>
<tr><td></td><td>Train dataset</td><td>Test dataset</td></tr>
<tr><td>Accuracy</td><td>0.8557964184731386</td><td>0.8245614035087719</td></tr>
<tr><td>Confusion Matrix</td><td>[[690 64]
[89 218]]</td><td>[[271 32]
[48 105]]</td></tr>
<tr><td>Classification report-</td><td>
<table>
<tr><td></td><td>precision</td><td>recall</td><td>f1-score</td><td>support</td></tr>
<tr><td>0</td><td>0.89</td><td>0.92</td><td>0.90</td><td>754</td></tr>
<tr><td>1</td><td>0.77</td><td>0.71</td><td>0.74</td><td>307</td></tr>
<tr><td>accuracy</td><td></td><td></td><td>0.86</td><td>1061</td></tr>
<tr><td>macro avg</td><td>0.83</td><td>0.81</td><td>0.82</td><td>1061</td></tr>
<tr><td>weighted avg</td><td>0.85</td><td>0.86</td><td>0.85</td><td>1061</td></tr>
</table>
</td><td>
<table>
<tr><td></td><td>precision</td><td>recall</td><td>f1-score</td><td>support</td></tr>
<tr><td>0</td><td>0.85</td><td>0.89</td><td>0.87</td><td>303</td></tr>
<tr><td>1</td><td>0.77</td><td>0.69</td><td>0.72</td><td>153</td></tr>
<tr><td>accuracy</td><td></td><td></td><td>0.82</td><td>456</td></tr>
<tr><td>macro avg</td><td>0.81</td><td>0.79</td><td>0.80</td><td>456</td></tr>
<tr><td>weighted avg</td><td>0.82</td><td>0.82</td><td>0.82</td><td>456</td></tr>
</table>
</td></tr>
</table>

Running the KNN with no of neighbours to be 1,3,5..19 and finding the optimal number of neighbours using the Misclassification error.

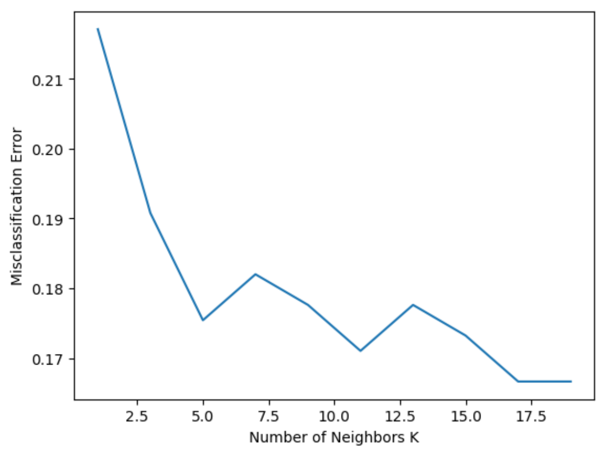
Misclassification error (MCE) = 1 - Test accuracy score.

Calculating MCE for each model with neighbours = 1,3,5...19 and finding the model with lowest MCE

MCE:

```
[0.2171052631578947,
 0.1907894736842105,
 0.17543859649122806,
 0.18201754385964908,
 0.17763157894736847,
 0.17105263157894735,
 0.17763157894736847,
 0.17324561403508776,
 0.16666666666666663,
 0.16666666666666663]
```

Plot misclassification error vs k (with k value on X-axis) using matplotlib



For **K = 17** it is giving the best test accuracy lets check train and test for K=17 with other evaluation metrics

```
▾ KNeighborsClassifier
KNeighborsClassifier(n_neighbors=17)
```

	Train dataset	Test dataset
Accuracy	0.8397737983034873	0.8333333333333334
Confusion Matrix	[[685 69] [101 206]]	[[279 24] [52 101]]
Classification Report	precision recall f1-score support 0 0.87 0.91 0.89 754 1 0.75 0.67 0.71 307 accuracy 0.84 1061 macro avg 0.81 0.79 0.80 1061 weighted avg 0.84 0.84 0.84 1061	precision recall f1-score support 0 0.84 0.92 0.88 303 1 0.81 0.66 0.73 153 accuracy 0.83 456 macro avg 0.83 0.79 0.80 456 weighted avg 0.83 0.83 0.83 456

Inference-

- While the model's overall accuracy is relatively good, there's a noticeable discrepancy between the performance metrics for the two classes. Hence, it is possible that the model might be slightly overfitting.

Lets check train and test for **K=11** with other evaluation metrics

```
▾ KNeighborsClassifier
KNeighborsClassifier(n_neighbors=11)
```

	Train dataset	Test dataset
Accuracy	0.8397737983034873	0.8289473684210527
Confusion Matrix	[[683 71] [99 208]]	[[273 30] [48 105]]
Classification Report	precision recall f1-score support 0 0.87 0.91 0.89 754 1 0.75 0.68 0.71 307 accuracy 0.84 1061 macro avg 0.81 0.79 0.80 1061 weighted avg 0.84 0.84 0.84 1061	precision recall f1-score support 0 0.85 0.90 0.88 303 1 0.78 0.69 0.73 153 accuracy 0.83 456 macro avg 0.81 0.79 0.80 456 weighted avg 0.83 0.83 0.83 456

As the difference between train and test accuracies is 1.08 % which is less than 10% (Industry standard), it is a valid model. So, we can consider 11 as the best value of K.

