Business Report: Machine Learning

Great Learning

POWER AHEAD

Dhruv Dosad

TABLE OF CONTENTS

List of Figures

List of Tables

Problem 1

Problem Statement:

You are hired by one of the leading news channels CNBE who wants to analyse 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.

Data Dictionary:

1. **vote**: Party choice - Conservative or Labour [Target Variable].
2. **age**: Age in years.
3. **economic.cond.national**: Assessment of current national economic conditions, 1 to 5 (1 being the lowest rating, 5 being the highest rating).
4. **economic.cond.household**: Assessment of current household economic conditions, 1 to 5.
5. **Blair**: Assessment of the Labour leader Tony Blair, 1 to 5.
6. **Hague**: Assessment of the Conservative leader William Hague, 1 to 5.
7. **Europe**: An 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Eurosceptic' sentiment (A person who is opposed to increasing the powers of the European Union - against European integration of the UK).
8. **political.knowledge**: Knowledge of parties' positions on European integration, 0 to 3.
9. **gender**: Female or male.

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.

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

Table 1.1 Sample of the Dataset

Dropping Index Column: The unneeded Serial no. column Unnamed: 0 is removed from the dataset.

Datatypes of features:

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

Table 1.2 Datatype of the features

Information about the Dataset:

```
<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
```

Table 1.3 Info of the Dataset

The dataset has 9 Features: 7 numerical type & 2 object type, 1525 records

Missing values check: There are no missing values in the dataset.

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

Table 1.4 Missing Values Check

Duplicate records check: There are 8 duplicate records in the dataset which have been dropped

Summary of the Dataset:

	count	mean	std	min	25%	50%	75%	max	skew
age	1517.0	54.24	15.70	24.0	41.0	53.0	67.0	93.0	0.14
economic.cond.national	1517.0	3.25	0.88	1.0	3.0	3.0	4.0	5.0	-0.24
economic.cond.household	1517.0	3.14	0.93	1.0	3.0	3.0	4.0	5.0	-0.14
Blair	1517.0	3.34	1.17	1.0	2.0	4.0	4.0	5.0	-0.54
Hague	1517.0	2.75	1.23	1.0	2.0	2.0	4.0	5.0	0.15
Europe	1517.0	6.74	3.30	1.0	4.0	6.0	10.0	11.0	-0.14
political.knowledge	1517.0	1.54	1.08	0.0	0.0	2.0	2.0	3.0	-0.42

Table 1.5 Summary of the Numerical Variables of the Dataset

Summary:

* **Data quality**: **No** instances of "bad" or corrupt data were found.

* **Economic condition ratings**: Both national and household levels **lean positively towards Tony Blair** (ratings >3), indicating public approval of his economic stewardship.

* **Leadership perception**: William Hague's rating **falls below** the neutral score, **signifying less public confidence** in his leadership.

* **Skewness**: All variables show **skewness values within |0.5|,** suggesting **minimal data asymmetry**, implying data can be treated as **essentially unskewed**

	count	unique	top	freq
vote	1517	2	Labour	1057
gender	1517	2	female	808

Table 1.6 Summary of the Categorical Variables of the Dataset

Summary

* **Categorical variables**: Two levels present - **Labour and gender**.
* **Labour frequency**: **Higher** representation with **1057** occurrences out of **1517** in the target variable.
* **Gender frequency**: **Females** are **more prevalent** with **808 out of 1517** occurrences

1.2 Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also check for outliers.

Interpret the inferences for each Distribution plots(histogram) or similar plots for the continuous columns. Box plots. Appropriate plots for categorical variables. Inferences on each plot. Outliers 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.

For univariate & bivariate analysis we treat the rating variables from survey (economic.cond.national, economic.cond.household, Blair, Hague, Europe, political.knowledge) as ordinal categorical variables or discrete numerical variables as when necessary.

Univariate Analysis:

We first take a look at the numerical variables.

1. **age:** Age in years.

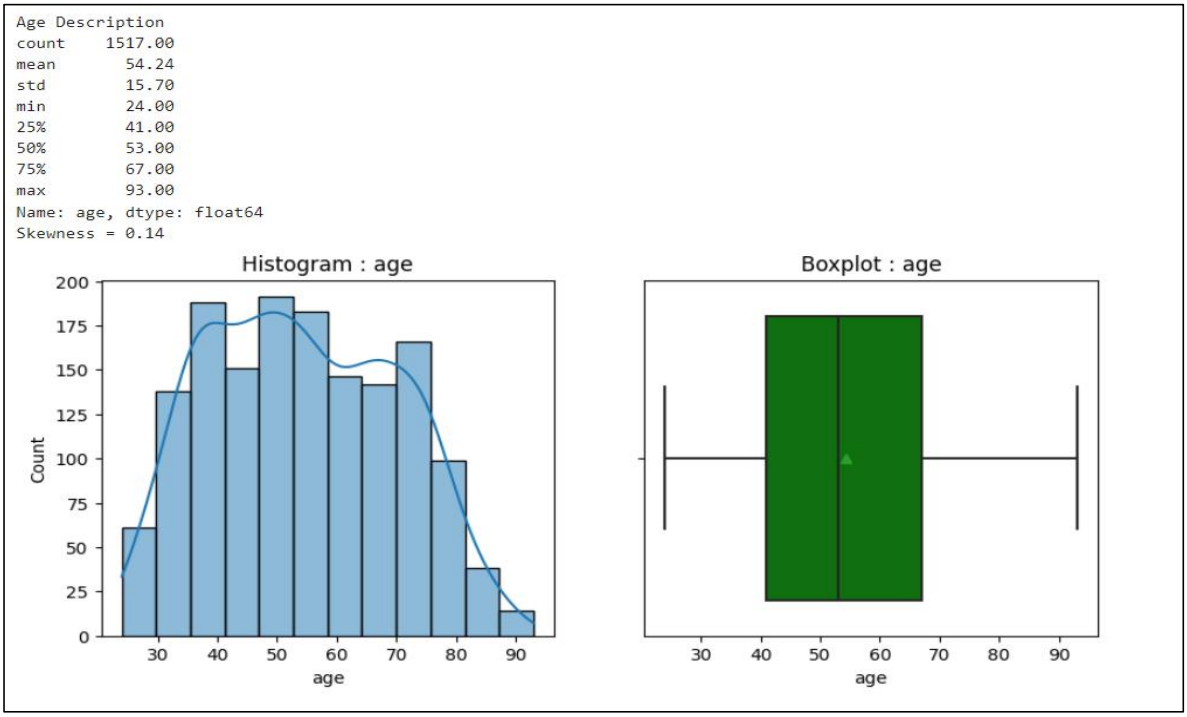

Fig 1.1 Univariate Analysis – age

Age, ranging from 24 to 93, has a slight right skew but can be considered normally distributed for analysis due to minimal skewness and a wavy peak.

Its mean (54.24) slightly exceeds the median (53), indicating this minor skewness. No outliers are present

2. **economic.cond.national:** Assessment of current national economic conditions, 1 to 5 (1 being the lowest rating, 5 being the highest rating).

```
economic.cond.national %age Count
3     39.82
4     35.46
2     16.88
5      5.41
1      2.44
Name: economic.cond.national, dtype: float64
Average Score = 3.25
```

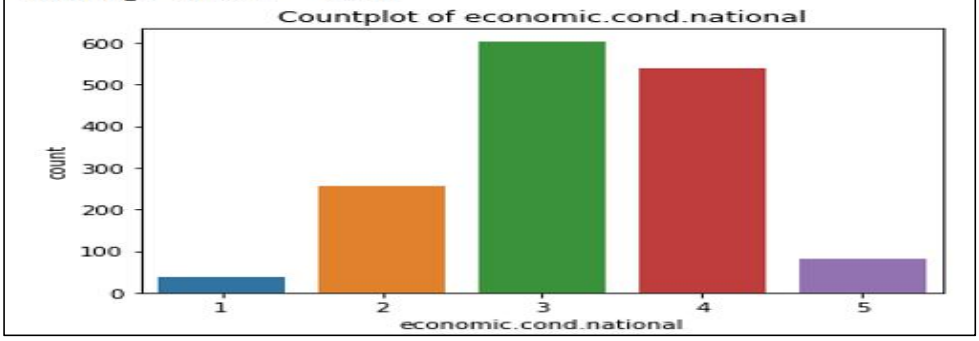

Fig 1.2 Univariate Analysis – economic.cond.national

3. **economic.cond.household:** Assessment of current household economic conditions, 1 to 5 (1 being the lowest rating, 5 being the highest rating).

```
economic.cond.household %age Count
3    42.52
4    28.68
2    18.46
5     6.06
1     4.28
Name: economic.cond.household, dtype: float64
Average Score = 3.14
```

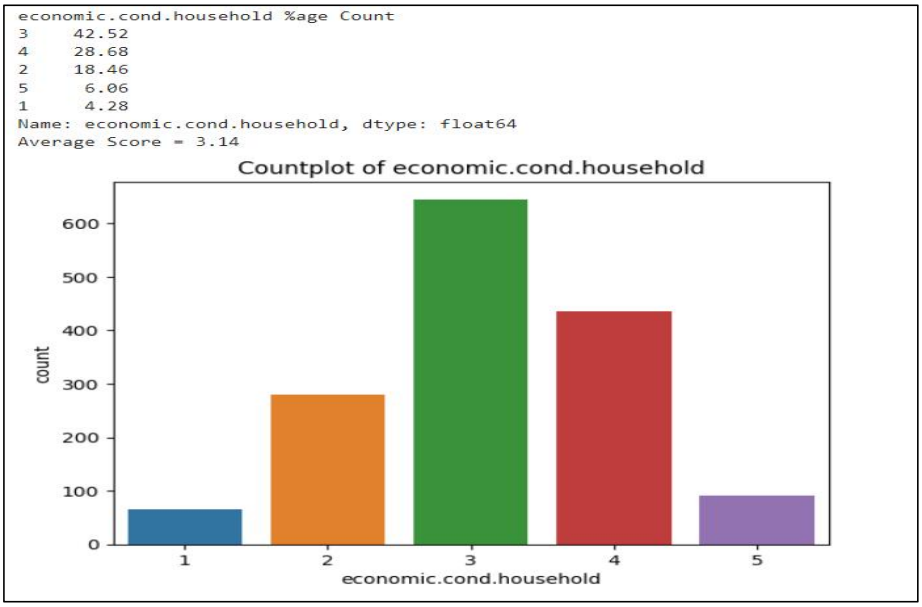

Fig 1.3 Univariate Analysis – economic.cond.household

4. **Blair:** Assessment of the Labour leader Tony Blair, 1 to 5.

```
Blair %age Count
4     54.91
2     28.61
5     10.02
1      6.39
3      0.07
Name: Blair, dtype: float64
Average Score = 3.34
```

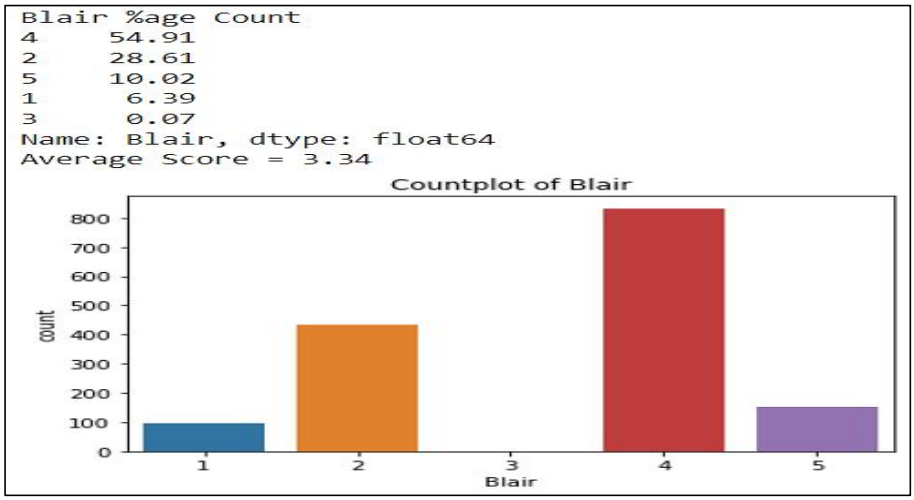

Fig 1.4 Univariate Analysis – Blair

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

```
Hague %age Count
2     40.67
4     36.72
1     15.36
5      4.81
3      2.44
Name: Hague, dtype: float64
Average Score = 2.75
```

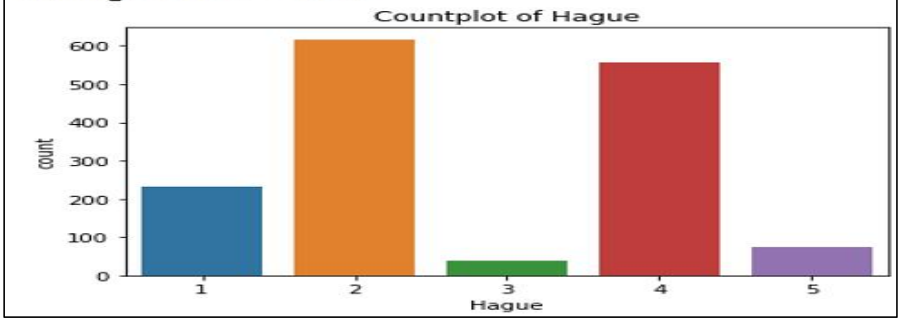

Fig 1.5 Univariate Analysis – Hague

6. **Europe:** An 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Eurosceptic' sentiment

```
Europe %age Count
11      22.28
6       13.65
3        8.44
4        8.31
5        8.11
9        7.32
8        7.32
1        7.19
10       6.66
7        5.67
2        5.08
Name: Europe, dtype: float64
Average Score = 6.74
```

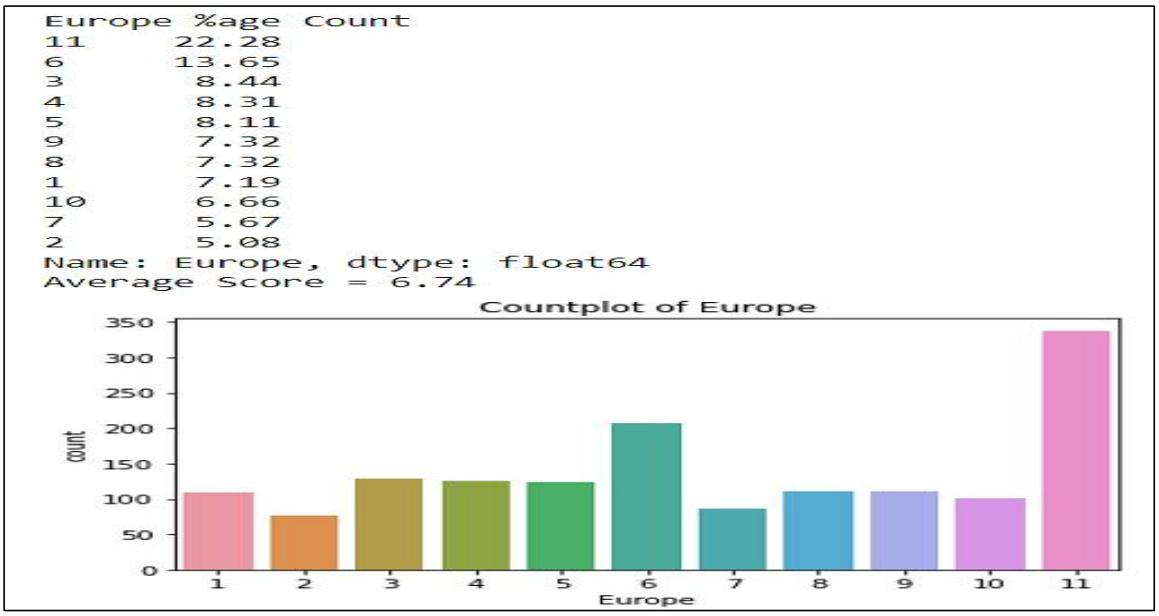

Fig 1.6 Univariate Analysis – Europe

7. **Political.knowledge:** Knowledge of parties' positions on European integration, 0 to 3.

```
political.knowledge %age Count
2     51.15
0     29.93
3     16.41
1      2.50
Name: political.knowledge, dtype: float64
Average Score = 1.54
```

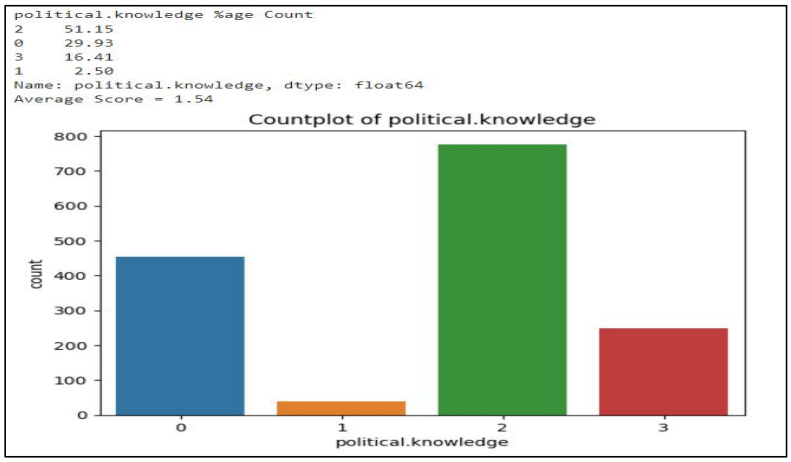

Fig 1.7 Univariate Analysis – political.knowledge

8. **vote:** Party choice - Conservative or Labour [Target Variable].

```
vote %age Count
Labour          69.68
Conservative    30.32
Name: vote, dtype: float64
```

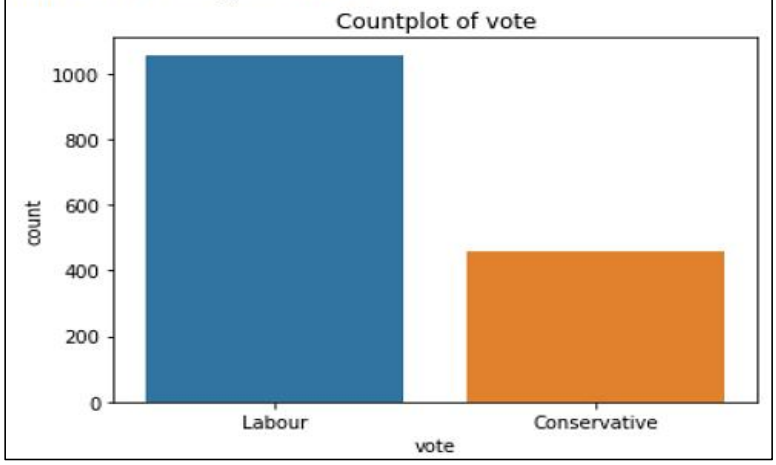

Fig 1.8 Univariate Analysis – vote

9. **Gender:** Male/Female.

```
gender %age Count
female    53.26
male      46.74
Name: gender, dtype: float64
```

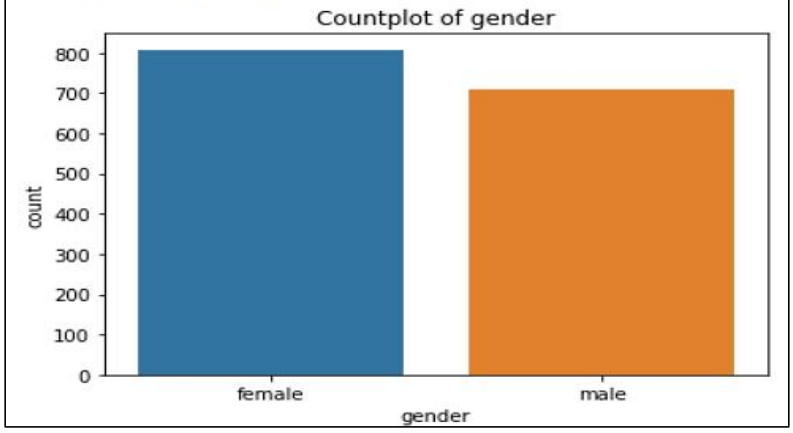

Fig 1.9 Univariate Analysis – gender

Inferences from above

* **National economic condition** received a rating of **3 or 4 from ~75%** respondents, **average score: 3.25.**
* **Household economic condition** rated **3 or 4** by **~70% of individuals**, **average score: 3.14**.
* **Blair's** favourable rating by **~65% respondents**, **average score: 3.34**, suggests satisfaction with **Labour party.**
* **Hague rated poorly** by **~55%** respondents, **average score: 2.75**.
* **Average score of 6.74** indicates **majority** leaning towards **Brexit**, with **22% strongly favouring**.
* **30% respondents** unaware of their party's stance on **European Integration**, while **50% are well-informed**.
* Survey of 1500 individuals shows **~70% support for Labour Party**, ~30% for Conservative Party.
* Slight class imbalance noted in target variable, but no drastic underrepresentation of Conservative class. Over/under sampling techniques not required

Bivariate Analysis

To analyze different variable types, we use specific techniques:

Numerical Variables:

- Pair plot: Visualize the relationship between two numerical variables.
- Correlation matrix heatmap: Assess the correlation between numerical variables.

Categorical Variables:

- Cross tables: Examine the relationship between two categorical variables.
- Bar plots: Visualize the distribution and frequencies of categories in each variable.

Numerical vs. Categorical Variables:

- Boxplots, violin plots, or bar plots: Compare the distribution of a numerical variable across categories of a categorical variable.

Numerical VS Numerical Variable:

Pair Plot:

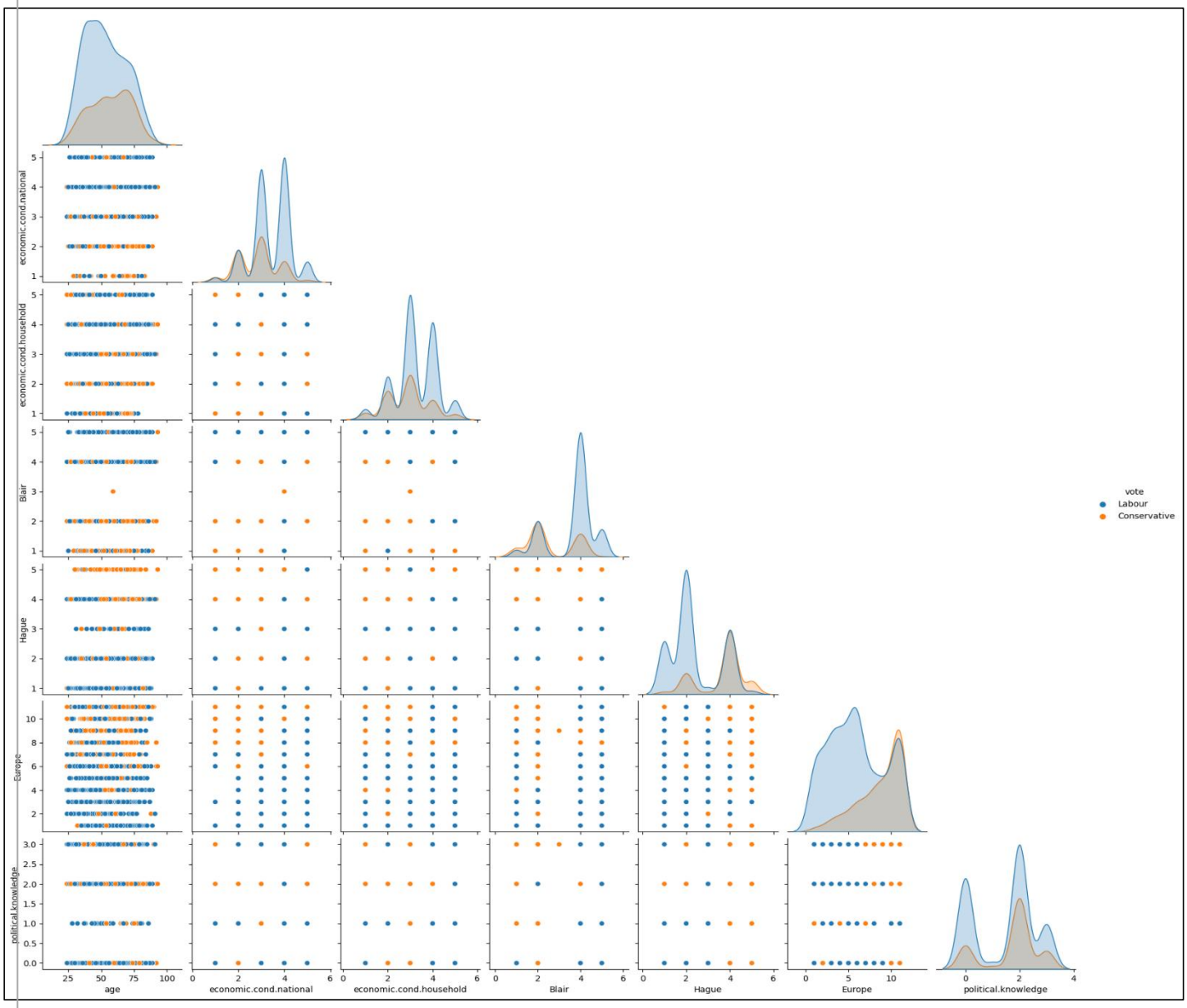

Fig 1.10 Pair plot

Correlation Matrix Heatmap:

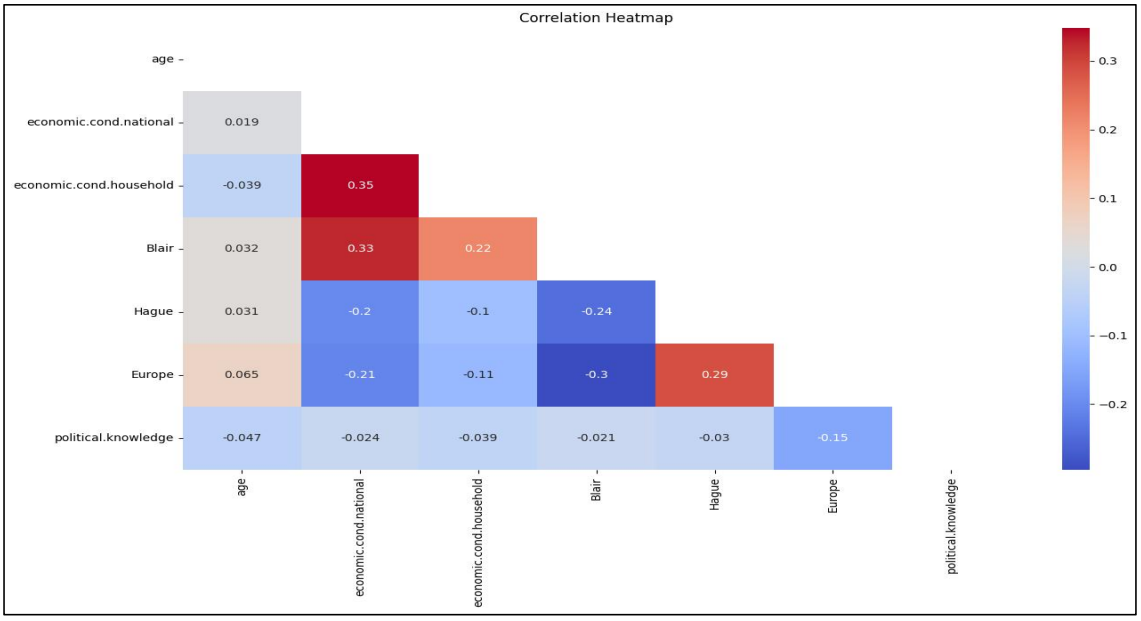

Fig 1.11 Correlation Matrix Heatmap

Summary:

* **Mild positive correlation** exists between **national and household economic condition ratings**, and also with Labour Party leader Tony Blair's ratings. Conversely, **slight negative correlation** with **Conservative Party leader William Hague's ratings**, suggesting **general satisfaction** with current economy and preference for Labour.

* **Mild negative correlation** between **Brexit sentiments and Blair's ratings**, mild positive correlation with Hague's ratings, suggesting **Brexit supporters are discontent with Labour's EU stance** and **prefer Conservatives.**

* **Blair and Hague's** ratings exhibit **weak negative correlation**, as expected from opposing election candidates.

Categorical VS Categorical Variable:

gender	female	male
vote		
Conservative	31.81	28.63
Labour	68.19	71.37

Table 1.7 Cross table of Vote & Gender

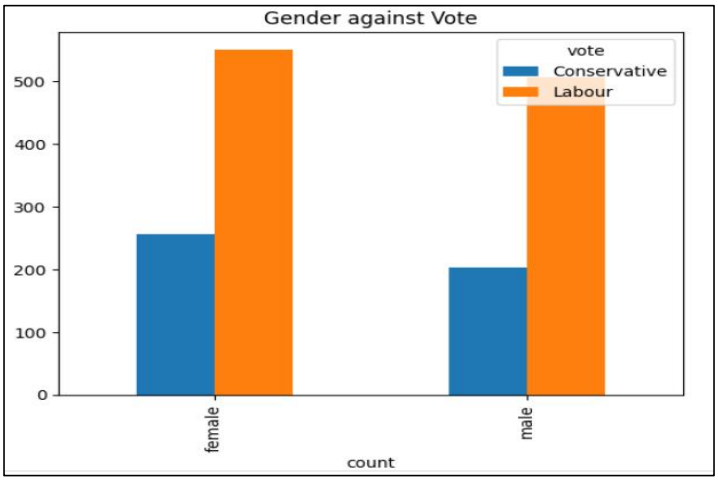

Fig 1.12 Gender against vote

Insights:

1. **Gender alone seems to have no significant impact on the votes**. Although one can say that the female percentage that have voted for the conservatives is slightly more when compared with the labour party.