Inference-

The model's performance on the test seems to be consistent with its performance on the training set, indicating that it's not significantly overfitting or underfitting.

Gaussian Naive Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable.

GaussianNB implements the Gaussian Naive Bayes algorithm for classification

```
GaussianNB
GaussianNB()
```

Now GaussianNB classifier is built. The classifier is trained using training data. We can use fit() method for training it.

After building a classifier, our model is ready to make predictions. We can use predict() method with test set features as its parameters.

	Train Dataset	Test Dataset
Accuracy	0.8350612629594723	0.8223684210526315
Confusion Matrix	[[675 79] [96 211]]	[[263 40] [41 112]]
Classification report	precision recall f1-score support 0 0.88 0.90 0.89 754 1 0.73 0.69 0.71 307 accuracy 0.84 1061 macro avg 0.80 0.79 0.80 1061 weighted avg 0.83 0.84 0.83 1061	precision recall f1-score support 0 0.87 0.87 0.87 303 1 0.74 0.73 0.73 153 accuracy 0.82 456 macro avg 0.80 0.80 0.80 456 weighted avg 0.82 0.82 0.82 456

For both classes, precision and recall values are relatively good. But class 0 tends to have higher precision and recall compared to class 1 in both training and test data, which denotes a better predictability for class 0.

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

When the model is over-fitting i.e. the model's training accuracy is significantly higher than its test accuracy, it means that the model performs well on the training data and its ability to make accurate predictions on new, unseen data is not as strong. In that case, we will make use of Grid Search to get the best parameters and prune the tree.

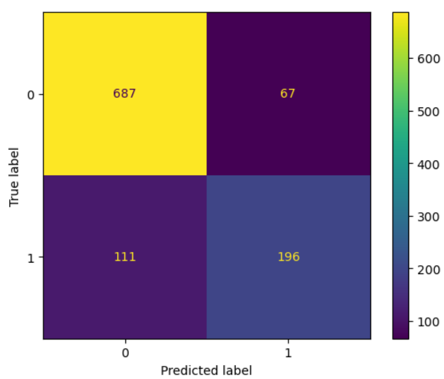
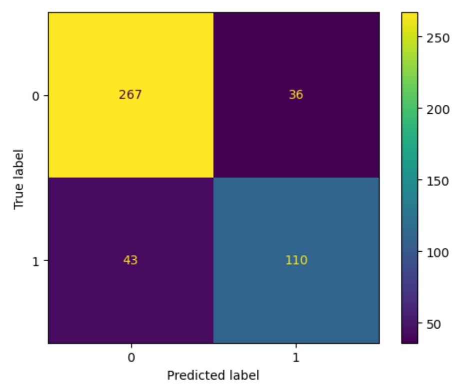
Model tuning is the process of optimizing the performance of a model by adjusting its hyperparameters.

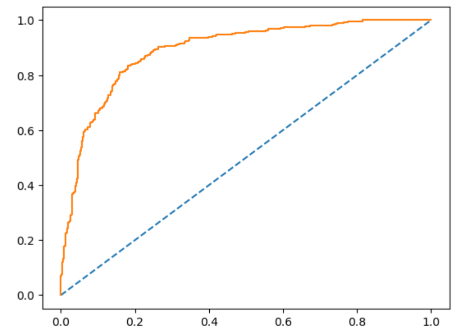
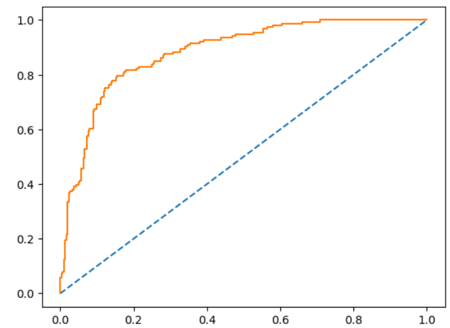
Applying GridSearchCV for Logistic Regression

```
GridSearchCV
estimator: LogisticRegression
LogisticRegression
```

Best params and best estimators-

```
{'penalty': 'none', 'solver': 'sag', 'tol': 1e-05}

LogisticRegression(max_iter=10000, n_jobs=2, penalty='none', solver='sag',
                   tol=1e-05)
```

<table>
<tr><td></td><td>Train dataset</td><td>Test dataset</td></tr>
<tr><td>Predicted class & Probabilities</td>
<td><table>
<tr><th></th><th>0</th><th>1</th></tr>
<tr><td>0</td><td>0.067424</td><td>0.932576</td></tr>
<tr><td>1</td><td>0.902829</td><td>0.097171</td></tr>
<tr><td>2</td><td>0.704545</td><td>0.295455</td></tr>
<tr><td>3</td><td>0.889109</td><td>0.110891</td></tr>
<tr><td>4</td><td>0.982795</td><td>0.017205</td></tr>
</table></td>
<td><table>
<tr><th></th><th>0</th><th>1</th></tr>
<tr><td>0</td><td>0.574859</td><td>0.425141</td></tr>
<tr><td>1</td><td>0.850442</td><td>0.149558</td></tr>
<tr><td>2</td><td>0.993050</td><td>0.006950</td></tr>
<tr><td>3</td><td>0.161871</td><td>0.838129</td></tr>
<tr><td>4</td><td>0.932308</td><td>0.067692</td></tr>
</table></td></tr>
<tr><td>Confusion Matrix</td>
<td>[[687 67]
 [111 196]]