Numerical VS Categorical Variable:

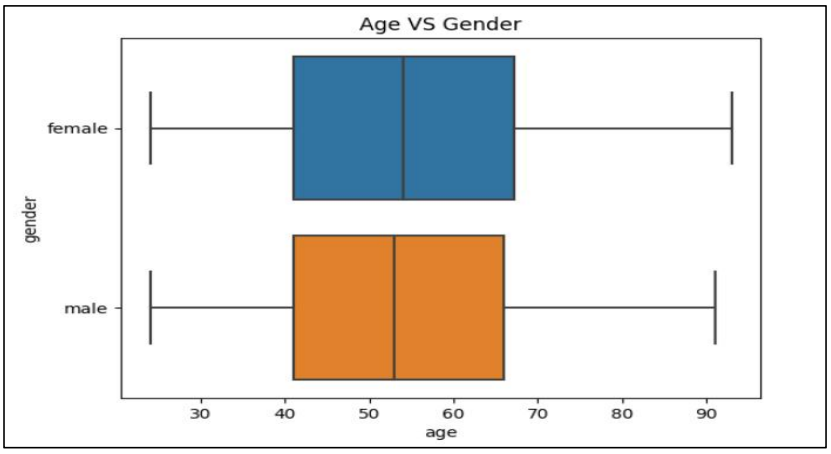

Fig 1.12 Age VS Gender

Insights:

* Boxplots are alike, suggesting the sampling for analysis is unbiased and random, ensuring data Reliability

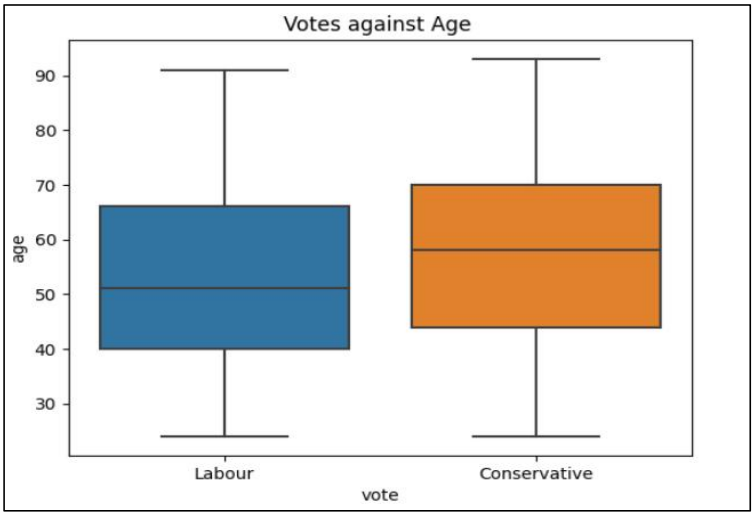

Fig 1.13 Age VS Votes

Insights:

* **Younger to middle-aged voters** (under 50) favour the **Labour Party**, as shown by median and probability distribution. Conversely, **voters over 60 tend to prefer the Conservative Party**

economic.cond.national	1	2	3	4	5
vote					
Conservative	56.76	54.69	32.95	16.91	10.98
Labour	43.24	45.31	67.05	83.09	89.02

Table 1.8 Cross table of National economic condition rating against Votes

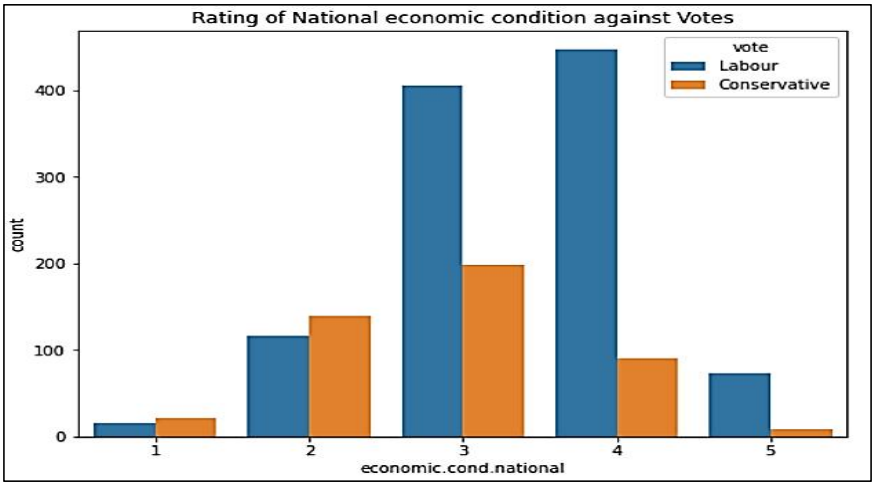

Fig 1.14 Rating of National economic condition against Votes

Insights:

* **The plot confirms Labour's majority vote** and highlights that **voters rating national economic conditions as average or above** lean towards Labour. This echoes the previously observed positive correlation with Labour leader's ratings

economic.cond.household	1	2	3	4	5
vote					
Conservative	43.08	45.0	30.54	19.77	25.0
Labour	56.92	55.0	69.46	80.23	75.0

Table 1.9 Cross table of Household economic condition rating against Votes

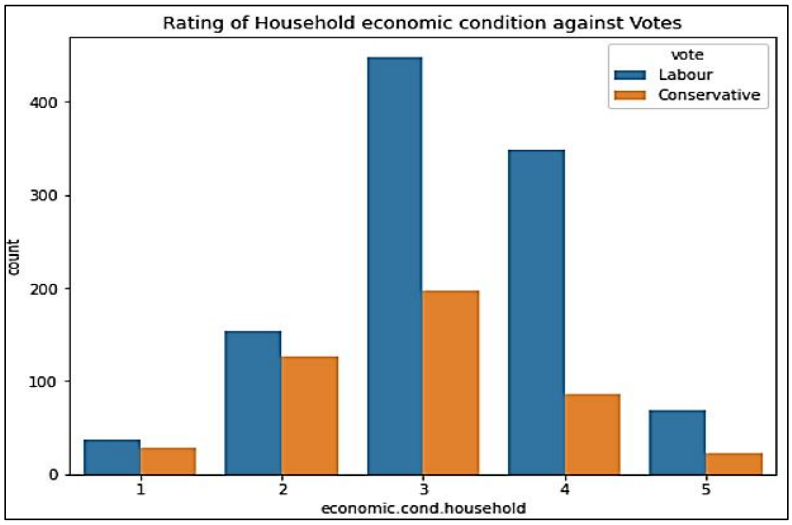

Fig 1.15 Rating of Household economic condition against Votes

Insights:

* **Majority of voters rating their household economic conditions as average or above prefer Labour**. This noticeable vote difference for these ratings between Labour and Conservatives reinforces Labour's appeal.

Blair	1	2	3	4	5
vote					
Conservative	60.82	55.3	100.0	18.85	1.97
Labour	39.18	44.7	0.0	81.15	98.03

Table 1.10 Cross table of Rating of Blair against Votes

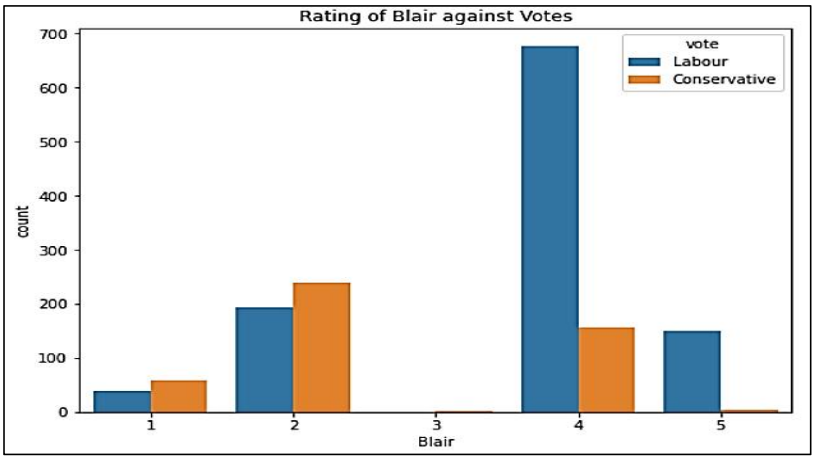

Fig 1.16 Rating of Labour leader Blair against Votes

Insights:

* **Voters giving high ratings to Tony Blair typically support Labour**, while those rating him low typically vote Conservative, as expected.

Hague	1	2	3	4	5
vote					
Conservative	4.72	15.4	24.32	51.35	80.82
Labour	95.28	84.6	75.68	48.65	19.18

Table 1.11 Cross table of Rating of Hague against Votes

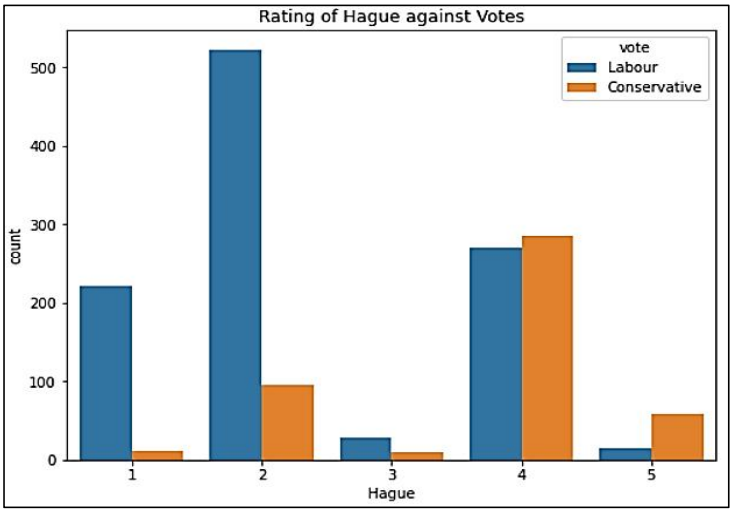

Fig 1.17 Rating of Conservative leader Hague against Votes

Insights:

* **Voters who rated William Hague highly tend to vote Conservative**, while those giving him low ratings generally support Labour.

Europe	1	2	3	4	5	6	7	8	9	10	11
vote											
Conservative	4.59	7.79	10.94	14.29	16.26	16.91	37.21	43.24	50.45	53.47	50.89
Labour	95.41	92.21	89.06	85.71	83.74	83.09	62.79	56.76	49.55	46.53	49.11

Table 1.12 Cross table of Eurosceptic sentiment against Votes

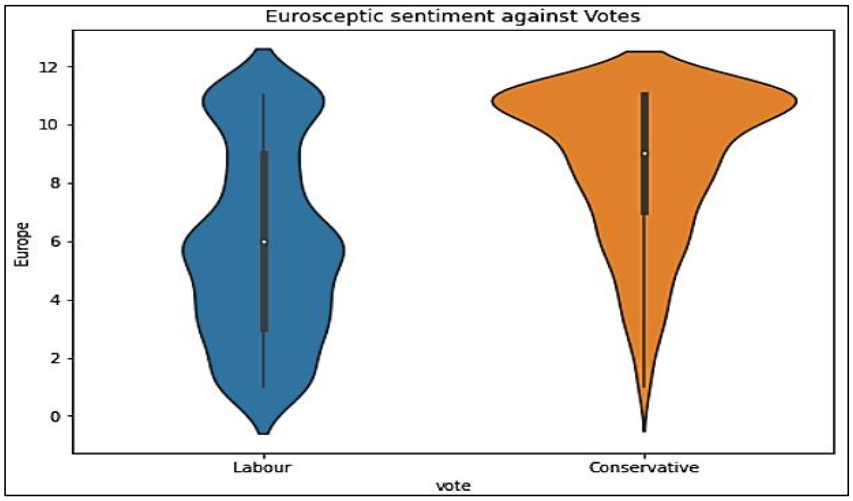

Fig 1.18 Eurosceptic sentiment against Votes

Insights:

* **Voters desiring UK's EU membership favour Labour**, while Eurosceptics divide their votes between both Labour and Conservative parties

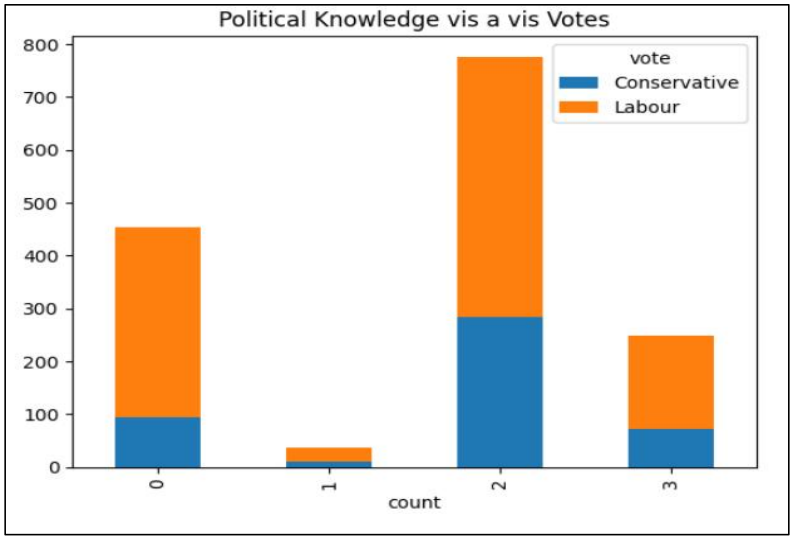

Fig 1.19 Political knowledge vis a vis Votes

Insights:

* Both politically unaware individuals and those with strong knowledge of their parties' stance on Europe largely vote for Labour

Outlier Check: Considering only the 'age' variable for outliers, as other features are ordinal, no outliers were detected

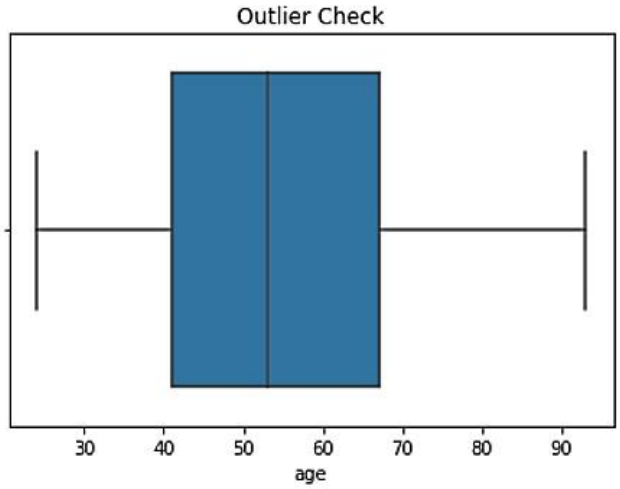

Fig 1.20 Outlier Check – age

There are no outliers in the age variable.

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). 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.

One-hot encoding is applied solely to the 'gender' variable, dropping the dummy variable, as all other independent features are numerical

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

Table 1.13 Dataset after One Hot Encoding

Numerical variables are scaled to avoid inaccuracies in distance-based algorithms like KNN. Though tree-based algorithms and other techniques used in this study are minimally affected by scaling, it's still performed for accuracy

	range	std
age	69.0	15.70
economic.cond.national	4.0	0.88
economic.cond.household	4.0	0.93
Blair	4.0	1.17
Hague	4.0	1.23
Europe	10.0	3.30
political.knowledge	3.0	1.08

Table 1.14 Measures of dispersion of numerical variables

Observing the table, it is evident that the **'age'** variable has a **larger scale compared to the other** rating variables.

Therefore, the **Min-Max scaling method** is chosen for scaling, considering the presence of mostly ordinal and encoded variables in the dataset.

	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
0	Labour	0.275362	0.50	0.50	0.75	0.00	0.1	0.666667	0
1	Labour	0.173913	0.75	0.75	0.75	0.75	0.4	0.666667	1
2	Labour	0.159420	0.75	0.75	1.00	0.25	0.2	0.666667	1
3	Labour	0.000000	0.75	0.25	0.25	0.00	0.3	0.000000	0
4	Labour	0.246377	0.25	0.25	0.00	0.00	0.5	0.666667	1

Table 1.15 Dataset after Min Max scaling

We now split the data into train & test set with 30% of the records going to the test set. The train set has 1061 records whereas the test set has 456 records.

1.4 Apply Logistic Regression and LDA (Linear Discriminant Analysis). Interpret the inferences of both models.

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)

Considering the skewed dataset and the costly nature of both type I and type II errors, the evaluation of model performance should prioritize the F1 score for both classes, in addition to accuracy.