</td>
<td>[[267 36]
 [43 110]]

</td></tr>
<tr><td>Classification report</td>
<td><table>
<tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr>
<tr><td>0</td><td>0.86</td><td>0.91</td><td>0.89</td><td>754</td></tr>
<tr><td>1</td><td>0.75</td><td>0.64</td><td>0.69</td><td>307</td></tr>
<tr><td>accuracy</td><td></td><td></td><td>0.83</td><td>1061</td></tr>
<tr><td>macro avg</td><td>0.80</td><td>0.77</td><td>0.79</td><td>1061</td></tr>
<tr><td>weighted avg</td><td>0.83</td><td>0.83</td><td>0.83</td><td>1061</td></tr>
</table></td>
<td><table>
<tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr>
<tr><td>0</td><td>0.86</td><td>0.88</td><td>0.87</td><td>303</td></tr>
<tr><td>1</td><td>0.75</td><td>0.72</td><td>0.74</td><td>153</td></tr>
<tr><td>accuracy</td><td></td><td></td><td>0.83</td><td>456</td></tr>
<tr><td>macro avg</td><td>0.81</td><td>0.80</td><td>0.80</td><td>456</td></tr>
<tr><td>weighted avg</td><td>0.83</td><td>0.83</td><td>0.83</td><td>456</td></tr>
</table></td></tr>
<tr><td>Accuracy</td><td>0.8322337417530632</td><td>0.8267543859649122</td></tr>
<tr><td>AUC</td><td>0.890</td><td>0.890</td></tr>
</table>

ROC		

- We do not observe much variation on the model post optimization.
- For class 0, the model's predictions of being correct are around 86% in both training and test sets, and for class 1 it's around 75%.
- The model captures around 91% of training data and 88% of test data accurately for the class 0 but around 64% of training data and 72% test data for the class 1.
- The model is better at predicting class 0 than class 1, as derived from the recall and F1-score values.
- Overall, the model appears valid and its performance on the training and test sets is relatively similar, which indicates good generalization.

Cross Validation on Naive Bayes Model-

We are performing 5-fold cross-validation on the training and testing dataset.

Train data set-

Performance scores of the model on each fold of the training data-

```
array([0.79342723, 0.84433962, 0.87735849, 0.80660377, 0.81603774])
```

Average performance estimate of the Naive Bayes model on the training data-

```
0.8275533705376915
```

Test data set-

Performance scores of the model on each fold of the test data-

```
array([0.7826087 , 0.84615385, 0.86813187, 0.85714286, 0.78021978])
```

Average performance estimate of the Naive Bayes model on the test data-

```
0.8268514094601052
```

Ensemble: Random Forest

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction.

In random forests, each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. Furthermore, when splitting each node during the construction of a tree, the best split is found either from all input features or a random subset of size max_features.

The purpose of these two sources of randomness is to decrease the variance of the forest estimator. Indeed, individual decision trees typically exhibit high variance and tend to overfit. The injected randomness in forests yield decision trees with somewhat decoupled prediction errors. By taking an average of those predictions, some errors can cancel out. Random forests achieve a reduced variance by combining diverse trees, sometimes at the cost of a slight increase in bias. In practice the variance reduction is often significant, hence yielding an overall better model.

The scikit-learn implementation combines classifiers by averaging their probabilistic prediction.

```
RandomForestClassifier
RandomForestClassifier(n_estimators=50, random_state=1)
```

	Train Dataset	Test Dataset
Accuracy	1.0	0.8135964912280702
Confusion Matrix	[[754 0] [0 307]]	[[274 29] [56 97]]

Classification Report	<pre> precision recall f1-score support
	
	           0 1.00 1.00 1.00 754
	           1 1.00 1.00 1.00 307
	
	    accuracy 1.00 1061
	   macro avg 1.00 1.00 1.00 1061
	weighted avg 1.00 1.00 1.00 1061</pre>
		<pre> precision recall f1-score support
		
		           0 0.83 0.90 0.87 303
		           1 0.77 0.63 0.70 153
		
		    accuracy 0.81 456
		   macro avg 0.80 0.77 0.78 456
		weighted avg 0.81 0.81 0.81 456</pre>

It is an overfitted model since it has 100% accuracy on the training data while it is unable to perform the same on unseen data.

Cross-validation-

Performing 10-fold cross-validation on the training and test dataset

```
Accuracy of trained dataset: 0.8190354434843943
Accuracy of test dataset: 0.7892753623188407

 Classification report for train dataset:
              precision    recall  f1-score   support

           0       0.86      0.91      0.89       754
           1       0.75      0.64      0.69       307

    accuracy                           0.83      1061
   macro avg       0.80      0.77      0.79      1061
weighted avg       0.83      0.83      0.83      1061


Classification report for test dataset:
              precision    recall  f1-score   support

           0       0.86      0.88      0.87       303
           1       0.75      0.72      0.74       153

    accuracy                           0.83       456
   macro avg       0.81      0.80      0.80       456
weighted avg       0.83      0.83      0.83       456
```

The model is a valid good fit with consistent performance metrics between the training and test data sets.

Ensemble: Boosting

The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction.