Logistic Regression:

Logistic Regression is a supervised learning technique for classification. It is a type of discriminative classifier that establishes relationship between dependent class variables and the independent variables using regression.

We build a logistic regression model using the default solver 'lbfgs' to predict the train & test labels.

	precision	recall	f1-score	support
Conservative	0.76	0.63	0.69	307
Labour	0.86	0.92	0.89	754
accuracy			0.83	1061
macro avg	0.81	0.77	0.79	1061
weighted avg	0.83	0.83	0.83	1061

Table 1.16 Classification Report – Logistic Regression - Train

	precision	recall	f1-score	support
Conservative	0.76	0.71	0.73	153
Labour	0.86	0.89	0.87	303
accuracy			0.83	456
macro avg	0.81	0.80	0.80	456
weighted avg	0.82	0.83	0.83	456

Table 1.17 Classification Report – Logistic Regression - Test

Model Performance Metrics	Train			Test		
	F1-Conservative	F1 – Labour	Accuracy	F1-Conservative	F1 – Labour	Accuracy
	0.69	0.89	0.83	0.73	0.87	0.83

Table 1.18 Model Performance Summary – Logistic Regression

Since the the data is scaled, the coefficients of the Logit Function will give the feature importance.

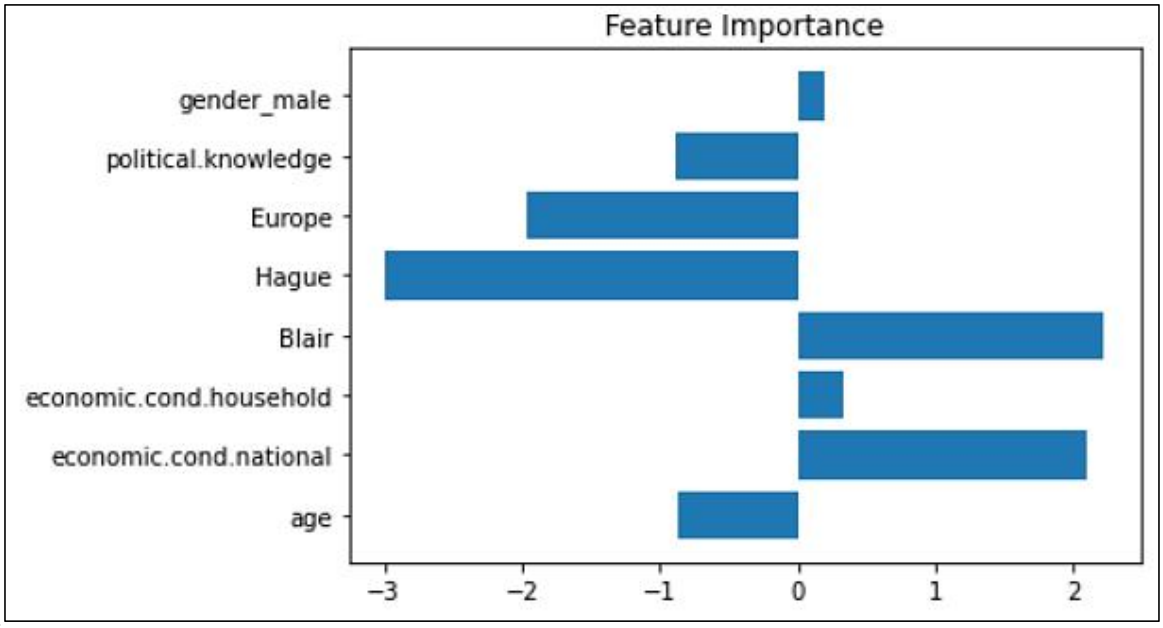

Fig 1.21 Feature Importance - Logistic Regression

Insights:

* The model's precision, recall, accuracy, and AUC on the training and testing data are comparable and high, **indicating no overfitting or underfitting**. The **model is suitable for making predictions**.

* The model **demonstrates better performance in predicting** the majority class, while its **performance for the minority class is relatively lower**.

* The **four most influential features** for classification are the **ratings of Hague and Blair, ratings of national economic conditions, and Eurosceptic sentiment**

Linear Discriminant Analysis:

Linear Discriminant Analysis predicts the class in dependent variable using a linear combination of independent variables. LDA projects the features in higher-dimensional space onto a lower dimensional space. LDA then searches for a linear combination of independent variables (line, plane, or hyperplane) that best separates the classes of the dependent variable.

We build a LDA model & use it for predicting the train & test labels based on the independent variable values.

	precision	recall	f1-score	support
Conservative	0.74	0.65	0.69	307
Labour	0.86	0.91	0.89	754
accuracy			0.83	1061
macro avg	0.80	0.78	0.79	1061
weighted avg	0.83	0.83	0.83	1061

Table 1.19 Classification Report – LDA - Train

	precision	recall	f1-score	support
Conservative	0.77	0.73	0.74	153
Labour	0.86	0.89	0.88	303
accuracy			0.83	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.83	0.83	456

Table 1.20 Classification Report – LDA – Test

	Train	Test

Performance Metrics	F1-Conservative	F1 – Labour	Accuracy	F1-Conservative	F1 – Labour	Accuracy
	0.69	0.89	0.83	0.74	0.88	0.83

Table 1.21 Model Performance Summary – LDA

Feature importance :

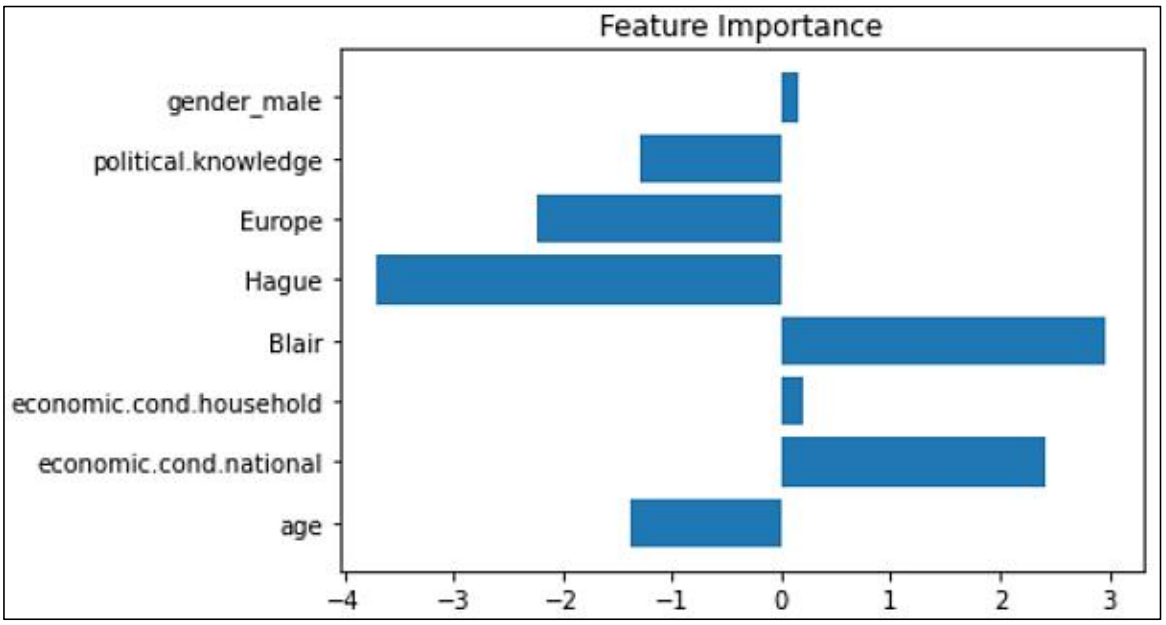

Fig 1.22 Feature Importance - LDA

Insights:

* The precision, recall, accuracy, and AUC of the model on both training and testing data are **high and consistent**, indicating **no overfitting or underfitting**. The **model is suitable for making predictions.**

* The **model's performance** is **similar** to that of a **Logistic Regression model**, with **better predictions for the majority class** and relatively **poorer performance for the minority class**.

* According to the LDA model, the **four most important features** for classification remain the same: **ratings of Hague and Blair, ratings of national economic conditions, and Eurosceptic sentiment**

1.5 Apply KNN Model and Naïve Bayes Model. Interpret the inferences of each model.

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)

K – Nearest Neighbour:

KNN is a non-parametric method based on the principle that a data point is most similar to its neighbouring data points. The 'k' nearest neighbouring data points of a point is computed by calculating the distance between the point and all other points. The class of most of its nearest neighbours is then assigned to the point.

We build a KNN classifier using the default number of nearest neighbours, i.e., 5 & assign weights as per the distance (inverse distance weights) to predict the train and test labels.

	precision	recall	f1-score	support
Conservative	1.00	1.00	1.00	307
Labour	1.00	1.00	1.00	754
accuracy			1.00	1061
macro avg	1.00	1.00	1.00	1061
weighted avg	1.00	1.00	1.00	1061

Table 1.22 Classification Report – KNN - Train

	precision	recall	f1-score	support
Conservative	0.74	0.66	0.70	153
Labour	0.84	0.88	0.86	303
accuracy			0.81	456
macro avg	0.79	0.77	0.78	456
weighted avg	0.80	0.81	0.80	456

Table 1.23 Classification Report – KNN - Test

Model Performance Metrics	Train			Test		
	F1-Conservative	F1 – Labour	Accuracy	F1-Conservative	F1 – Labour	Accuracy
	1	1	1	0.70	0.86	0.81

Table 1.24 Model Performance Summary – KNN

Insights:

* The **model exhibits overfitting** as it **achieves perfect predictions, such as 100% accuracy** and other performance metrics, **on the training data**. However, it **struggles to perform similarly well on the test data**, resulting in a **notable difference in accuracy (>10%).**

Naïve Bayes:

Naïve Bayes is a probabilistic model that is based on the Bayes rule (that helps us in computing Posterior probability using Prior probability). It is called naïve due to the assumption that the predictor variables are mutually independent of each other. If there are continuous independent variables in the dataset then we cannot compute conditional probability. Hence, we use Gaussian Naïve Bayes which uses an additional assumption that the continuous independent variables are normally distributed. Here this assumption holds true since, the age variable can be considered having a normal distribution.

We build a Gaussian Naïve Bayes model & use it for predicting the train & test labels based on the independent variable values.

	precision	recall	f1-score	support
Conservative	0.73	0.69	0.71	307
Labour	0.88	0.90	0.89	754
accuracy			0.84	1061
macro avg	0.80	0.79	0.80	1061
weighted avg	0.83	0.84	0.83	1061

Table 1.25 Classification Report – Naïve Bayes - Train

	precision	recall	f1-score	support
Conservative	0.74	0.73	0.73	153
Labour	0.87	0.87	0.87	303
accuracy			0.82	456
macro avg	0.80	0.80	0.80	456
weighted avg	0.82	0.82	0.82	456

Table 1.26 Classification Report – Naïve Bayes – Test

Model Performance Metrics	Train			Test		
	F1-Conservative	F1 – Labour	Accuracy	F1-Conservative	F1 – Labour	Accuracy
	0.71	0.89	0.84	0.73	0.87	0.82

Table 1.27 Model Performance Summary – Naïve Bayes

Insights:

* The precision, recall, accuracy, and AUC of the model on both the training and testing data are **high and consistent**, **indicating no overfitting or underfitting**. The **model is suitable for making predictions**.

* The **model's performance is comparable to Logistic Regression and LDA models**, showing better predictions for the majority class but relatively poorer performance for the minority class

1.6 Model Tuning , Bagging and Boosting.

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.

To address the issue of lower performance for the minority class, we can utilize model tuning with a scoring parameter that focuses on **improving the F1-score for the minority class.**

To achieve this, we need to **custom encode the classes**, **assigning the minority class** (Conservative) as our class of interest (1) and the majority class (Labour) as the other class (0).

Logistic Regression:

We use Grid Search Cross Validation Technique to obtain the best fit logistic regression model.

We began the search for the best fit model with the following values:

Solver: newton-cg, lbfgs, liblinear, sag

Tolerance (learning rate): 0.0001, 0.00001, 0.000001

We obtain the **best fit model** with the parameters:

Solver = sag

Tolerance = 0.1

The best fit model parameters are then used for creating a Logistic regression model & predicting the train & test labels based on the independent variables.

	precision	recall	f1-score	support
0	0.86	0.92	0.89	754
1	0.77	0.63	0.69	307
accuracy			0.84	1061
macro avg	0.81	0.77	0.79	1061
weighted avg	0.83	0.84	0.83	1061

Table 1.28 Classification Report – Tuned Logistic Regression - Train

	precision	recall	f1-score	support
0	0.86	0.89	0.87	303
1	0.76	0.71	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

Table 1.29 Classification Report – Tuned Logistic Regression – Test

Model Performance Metrics	Train			Test		
	F1-Conservative	F1 – Labour	Accuracy	F1-Conservative	F1 – Labour	Accuracy
	0.69	0.89	0.84	0.74	0.87	0.83

Table 1.30 Model Performance Summary – Tuned Logistic Regression

Feature importance :

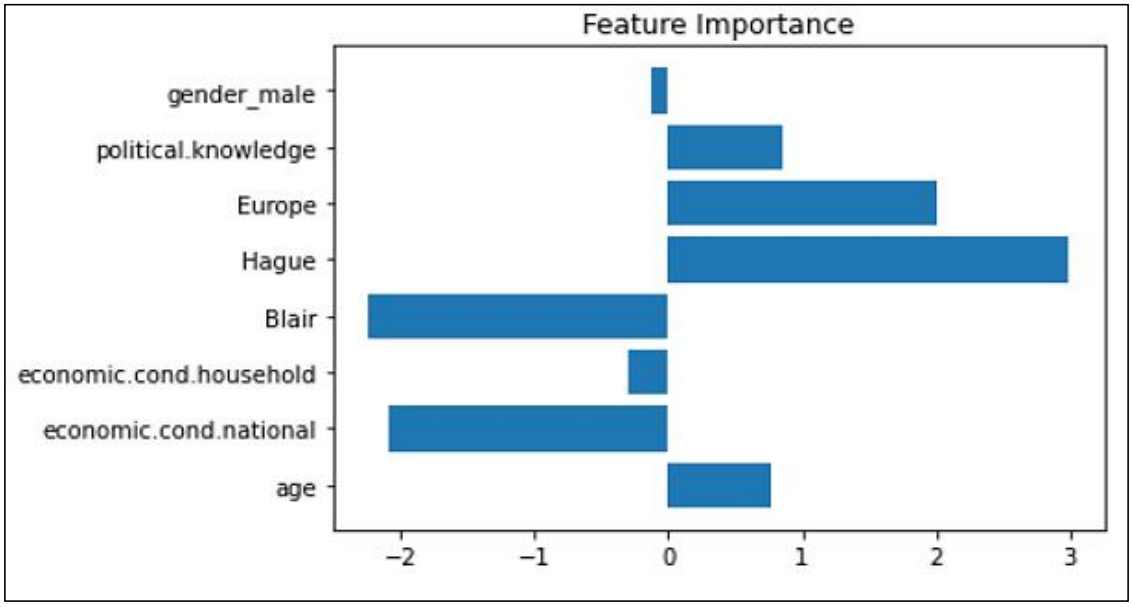

Fig 1.23 Feature Importance – Tuned Logistic Regression

Insights:

* Despite tuning the model based on learning rate and solver, there is **no significant impact on the model's performance.**

* The precision, recall, accuracy, and AUC of the model on both training and testing data are high and consistent, indicating no overfitting or underfitting. The model can be used for making predictions.

* Similar to previous observations, the model demonstrates better performance for the majority class and relatively poorer performance for the minority class.

* The four most important features for classification remain consistent: the ratings of Hague and Blair, the ratings of national economic conditions, and Eurosceptic sentiment

Linear Discriminant Analysis:

We use Grid Search Cross Validation Technique to obtain the best fit LDA model.

We began the search for the best fit model with the following values:

Solver: svd, lsqr, eigen

Tolerance (learning rate): 0.001, 0.0001, 0.00001

We obtain the **best fit model** with the parameters:

Solver = lsqr

Tolerance = 0.1

The best fit model parameters are then used for creating a LDA model & predicting the train & test labels based on the independent variables.

	precision	recall	f1-score	support
0	0.86	0.91	0.89	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

Table 1.31 Classification Report – Tuned LDA - Train

	precision	recall	f1-score	support
0	0.86	0.89	0.87	303
1	0.76	0.71	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

Table 1.32 Classification Report – Tuned LDA– Test

Model Performance Metrics	Train			Test		
	F1-Conservative	F1 – Labour	Accuracy	F1-Conservative	F1 – Labour	Accuracy
	0.69	0.89	0.83	0.74	0.87	0.83

Table 1.33 Model Performance Summary – Tuned Logistic Regression

We look at the feature importance derived from the discriminant function to understand which variables are contributing more to the classification.

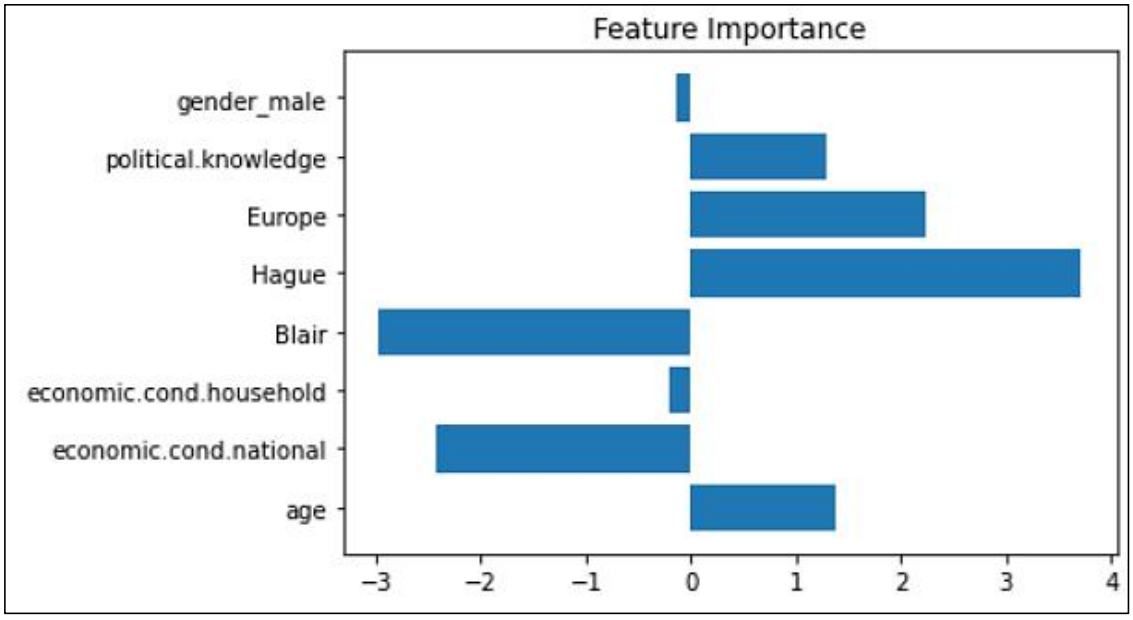

Fig 1.24 Feature Importance – Tuned LDA

Insights:

* Despite tuning the model based on the learning rate and solver, there is **no significant impact observed**.

* The precision, recall, accuracy, and AUC of the model on both training and testing data are high and consistent, indicating no signs of overfitting or underfitting. The model is reliable for making predictions.

* Similar to previous findings, the model exhibits better predictions for the majority class and relatively poorer performance for the minority class.

* The four most influential features for classification remain consistent: ratings of Hague and Blair, ratings of national economic conditions, and Eurosceptic sentiment.

K – Nearest Neighbour:

We search for the optimum k - value for both uniform weights (default weight value – all data points have equal influence) & inverse distance weights (the nearer data points will have a higher influence) by computing the train & test accuracies to find out for which k values the model is performing equally well on both train & test data.

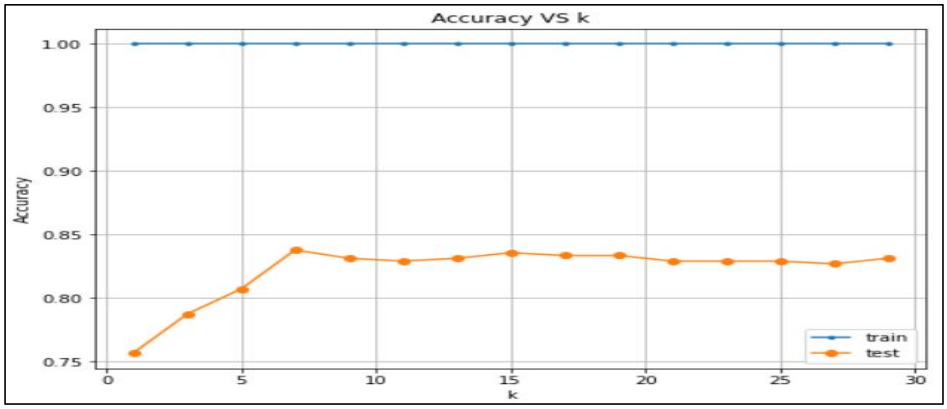

Fig 1.25 Accuracy VS k – inverse distance weights

Obtaining overfit models for all k values when weights are given as inverse of distance (implying the nearer ones will have a higher influence)

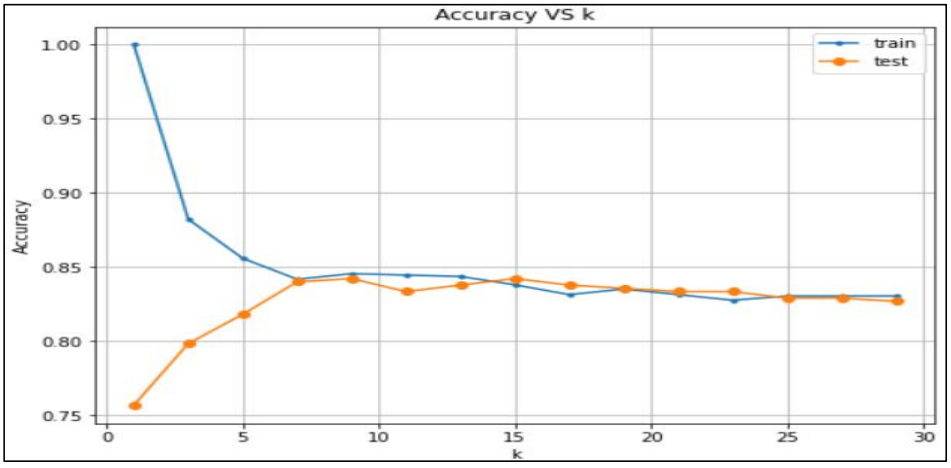

Fig 1.26 Accuracy VS k – uniform weights

When using uniform weights, overfitting of the model seems to have resolved for k $\geqslant$ 7. From the graph it is very clear that the highest train & test accuracies (least misclassifications) are obtained when k = 7, 9, 13, 15. Post k = 15 the accuracy of the model is decreasing for both train & test set.

We feed k = 7, 9, 13 & 15 with uniform weights into the Grid Search Cross Validation function to obtain the best fit KNN classifier. The **optimum k – value as obtained is 7** from the Grid Search Cross Validation Technique.

	precision	recall	f1-score	support
0	0.88	0.90	0.89	754
1	0.74	0.70	0.72	307
accuracy			0.84	1061
macro avg	0.81	0.80	0.80	1061
weighted avg	0.84	0.84	0.84	1061

Table 1.34 Classification Report – Tuned KNN - Train

	precision	recall	f1-score	support
0	0.87	0.90	0.88	303
1	0.78	0.73	0.75	153
accuracy			0.84	456
macro avg	0.82	0.81	0.82	456
weighted avg	0.84	0.84	0.84	456

Table 1.35 Classification Report – Tuned KNN - Test

Model Performance Metrics	Train			Test		
	F1-Conservative	F1 – Labour	Accuracy	F1-Conservative	F1 – Labour	Accuracy
	0.72	0.89	0.84	0.75	0.88	0.84

Table 1.36 Model Performance Summary – Tuned KNN

Insights:

* The precision, recall, accuracy, and AUC of the K-nearest neighbors (KNN) algorithm on both training and testing data are high and consistent, indicating no signs of overfitting or underfitting. The KNN algorithm is suitable for making predictions.

* Comparing with previous models, **the KNN algorithm slightly outperforms in terms of accuracy** and has the **best F1-score for the minority class** among the models evaluated. However, **it still exhibits a slightly inferior performance for the minority class.**

Bagging (Random Forest):

Bagging also known as bootstrap aggregation is an ensemble technique that trains multiple complex models (overfit models) in parallel and tries to smooth out their predictions by aggregating. Random Forest is a type of bagging classifier that ensembles multiple decision trees.

We build a random forest classifier & search for the best fit model, using Grid Search Cross Validation Technique with the following values:

Max depth: 7, 9, 11 => Depth of each decision tree

Max features: 3, 4, 5, 6 => based on $\sqrt{\text{(No. of Independent Variables)}}$

Min samples leaf: 10, 15, 20 => based on 1 - 3 % of the records

Min samples split: 30, 45, 60 => based on 3 times min sample split

No. of estimators: 101, 301, 501 => No. of Decision Trees

Class Weight: {0: 1, 1: 1}, {0: 1, 1: 1.5}, {0: 1, 1: 2.3}, {0: 1, 1: 2.5} => Based on Count of Labour/Count of conservative = 2.3

We obtain the **best fit model** with the parameters:

Max depth = 9

Max features = 2

Min samples leaf = 5

Min samples split = 30

No. of estimators = 101

Class Weight = {0: 1, 1: 2}

The best fit model parameters are then used for creating a random forest classifier model & predicting the train & test labels based on the independent variable values.

	precision	recall	f1-score	support
0	0.92	0.87	0.90	754
1	0.72	0.82	0.77	307
accuracy			0.86	1061
macro avg	0.82	0.84	0.83	1061
weighted avg	0.86	0.86	0.86	1061

Table 1.37 Classification Report – Bagging (Random Forest) - Train

	precision	recall	f1-score	support
0	0.89	0.85	0.87	303
1	0.72	0.79	0.76	153
accuracy			0.83	456
macro avg	0.81	0.82	0.81	456
weighted avg	0.83	0.83	0.83	456

Table 1.38 Classification Report – Bagging (Random Forest) – Test

Model Performance Metrics	Train			Test		
	F1-Conservative	F1 – Labour	Accuracy	F1-Conservative	F1 – Labour	Accuracy
	0.77	0.90	0.86	0.76	0.87	0.83

Table 1.39 Model Performance Summary – Bagging (Random Forest)

Feature importance -:

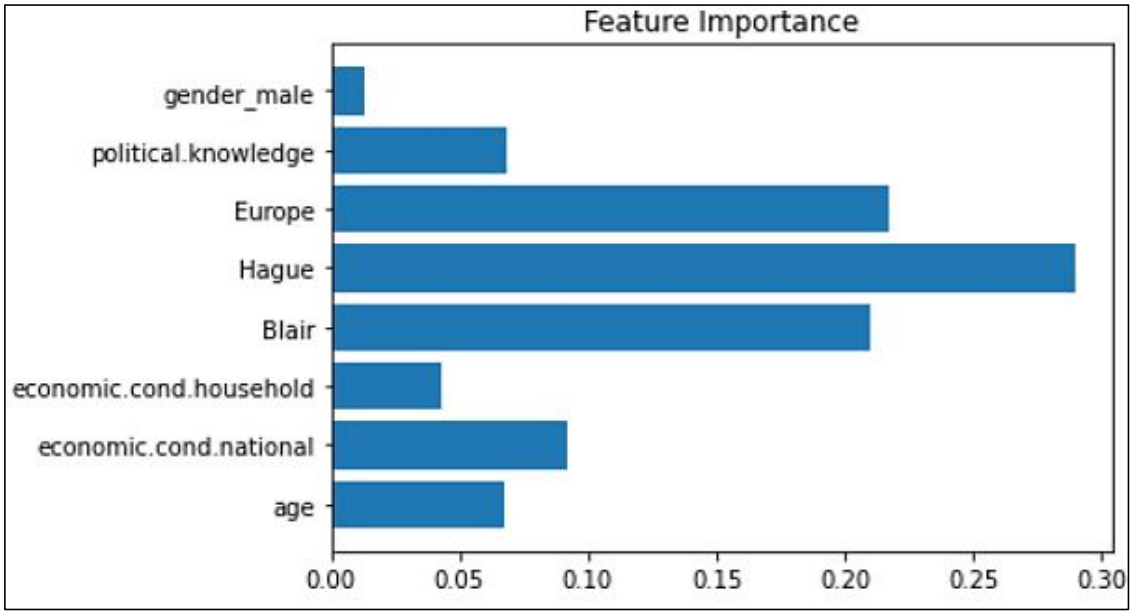

Fig 1.27 Feature Importance – Bagging (Random Forest)

Insights:

* The precision, recall, accuracy, and AUC of the model on both training and testing data are high and consistent, indicating no overfitting or underfitting. The model is suitable for making predictions.

* Similar to previous models, the **model demonstrates better predictions for the majority class and slightly inferior performance for the minority class**. However, it **exhibits a slightly better prediction** for the **minority class compared to the preceding models**.

* The ratings of the **prime ministerial candidates and Eurosceptic sentiment** are the **most important features for classification.**

Boosting:

Boosting trains, a large number of simple models (weak/ underfitting models) in sequence & combines them into a single strong model.

Adaptive Boosting:

Training data remains same during the creation of successive models only weights are added to the misclassified datapoints of the previous model & reduced for the correctly classified datapoints of the previous model.

We build an adaptive boosting classifier using the default simple model as base estimator & search for the appropriate number of simple models using the Grid Search Cross Validation Technique.

No. of estimators: 21, 51, 101

We obtain the **best fit model** with **No. of estimators = 21**

We then create an adaptive boosting classifier model & predict the train & test labels based on the independent variables.

	precision	recall	f1-score	support
0	0.88	0.92	0.89	754
1	0.77	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

Table 1.40 Classification Report – Adaptive Boosting - Train

	precision	recall	f1-score	support
0	0.86	0.88	0.87	303
1	0.75	0.71	0.73	153
accuracy			0.82	456
macro avg	0.80	0.79	0.80	456
weighted avg	0.82	0.82	0.82	456

Table 1.41 Classification Report – Adaptive Boosting – Test

Model Performance Metrics	Train			Test		
	F1-Conservative	F1 – Labour	Accuracy	F1-Conservative	F1 – Labour	Accuracy
	0.72	0.89	0.85	0.73	0.87	0.82

Table 1.42 Model Performance Summary – Adaptive Boosting

Insights:

* The precision, recall, accuracy, and AUC of the model on both training and testing data are high and consistent, indicating no signs of overfitting or underfitting. The model is suitable for making predictions.

* Similar to other models, the model shows better predictions for the majority class and relatively poorer performance for the minority class.