The number of weak learners is controlled by the parameter n_estimators. The learning_rate parameter controls the contribution of the weak learners in the final combination. By default, weak learners are decision stumps. Different weak learners can be specified through the base_estimator parameter. The main parameters to tune to obtain good results are n_estimators and the complexity of the base estimators (e.g., its depth max_depth or minimum required number of samples to consider a split min_samples_split).

Ada Boosting-

```
AdaBoostClassifier
AdaBoostClassifier(random_state=1)
```

	Train Dataset	Test Dataset
Accuracy	0.8463713477851084	0.8135964912280702
Confusion Matrix	[[688 66] [97 210]]	[[266 37] [48 105]]
Classification Report	precision recall f1-score support 0 0.88 0.91 0.89 754 1 0.76 0.68 0.72 307 accuracy 0.85 1061 macro avg 0.82 0.80 0.81 1061 weighted avg 0.84 0.85 0.84 1061	precision recall f1-score support 0 0.85 0.88 0.86 303 1 0.74 0.69 0.71 153 accuracy 0.81 456 macro avg 0.79 0.78 0.79 456 weighted avg 0.81 0.81 0.81 456

The model is a reasonably valid one as the performance metrics are relatively consistent between the training and test sets and it seems to generalize reasonably well on unseen data.

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.

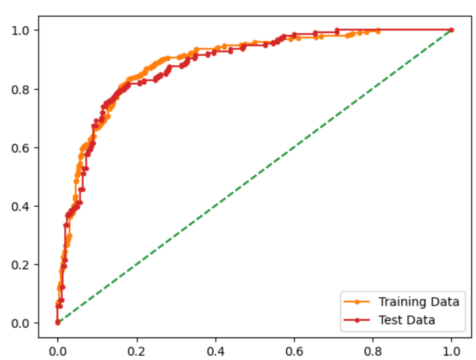
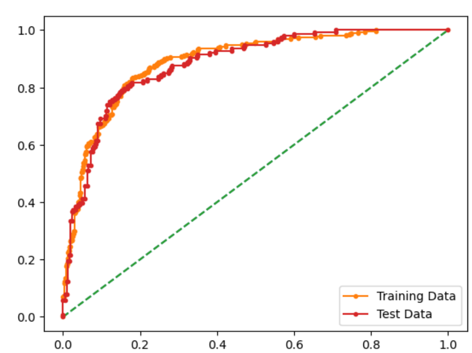
Comparison of Different Models-

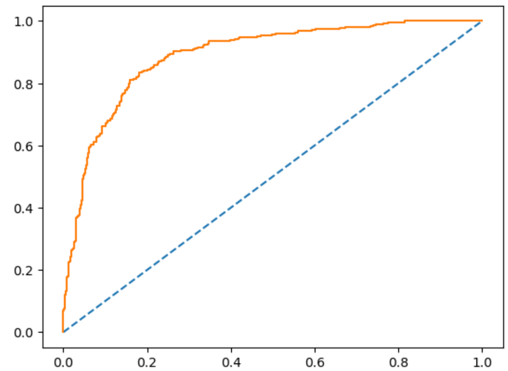
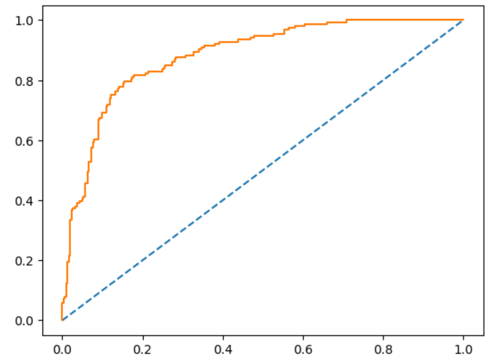
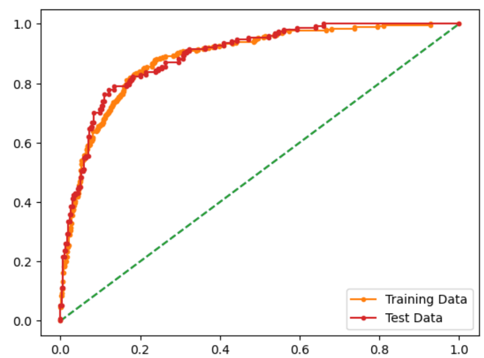
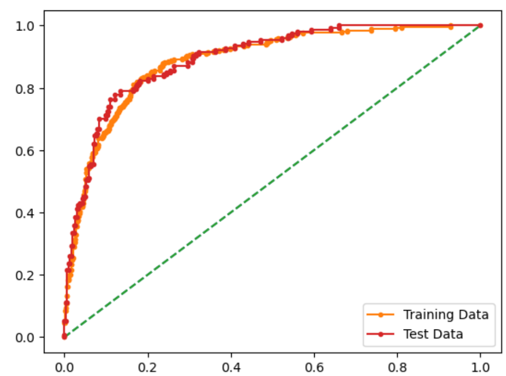
Interest Class is the political party that a voter is likely to vote (0 for Labour and 1 for Conservative).

Let's look at the performance of all the models on the Train Data set

	Train dataset	Test dataset
Logistic Regression	precision recall f1-score support 0 0.86 0.91 0.88 754 1 0.74 0.64 0.69 307 accuracy 0.83 1061 macro avg 0.80 0.77 0.79 1061 weighted avg 0.83 0.83 0.83 1061	precision recall f1-score support 0 0.87 0.88 0.88 303 1 0.76 0.74 0.75 153 accuracy 0.84 456 macro avg 0.82 0.81 0.81 456 weighted avg 0.83 0.84 0.83 456
Grid Search CV Logistic Regression	precision recall f1-score support 0 0.86 0.91 0.89 754 1 0.75 0.64 0.69 307 accuracy 0.83 1061 macro avg 0.80 0.77 0.79 1061 weighted avg 0.83 0.83 0.83 1061	precision recall f1-score support 0 0.86 0.88 0.87 303 1 0.75 0.72 0.74 153 accuracy 0.83 456 macro avg 0.81 0.80 0.80 456 weighted avg 0.83 0.83 0.83 456
LDA	precision recall f1-score support 0 0.87 0.90 0.88 754 1 0.74 0.65 0.69 307 accuracy 0.83 1061 macro avg 0.80 0.78 0.79 1061 weighted avg 0.83 0.83 0.83 1061	precision recall f1-score support 0 0.87 0.89 0.88 303 1 0.77 0.74 0.76 153 accuracy 0.84 456 macro avg 0.82 0.81 0.82 456 weighted avg 0.84 0.84 0.84 456
Naïve Bayes	precision recall f1-score support 0 0.88 0.90 0.89 754 1 0.73 0.69 0.71 307 accuracy 0.84 1061 macro avg 0.80 0.79 0.80 1061 weighted avg 0.83 0.84 0.83 1061	precision recall f1-score support 0 0.87 0.87 0.87 303 1 0.74 0.73 0.73 153 accuracy 0.82 456 macro avg 0.80 0.80 0.80 456 weighted avg 0.82 0.82 0.82 456
KNN Model (k=17)	precision recall f1-score support 0 0.87 0.91 0.89 754 1 0.75 0.67 0.71 307 accuracy 0.84 1061 macro avg 0.81 0.79 0.80 1061 weighted avg 0.84 0.84 0.84 1061	precision recall f1-score support 0 0.84 0.92 0.88 303 1 0.81 0.66 0.73 153 accuracy 0.83 456 macro avg 0.83 0.79 0.80 456 weighted avg 0.83 0.83 0.83 456
KNN Model (k=11)	precision recall f1-score support 0 0.87 0.91 0.89 754 1 0.75 0.68 0.71 307 accuracy 0.84 1061 macro avg 0.81 0.79 0.80 1061 weighted avg 0.84 0.84 0.84 1061	precision recall f1-score support 0 0.85 0.90 0.88 303 1 0.78 0.69 0.73 153 accuracy 0.83 456 macro avg 0.81 0.79 0.80 456 weighted avg 0.83 0.83 0.83 456

<table>
<tr><td>Random Forest</td><td>precision recall f1-score support
0 1.00 1.00 1.00 754
1 1.00 1.00 1.00 307
accuracy 1.00 1061
macro avg 1.00 1.00 1.00 1061
weighted avg 1.00 1.00 1.00 1061</td><td>precision recall f1-score support
0 0.83 0.90 0.87 303
1 0.77 0.63 0.70 153
accuracy 0.81 456
macro avg 0.80 0.77 0.78 456
weighted avg 0.81 0.81 0.81 456</td></tr>
<tr><td>CV Random Forest</td><td>Classification report for train dataset:
precision recall f1-score support
0 0.86 0.91 0.89 754
1 0.75 0.64 0.69 307
accuracy 0.83 1061
macro avg 0.80 0.77 0.79 1061
weighted avg 0.83 0.83 0.83 1061</td><td>Classification report for test dataset:
precision recall f1-score support
0 0.86 0.88 0.87 303
1 0.75 0.72 0.74 153
accuracy 0.83 456
macro avg 0.81 0.80 0.80 456
weighted avg 0.83 0.83 0.83 456</td></tr>
<tr><td>Ada Boost</td><td>precision recall f1-score support
0 0.88 0.91 0.89 754
1 0.76 0.68 0.72 307
accuracy 0.85 1061
macro avg 0.82 0.80 0.81 1061
weighted avg 0.84 0.85 0.84 1061</td><td>precision recall f1-score support
0 0.85 0.88 0.86 303
1 0.74 0.69 0.71 153
accuracy 0.81 456
macro avg 0.79 0.78 0.79 456
weighted avg 0.81 0.81 0.81 456</td></tr>
</table>

<table>
<tr><th></th><th>Train dataset</th><th>Test dataset</th></tr>
<tr><td>Logistic Regression</td><td>AUC: 0.890
ROC:
</td><td>AUC: 0.883
ROC:
</td></tr>
<tr><td>Grid Search CV</td><td>AUC: 0.890
ROC:</td><td>AUC: 0.890
ROC:</td></tr>
</table>

Logistic Regression		
LDA	AUC: 0.887 ROC: 	AUC: 0.893 ROC: 

Inferences:

- Logistic Regression (LR) and Linear Discriminant Analysis (LDA) seem to perform consistently well across various metrics on both training and test data sets with a balance between precision and recall values for both classes.
- Naive Bayes (NB) and K-Nearest Neighbors (KNN) also demonstrate good performance with balanced precision and recall but slightly lower accuracy compared to LR and LDA.
- Random Forest seems to overfit the training data due to its 100% training accuracy.
- Boosting provides a slightly lower accuracy compared to LR and LDA, with balanced precision and recall for both classes.
- Hence, Logistic Regression (LR) and Linear Discriminant Analysis (LDA) are optimal models as they maintain a good balance between the performance metrics and generalization to unseen data making them more reliable for predicting the party votes.