Gradient Boosting:

Training data is modified for each successive models with the successive models trying to fit the residuals (misclassifications).

We build a gradient boosting classifier using the default parameters. However, we search for the appropriate number of simple models using the Grid Search Cross Validation Technique.

No. of estimators: 21, 51, 101

We obtain the **best fit model** with **No. of estimators = 51**

We then create a gradient boosting classifier model & predict the train & test labels based on the independent variables.

	precision	recall	f1-score	support
0	0.90	0.94	0.92	754
1	0.83	0.74	0.78	307
accuracy			0.88	1061
macro avg	0.86	0.84	0.85	1061
weighted avg	0.88	0.88	0.88	1061

Table 1.43 Classification Report – Gradient Boosting - Train

	precision	recall	f1-score	support
0	0.84	0.91	0.88	303
1	0.79	0.67	0.72	153
accuracy			0.83	456
macro avg	0.82	0.79	0.80	456
weighted avg	0.83	0.83	0.82	456

Table 1.44 Classification Report – Gradient Boosting – Test

Model Performance Metrics	Train			Test		
	F1-Conservative	F1 – Labour	Accuracy	F1-Conservative	F1 – Labour	Accuracy
	0.78	0.92	0.88	0.72	0.88	0.83

Table 1.45 Model Performance Summary – Gradient Boosting

Insights:

* The precision, recall, accuracy, and AUC of the model on both training and testing data are high and consistent, indicating no signs of overfitting or underfitting. The model is suitable for making predictions.

* Similar to other models, the model demonstrates better predictions for the majority class and relatively poorer performance for the minority class.

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

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.

Since the Logistic Regression & LDA model showed similar performance even after tuning with the solver & learning rate, we can use any of the tuned or untuned model for comparison with other models. Here, we use the tuned model for comparison with other models.

[Reminder Note: 0 represents Labour, 1 represents Conservative]

Logistic Regression:

Train Set:

	precision	recall	f1-score	support
0	0.86	0.92	0.89	754
1	0.77	0.63	0.69	307
accuracy			0.84	1061
macro avg	0.81	0.77	0.79	1061
weighted avg	0.83	0.84	0.83	1061

Table 1.46 Classification Report – Logistic Regression – Train

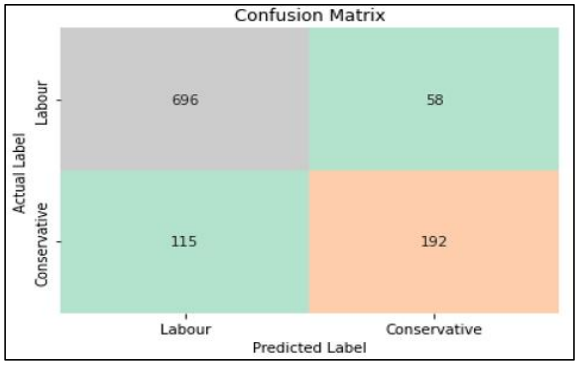

Fig 1.28 Confusion Matrix – Logistic Regression – Train

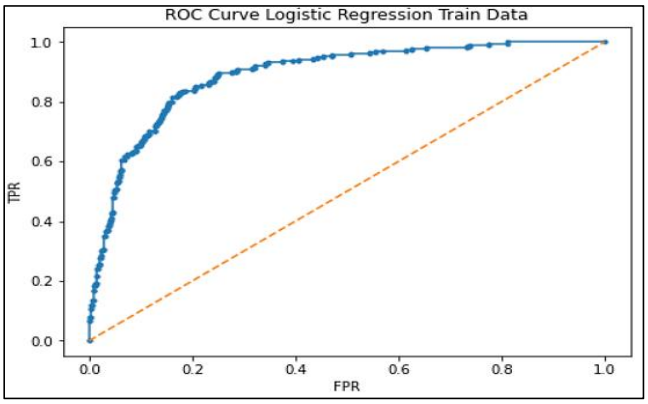

Fig 1.29 ROC curve – Logistic Regression – Train

Test Set:

	precision	recall	f1-score	support
0	0.86	0.89	0.87	303
1	0.76	0.71	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

Table 1.47 Classification Report – Logistic Regression – Test

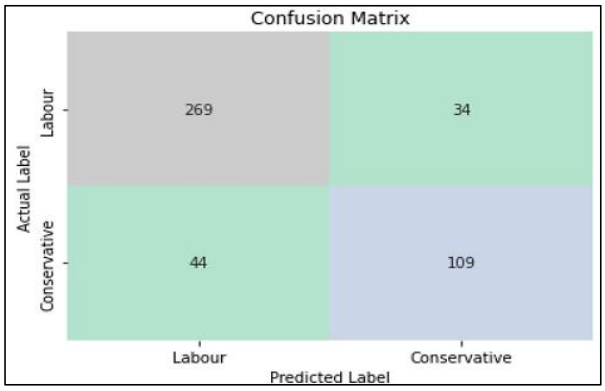

Fig 1.30 Confusion Matrix – Logistic Regression – Test

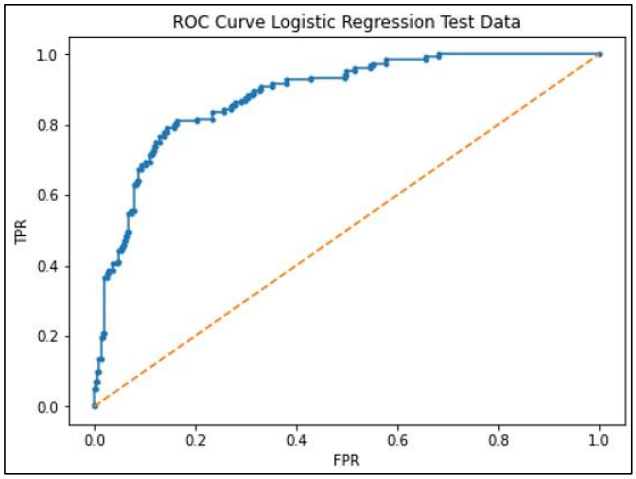

Fig 1.31 ROC curve – Logistic Regression – Test

Model Performance Metrics	Train				Test			
	F1-Conservative	F1 – Labour	Accuracy	AUC	F1-Conservative	F1 – Labour	Accuracy	AUC
	0.69	0.89	0.84	0.89	0.74	0.87	0.83	0.88

Table 1.48 Model Performance Summary – Logistic Regression

Insights:

1. The Precision, Recall, F1, Accuracy & AUC of training data for the model is in line with the testing data and is fairly high. Hence, **no overfitting or underfitting** has occurred & the model can be used for making predictions.
2. The **model is predicting better for the majority class and has a pretty inferior performance for the minority class**.

Linear Discriminant Analysis:

Train Set:

	precision	recall	f1-score	support
0	0.86	0.91	0.89	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

Table 1.49 Classification Report – LDA – Train

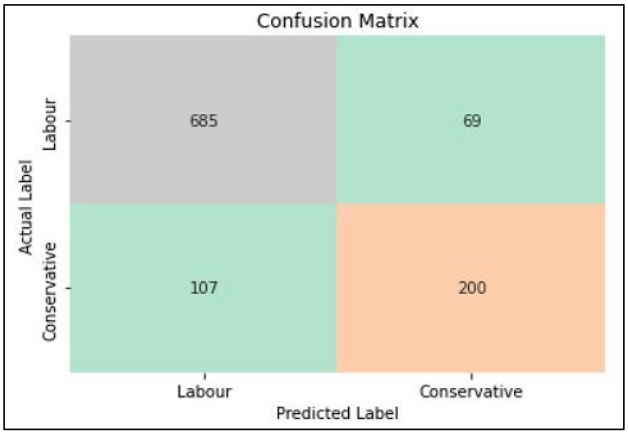

Fig 1.32 Confusion Matrix – LDA – Train

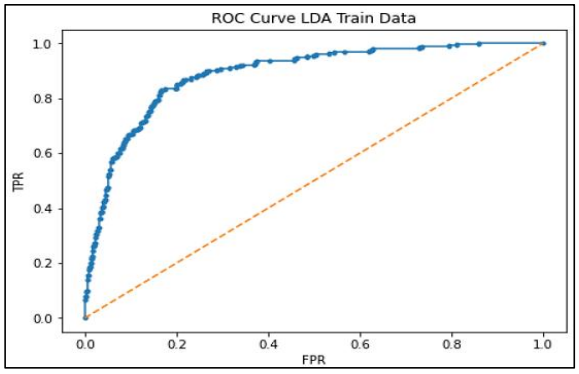

Fig 1.33 ROC curve – LDA – Train

Test Set:

	precision	recall	f1-score	support
0	0.86	0.89	0.87	303
1	0.76	0.71	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

Table 1.50 Classification Report – LDA – Test

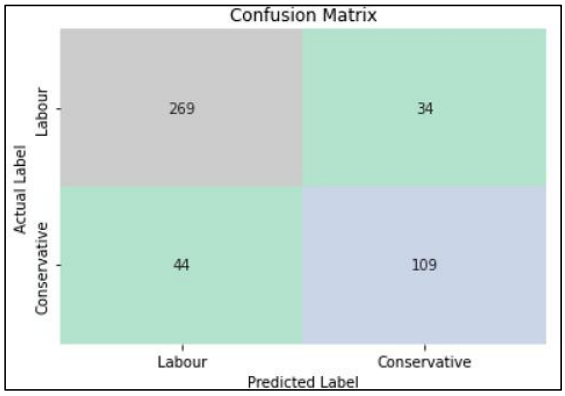

Fig 1.34 Confusion Matrix – LDA – Test

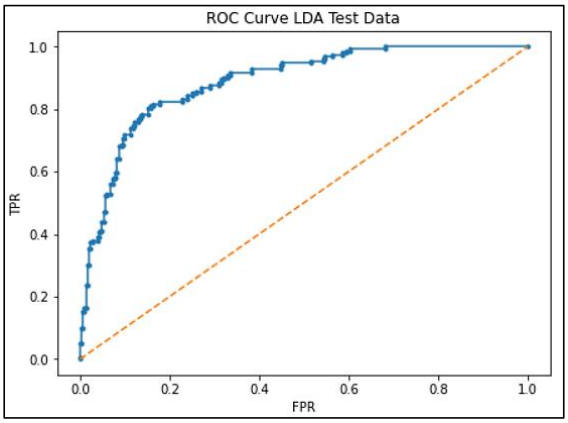

Fig 1.35 ROC curve – LDA – Test

Model Performance Metrics	Train				Test			
	F1-Conservative	F1 – Labour	Accuracy	AUC	F1-Conservative	F1 – Labour	Accuracy	AUC
	0.69	0.89	0.83	0.89	0.74	0.87	0.83	0.89

Table 1.51 Model Performance Summary – LDA

Insights:

1. The Precision, Recall, F1, Accuracy & AUC of training data for the model is in line with the testing data and is fairly high. Hence, **no overfitting or underfitting** has occurred & the model can be used for making predictions.
2. The **model is predicting better for the majority class and has a pretty inferior performance for the minority class**.

K – Nearest Neighbour:

Train Set:

	precision	recall	f1-score	support
0	0.88	0.90	0.89	754
1	0.74	0.70	0.72	307
accuracy			0.84	1061
macro avg	0.81	0.80	0.80	1061
weighted avg	0.84	0.84	0.84	1061

Table 1.52 Classification Report – KNN – Train

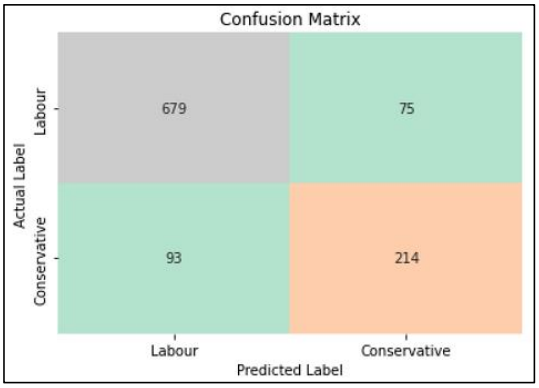

Fig 1.36 Confusion Matrix – KNN – Train

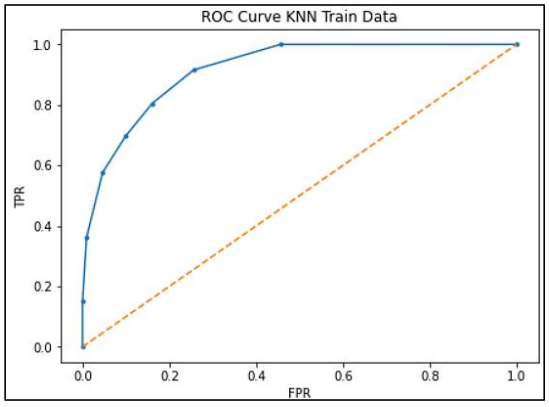

Fig 1.37 ROC curve – KNN – Train

Test Set:

	precision	recall	f1-score	support
0	0.87	0.90	0.88	303
1	0.78	0.73	0.75	153
accuracy			0.84	456
macro avg	0.82	0.81	0.82	456
weighted avg	0.84	0.84	0.84	456

Table 1.53 Classification Report – KNN – Test

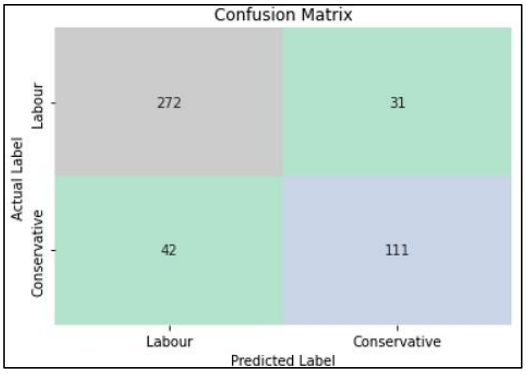

Fig 1.38 Confusion Matrix – KNN – Test

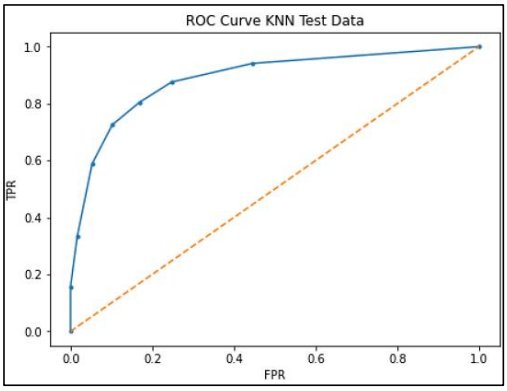

Fig 1.39 ROC curve – KNN – Test

Model Performance Metrics	Train				Test			
	F1-Conservative	F1 – Labour	Accuracy	AUC	F1-Conservative	F1 – Labour	Accuracy	AUC
	0.72	0.89	0.84	0.92	0.75	0.88	0.84	0.89

Table 1.54 Model Performance Summary – KNN

Insights:

1. The Precision, Recall, F1, Accuracy & AUC of training data for the model is in line with the testing data and is fairly high. Hence, **no overfitting or underfitting** has occurred & the model can be used for making predictions.
2. The **model is predicting better for the majority class and has a slightly inferior performance for the minority class**.