1.8 Based on these predictions, what are the insights?

- The dataset represents information on various socio-economic conditions including political parties, attitudes toward European integration, and knowledge about party positions on European integration.
- Interest Class is the political party that a voter is likely to vote (0 for Labour and 1 for Conservative).
- There's a moderate level of knowledge about parties positions.
- 'Labour' appears to be the more dominant voting choice.
- Also there is an imbalance in the gender, with more female participants compared to male participants.
- Majority of people casting votes are aged between 35-55.
- Participants who have voted for the Labour party has rated the current national economic condition and household economic condition to be good. While, those who have voted for the Conservative party has rated the current national economic condition and household economic condition to be moderate.
- Participants who have voted for the Labour party has rated the Labour leader with a major score of 4. And Conservative leader with a score of 2 and vice versa.
- Participants have a moderate level of knowledge on the parties' positions on European integration which needs to be improved.
- People who've voted for the Conservative party have high 'Eurosceptic' sentiment. While those voted for Labour party have mixed attitudes toward European integration
- There is a strong relationship between how people view current national economic conditions and current household economic conditions which influences their opinions on the leader.
- Also, there is a strong relation between perceptions of the Conservative leader and attitude toward European integration. This suggests that individuals who hold Eurosceptic sentiments might also have opinions aligned with that of the Conservative leader.

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

2.1 Find the number of characters, words, and sentences for the mentioned documents.

```
Number of Characters in President Franklin D. Roosevelt speech: 7571
Number of Characters in President John F. Kennedy speech: 7618
Number of Characters in President Richard Nixon speech: 9991
```

```
Number of Words in President Franklin D. Roosevelt speech: 1536
Number of Words in President John F. Kennedy speech: 1546
Number of Words in President Richard Nixon speech: 2028
```

```
Number of Sentences in President Franklin D. Roosevelt speech: 68
Number of Sentences in President John F. Kennedy speech: 52
Number of Sentences in President Richard Nixon speech: 69
```

2.2 Remove all the stopwords from all three speeches.

We have created a Data Frame containing a column 'Speech' with speeches by Roosevelt, Kennedy, and Nixon.

	Speech
0	On each national day of inauguration since 178...
1	Vice President Johnson, Mr. Speaker, Mr. Chief...
2	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...

Performing basic pre-processing-

Converting the words in the speeches to lower case -

```
0    on each national day of inauguration since 178...
1    vice president johnson, mr. speaker, mr. chief...
2    mr. vice president, mr. speaker, mr. chief jus...
Name: Speech, dtype: object
```

Removing punctuation from the text i.e. replacing any non-alphanumeric characters (except spaces) with an empty string in the 'Speech' column.

```
0    on each national day of inauguration since 178...
1    vice president johnson mr. speaker mr. chief j...
2    mr. vice president mr. speaker mr. chief justi...
Name: Speech, dtype: object
```

Stemming: Returning words to their original stem i.e. removal of suffices like “ing”, “ly”, “s”, etc.

```
0    on each nation day of inaugur sinc 1789 the pe...
1    vice presid johnson mr. speaker mr. chief just...
2    mr. vice presid mr. speaker mr. chief justic s...
Name: Speech, dtype: object
```

Stop words: Common words that are not useful in providing value or context. Eg: ‘the’, ‘an’, ‘in’ etc.

Printing some of the stopwords-

```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've"]
```

Printing length of stopwords in each speech –

```
0    667
1    628
2    911
Name: Speech, dtype: int64
```

Post removal of stopwords from the speech-

```
0    nation day inaugur sinc 1789 peopl renew sens ...
1    vice presid johnson mr. speaker mr. chief just...
2    mr. vice presid mr. speaker mr. chief justic s...
Name: Speech, dtype: object
```

Length of stopwords post removal-

```
0    0
1    0
2    0
Name: Speech, dtype: int64
```

Common Words Removal

We have created a list of 20 frequently occurring words and then removed few from our speech.

```
words
--          63
us          45
let         39
thi         36
nation      32
new         26
ha          26
america     20
peac        18
becaus      17
year        16
govern      16
respons     15
peopl       15
know        15
world       15
shall       13
freedom     12
human       12
everi       12
Name: count, dtype: int64
```

After removing some common words like: '--', 'us', 'let', 'thi', 'ha', 'becaus' which has high freq,

```
0    nation day inaugur sinc 1789 peopl renew sens ...
1    vice presid johnson mr. speaker mr. chief just...
2    mr. vice presid mr. speaker mr. chief justic s...
Name: Speech, dtype: object
```

After performing cleaning –

```
Total no. of words in Roosevelt speech: 633

Total no. of words in Kennedy speech: 695

Total no. of words in Nixon speech: 812
```

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)

```
Top 3 words with high frequency in Roosevelt speech: ['nation', 'know', 'spirit']

Top 3 words with high frequency in Kennedy speech: ['let', 'us', 'world']

Top 3 words with high frequency in Nixon speech: ['us', 'let', 'america']
```

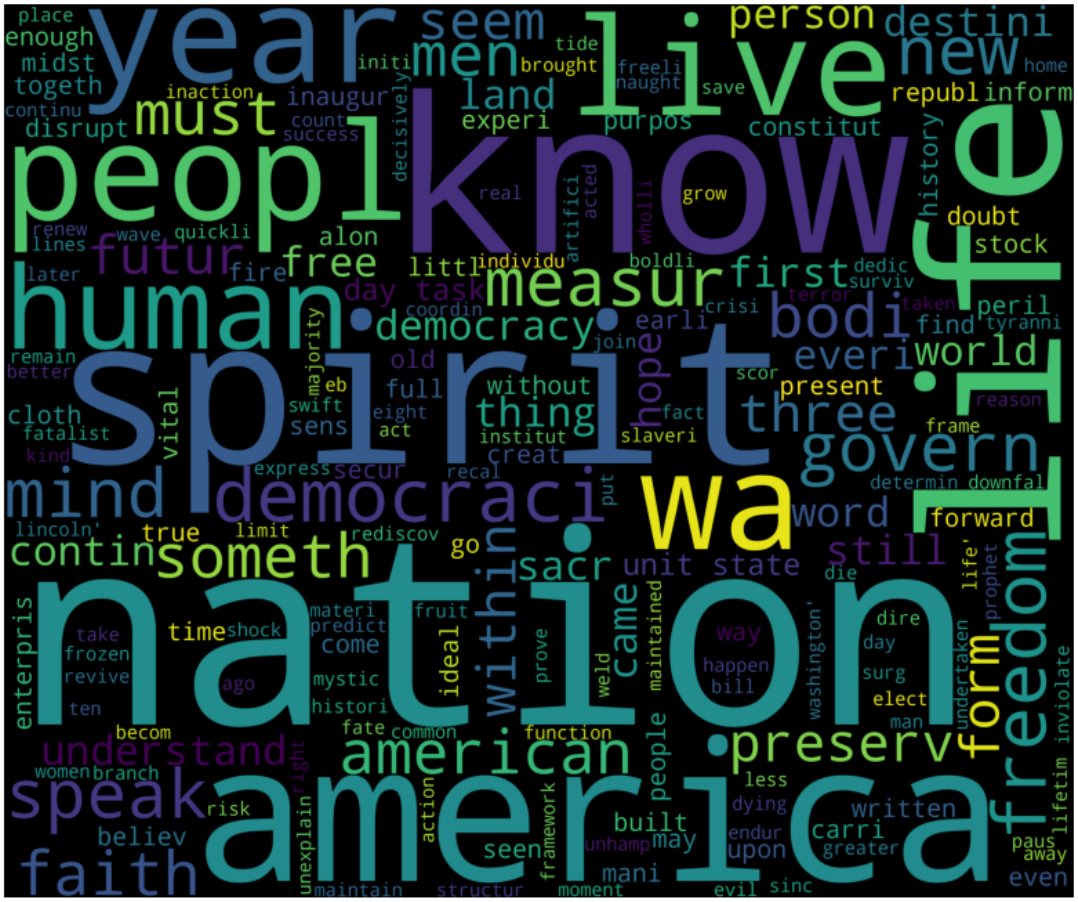
Post removal of common words,

```
Top 3 words with high frequency in Roosevelt speech: ['nation', 'know', 'peopl']

Top 3 words with high frequency in Kennedy speech: ['power', 'world', 'nation']

Top 3 words with high frequency in Nixon speech: ['america', 'peac', 'world']
```

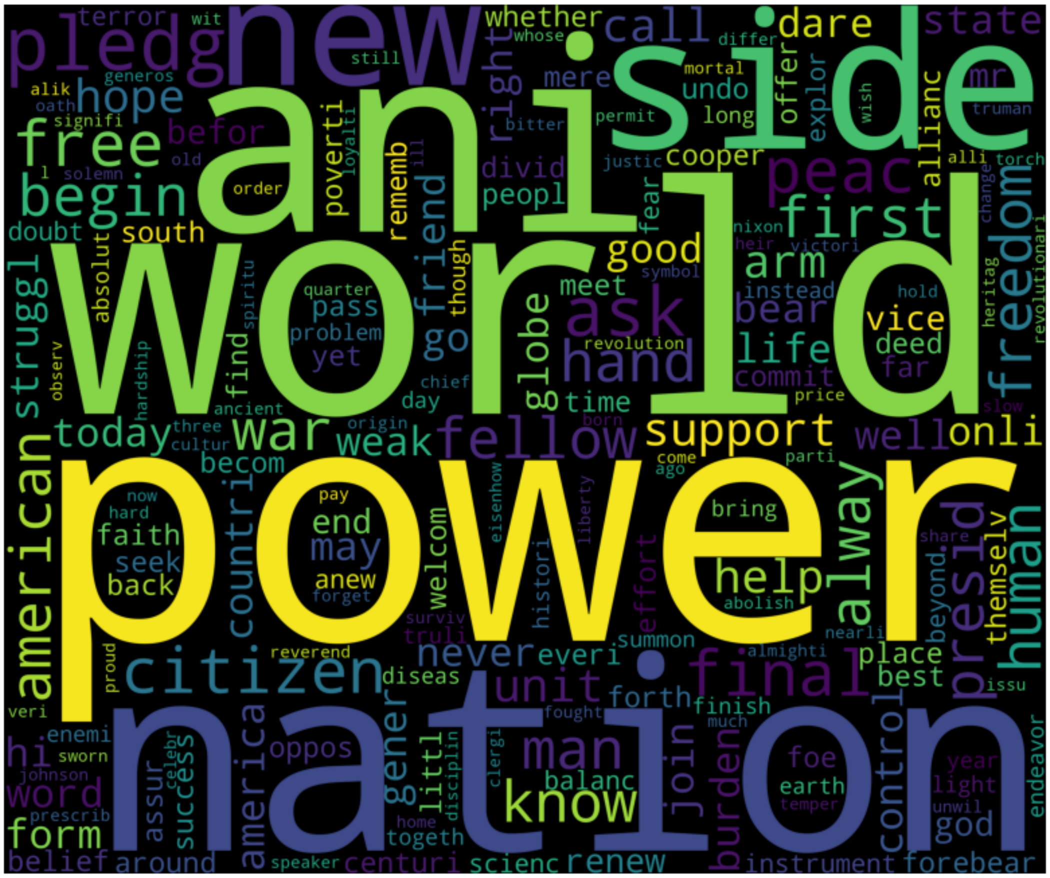
2.4 Plot the word cloud of each of the speeches of the variable. (after removing the stopwords)