Naïve Bayes:

Train Set:

	precision	recall	f1-score	support
Conservative	0.73	0.69	0.71	307
Labour	0.88	0.90	0.89	754
accuracy			0.84	1061
macro avg	0.80	0.79	0.80	1061
weighted avg	0.83	0.84	0.83	1061

Table 1.55 Classification Report – Naïve Bayes – Train

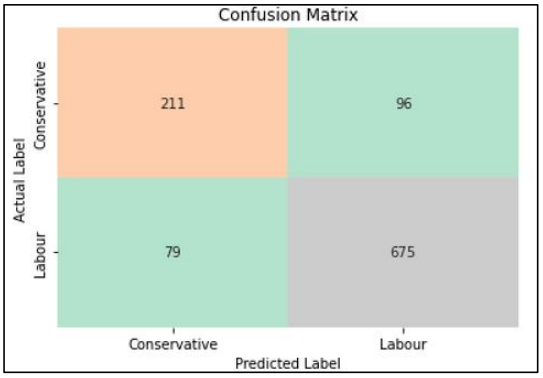

Fig 1.40 Confusion Matrix – Naïve Bayes – Train

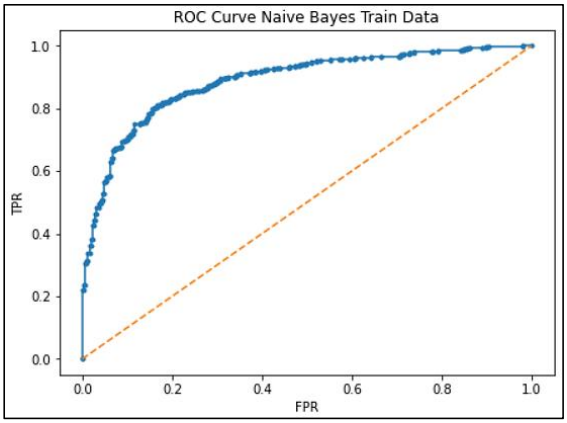

Fig 1.41 ROC curve – Naïve Bayes – Train

Test Set:

	precision	recall	f1-score	support
Conservative	0.74	0.73	0.73	153
Labour	0.87	0.87	0.87	303
accuracy			0.82	456
macro avg	0.80	0.80	0.80	456
weighted avg	0.82	0.82	0.82	456

Table 1.56 Classification Report – Naïve Bayes – Test

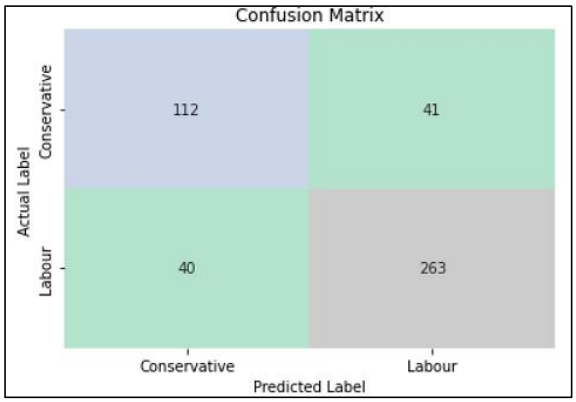

Fig 1.42 Confusion Matrix – Naïve Bayes – Test

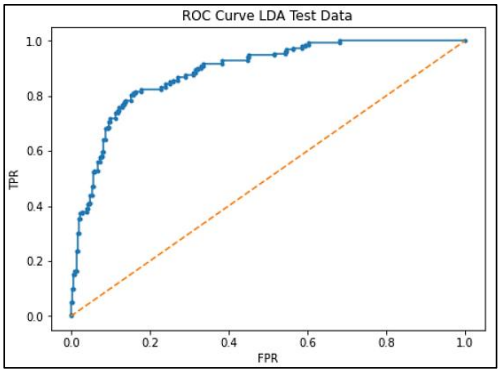

Fig 1.43 ROC curve – Naïve Bayes – Test

Model Performance Metrics	Train				Test			
	F1-Conservative	F1 – Labour	Accuracy	AUC	F1-Conservative	F1 – Labour	Accuracy	AUC
	0.71	0.89	0.84	0.89	0.73	0.87	0.82	0.88

Table 1.57 Model Performance Summary – Naïve Bayes

Insights:

1. The Precision, Recall, F1, Accuracy & AUC of training data for the model is in line with the testing data and is fairly high. Hence, **no overfitting or underfitting** has occurred & the model can be used for making predictions.
2. The **model is predicting better for the majority class and has a pretty inferior performance for the minority class**.

Bagging (Random Forest):

Train Set:

	precision	recall	f1-score	support
0	0.92	0.87	0.90	754
1	0.72	0.82	0.77	307
accuracy			0.86	1061
macro avg	0.82	0.84	0.83	1061
weighted avg	0.86	0.86	0.86	1061

Table 1.58 Classification Report – Bagging (Random Forest) – Train

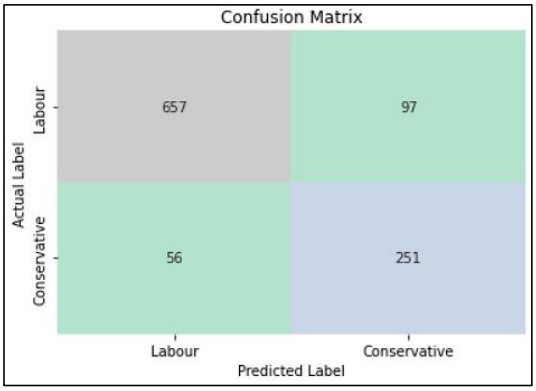

Fig 1.44 Confusion Matrix – Bagging (Random Forest) – Train

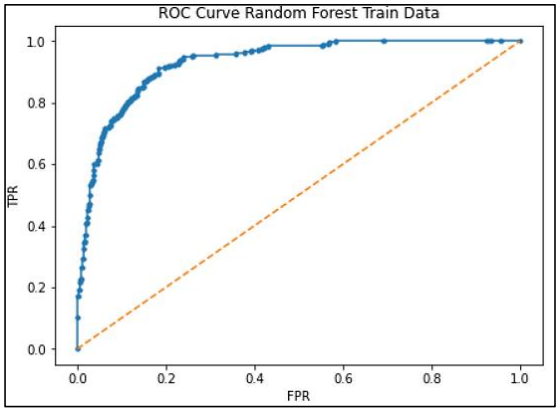

Fig 1.45 ROC curve – Bagging (Random Forest) – Train

Test Set:

	precision	recall	f1-score	support
0	0.89	0.85	0.87	303
1	0.72	0.79	0.76	153
accuracy			0.83	456
macro avg	0.81	0.82	0.81	456
weighted avg	0.83	0.83	0.83	456

Table 1.59 Classification Report – Bagging (Random Forest) – Test

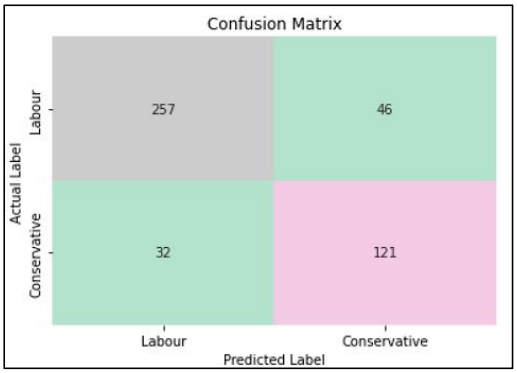

Fig 1.46 Confusion Matrix – Bagging (Random Forest) – Test

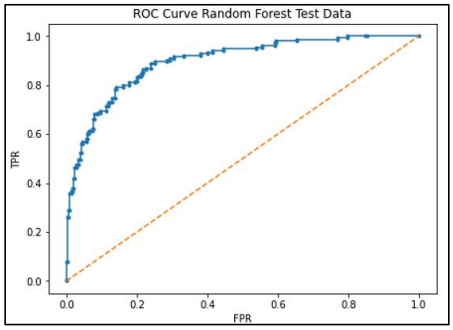

Fig 1.47 ROC curve – Bagging (Random Forest) – Test

Model Performance Metrics	Train				Test			
	F1-Conservative	F1 – Labour	Accuracy	AUC	F1-Conservative	F1 – Labour	Accuracy	AUC
	0.77	0.90	0.86	0.93	0.76	0.87	0.83	0.90

Table 1.60 Model Performance Summary – Bagging (Random Forest)

Insights:

1. The Precision, Recall, F1, Accuracy & AUC of training data for the model is in line with the testing data and is fairly high. Hence, **no overfitting or underfitting** has occurred & the model can be used for making predictions.

2. The model like other models is predicting better for the majority class and has a slightly inferior performance for the minority class. However, **the prediction for minority classes is slightly better than all the other preceding models**.

Adaptive Boosting:

Train Set:

	precision	recall	f1-score	support
0	0.88	0.92	0.89	754
1	0.77	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

Table 1.61 Classification Report – Adaptive Boosting – Train

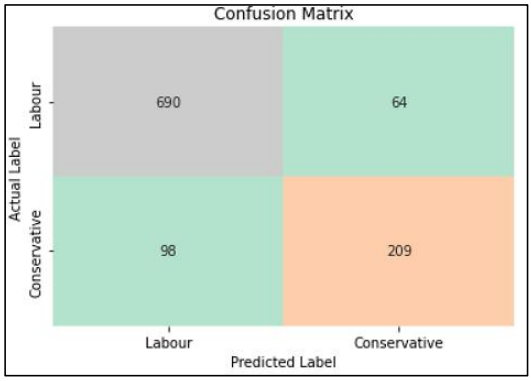

Fig 1.48 Confusion Matrix – Adaptive Boosting – Train

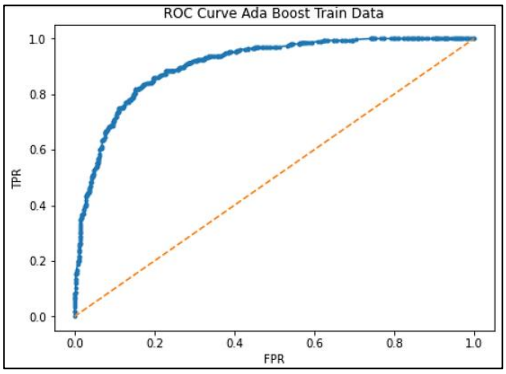

Fig 1.49 ROC curve – Adaptive Boosting – Train

Test Set:

	precision	recall	f1-score	support
0	0.86	0.88	0.87	303
1	0.75	0.71	0.73	153
accuracy			0.82	456
macro avg	0.80	0.79	0.80	456
weighted avg	0.82	0.82	0.82	456

Table 1.62 Classification Report – Adaptive Boosting – Test

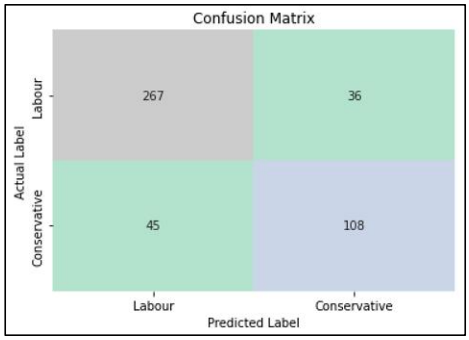

Fig 1.50 Confusion Matrix – Adaptive Boosting – Test

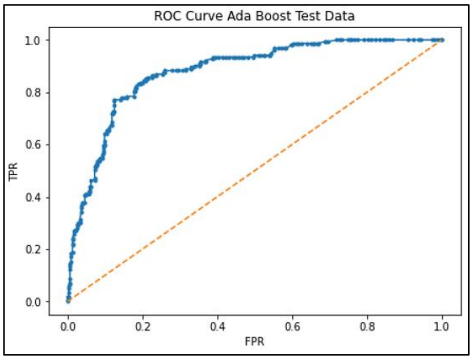

Fig 1.51 ROC curve – Adaptive Boosting – Test

Model Performance Metrics	Train				Test			
	F1-Conservative	F1 – Labour	Accuracy	AUC	F1-Conservative	F1 – Labour	Accuracy	AUC
	0.72	0.89	0.85	0.91	0.73	0.87	0.82	0.88

Table 1.63 Model Performance Summary – Adaptive Boosting

Insights:

1. The Precision, Recall, F1, Accuracy & AUC of training data for the model is in line with the testing data and is fairly high. Hence, **no overfitting or underfitting** has occurred & the model can be used for making predictions.
2. The model like other models is **predicting better for the majority class and has a pretty inferior performance for the minority class**.

Gradient Boosting:

Train Set:

	precision	recall	f1-score	support
0	0.90	0.94	0.92	754
1	0.83	0.74	0.78	307
accuracy			0.88	1061
macro avg	0.86	0.84	0.85	1061
weighted avg	0.88	0.88	0.88	1061

Table 1.64 Classification Report – Gradient Boosting – Train

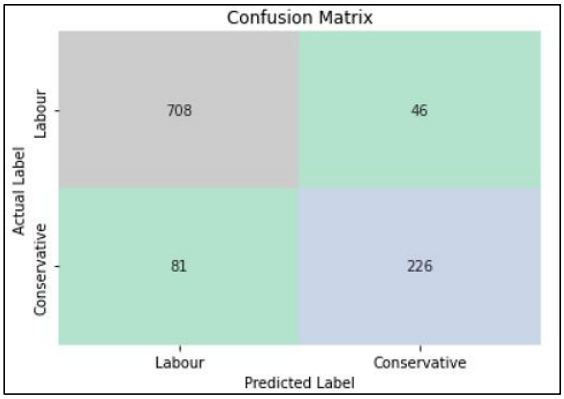

Fig 1.52 Confusion Matrix – Gradient Boosting – Train

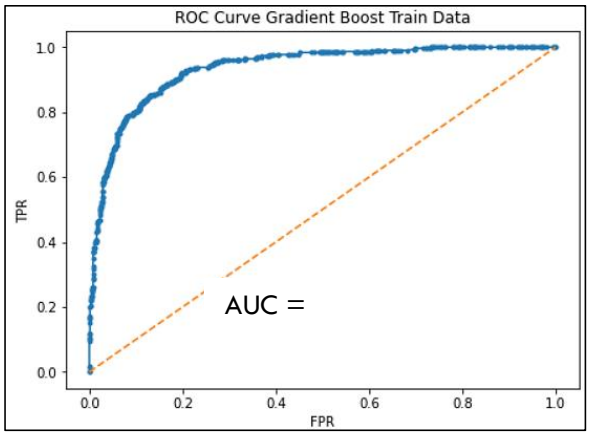

Fig 1.53 ROC curve – Gradient Boosting – Train

Test Set:

	precision	recall	f1-score	support
0	0.84	0.91	0.88	303
1	0.79	0.67	0.72	153
accuracy			0.83	456
macro avg	0.82	0.79	0.80	456
weighted avg	0.83	0.83	0.82	456

Table 1.65 Classification Report – Gradient Boosting – Test

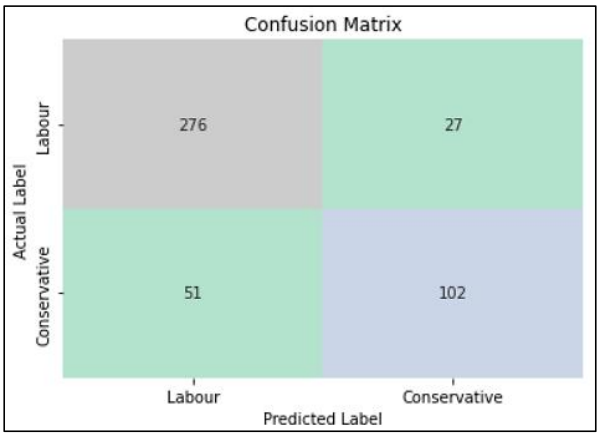

Fig 1.54 Confusion Matrix – Gradient Boosting – Test

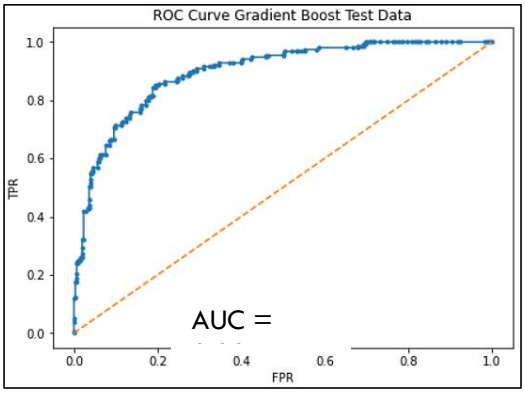

Fig 1.55 ROC curve – Gradient Boosting – Test

Model Performance Metrics	Train				Test			
	F1-Conservative	F1 – Labour	Accuracy	AUC	F1-Conservative	F1 – Labour	Accuracy	AUC
	0.78	0.92	0.88	0.94	0.72	0.88	0.83	0.90

Table 1.66 Model Performance Summary – Gradient Boosting

Insights:

1. The Precision, Recall, F1, Accuracy & AUC of training data for the model is in line with the testing data and is fairly high. Hence, **no overfitting or underfitting** has occurred & the model can be used for making predictions.
2. The model like other models is **predicting better for the majority class and has a pretty inferior performance for the minority class**.