Word cloud is a visual representation of texts where the size of each word is represented based on its frequency. We are plotting the word cloud post removal of stopwords and common words.

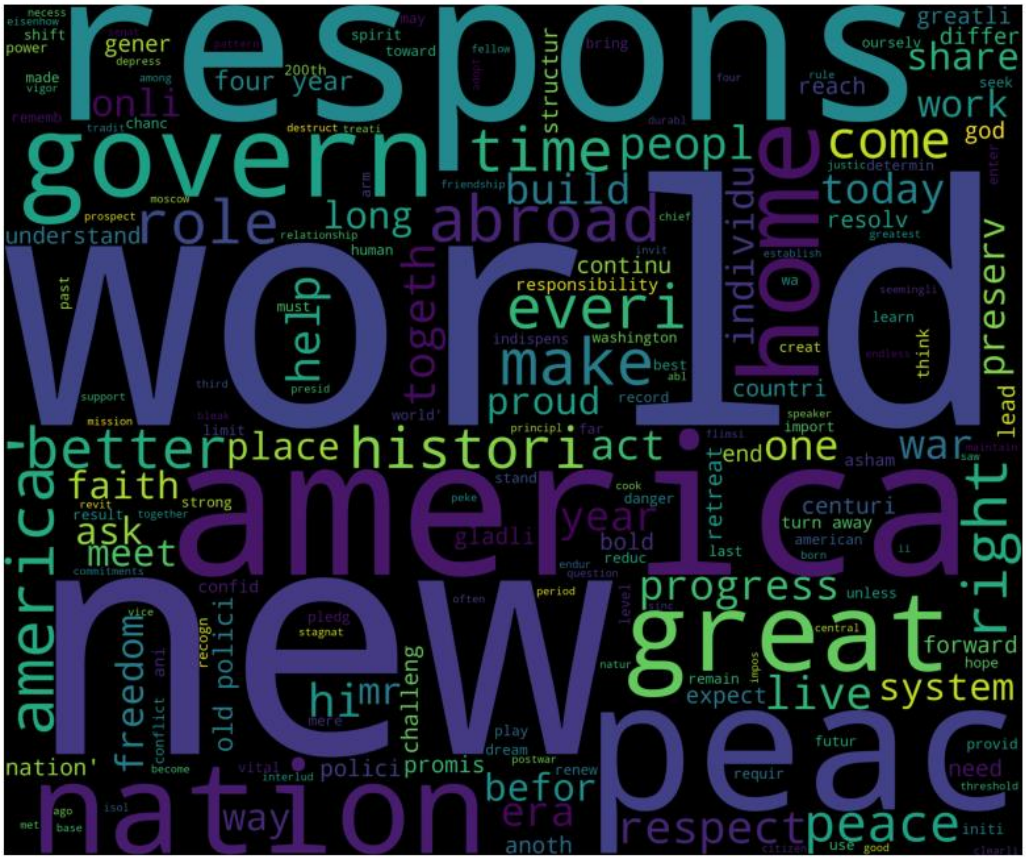
Roosevelt speech-



Kennedy Speech-



Nixon speech-



3. Dataset:

Problem 1: [Election_Data.xlsx]

Problem 2: [Speeches]

4. Data Dictionary for Problem 1:

1. vote: Party choice: Conservative or Labour
2. age: in years
3. economic.cond.national: Assessment of current national economic conditions, 1 to 5.

4. economic.cond.household: Assessment of current household economic conditions, 1 to 5.
5. Blair: Assessment of the Labour leader, 1 to 5.

6. Hague: Assessment of the Conservative leader, 1 to 5.

7. Europe: an 11-point scale that measures respondents' attitudes toward European integration. High scores represent ‘Eurosceptic’ sentiment.

8. political.knowledge: Knowledge of parties' positions on European integration, 0 to 3.

9. gender: female or male.

THE END.