Final Model:

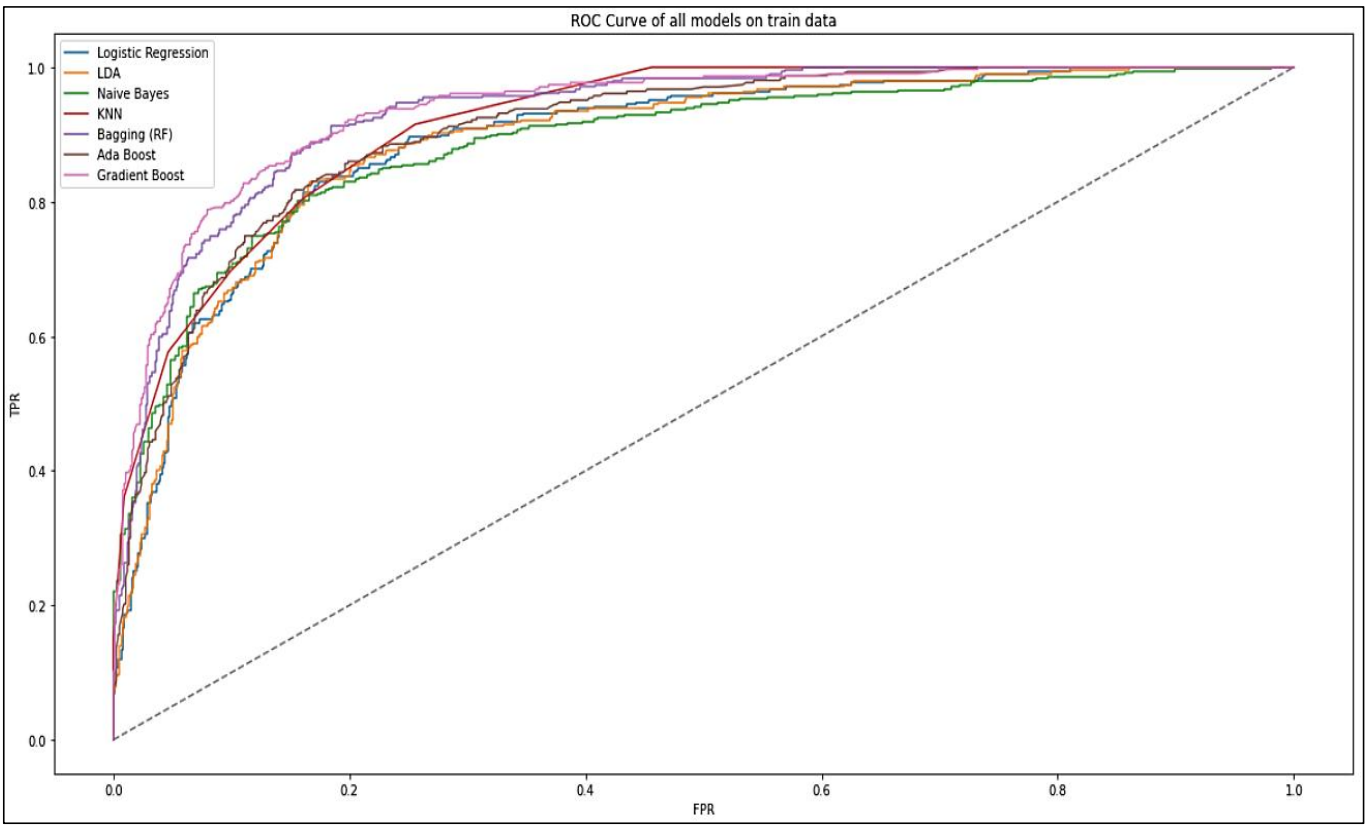

Fig 1.56 ROC curve of all models – Train

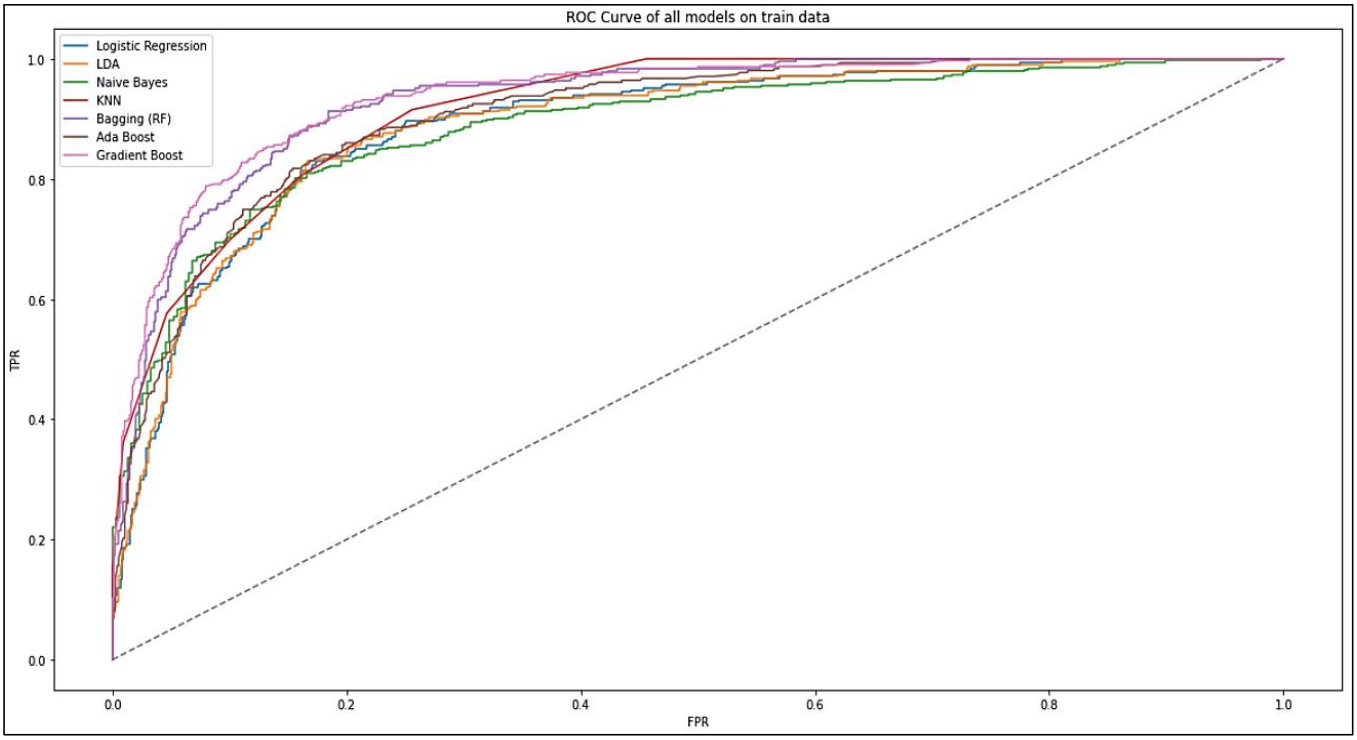

Fig 1.57 ROC curve of all models – Test

Model performance summary of all models: -

	f1_train_conservative	f1_test_conservative	f1_train_labour	f1_test_labour	Accuracy_train	Accuracy_test	AUC_train	AUC_test
Logistic Regression	68.94	73.65	88.95	88.95	83.69	82.89	88.97	88.40
LDA	69.44	73.65	88.62	88.62	83.41	83.33	88.94	88.76
Naive Bayes	70.69	73.44	88.52	88.52	83.51	82.24	88.79	87.64
KNN	71.81	75.25	88.99	88.99	84.17	83.99	91.70	88.83
Bagging (RF)	76.64	75.62	89.57	89.57	85.58	82.89	93.03	89.68
Ada Boost	72.07	72.73	89.49	89.49	84.73	82.24	90.54	87.98
Gradient Boost	78.07	72.34	91.77	91.77	88.03	82.89	93.55	89.67

Table 1.67 Model Performance Summary of all models

	f1_test_conservative	f1_test_labour	Accuracy_test	AUC_test	f1_test_diff
Logistic Regression	73.65	88.95	82.89	88.40	-15.30
LDA	73.65	88.62	83.33	88.76	-14.97
Naive Bayes	73.44	88.52	82.24	87.64	-15.08
KNN	75.25	88.99	83.99	88.83	-13.74
Bagging (RF)	75.62	89.57	82.89	89.68	-13.95
Ada Boost	72.73	89.49	82.24	87.98	-16.76
Gradient Boost	72.34	91.77	82.89	89.67	-19.43

Table 1.68 Model Performance Summary of all models – Test

Best models as per each of the metrics on test data: -

F1_conservative => ***Bagging (RF), KNN***
F1_labour => ***Gradient Boost, Bagging (RF)***
Accuracy => ***KNN, LDA***
AUC => ***Bagging (RF), Gradient Boost***
F1_diff => ***KNN, Bagging (RF)***

Insights:

* **Performance metrics of all models are similar.**

* **High accuracy and AUC scores** are observed for the **majority class (Labour).**

* **Slightly lower performance** is observed for the **minority class (Conservative) across all models**.

* **Random Forest model** shows **slightly better performance** in predicting the **minority class**.

Final Model: RandomForestClassifier (class_weight = {0: 1, 1: 2}, max_depth = 9, max_features = 2, min_samples_leaf = 5, min_samples_split = 30, n_estimators = 101, random_state = 1)

We compare the count of the actual and the predicted labels for the Bagging (Random Forest) Classifier:

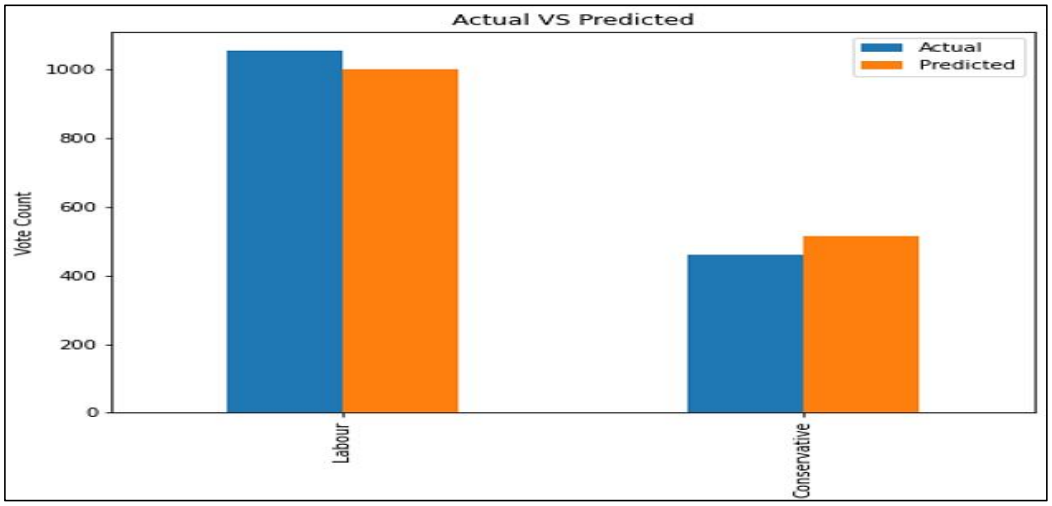

Fig 1.58 Actual VS Predicted Labels – Final Model

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.

Insights:

* **All the models** have **performed fairly well** and have approximately similar performance metrics after tuning.

* **Random Forest is slightly better** as it **performs slightly better in predicting** the **minority class** as compared to other models. The **Random Forest model predicts a slightly higher number of votes for conservatives** & **slightly lower** number of votes for **Labour**. This might help in **adjusting for the bias** in sampling using class weights parameter.

* **Labour is getting twice the number of votes as compared to conservatives**. Thus, **Labour is most likely to come back to power.**

* The **most important features** in classifying are -: The **ratings of each of the candidate**, **Eurosceptic sentiments**, followed by the rating of **national economic condition**, **age** & **political knowledge** of their parties position.

* **Gender plays no significant role** in the classification process but is important to ascertain whether the survey is unbiased.

* People who have **higher Eurosceptic sentiment**, have **voted for the conservative party**.

* **Young to middle aged** voters (< 50 yrs of age) seem **more inclined to vote for Labour party**, whereas people **beyond 60 yrs** of age prefer to vote for the **conservative party**.

* People are **generally happy** with the **national & household economic conditions** with ~**80% voting** it fair to very good and naturally have chosen labour to continue.

Recommendations:

* We can **drop the gender variable** since it has the least feature importance and after ensuring the survey is not biased.

* **Consider adding additional variables** such as **constituency/region, level of education, religion, race, immigrant status**, etc., to **enhance** the predictive power of the model.

* **Including constituency/region data** can help in clustering votes based on **geographical areas** and **enable predicting the range** (confidence interval) of the number of seats each **party is likely to win**.

* If possible, **gather more data points** to further evaluate and improve the model's performance.

Problem 2

Problem Statement:

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

Out of the 59 speeches of presidents in the corpus, we focus on three specific speeches: President Roosevelt's speech in 1941, President JFK's speech in 1961, and President Nixon's speech in 1973.

Our primary objective is to analyze Franklin Roosevelt's speech to gain insights into the usage of symbols and punctuation.

'On each national day of inauguration since 1789, the people have renewed their sense of dedication to the United States.\n\nIn Washington\'s day the task of the people was to create and weld together a nation.\n\nIn Lincoln\'s day the task of the people was to preserve that Nation from disruption from within.\n\nIn this day the task of the people is to save that Nation and its institutions from disruption from without.\n\nTo us there has come a time, in the midst of swift happenings, to pause for a moment and take stock -- to recall what our place in history has been, and to rediscover what we are and what we may be. If we do not, we risk the real peril of inaction.\n\nLives of nations are determined not by the count of years, but by the lifetime of the human spirit. The life of a man is three-score years and ten: a little more, a little less. The life of a nation is the fullness of the measure of its will to live.\n\nThere are men who doubt this. There are men who believe that democra...'

Fig 2.1 Sample text – Franklin D. Roosevelt's speech

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

No. of Characters:

Franklin D. Roosevelt's speech in 1941 :**7571**.

John F. Kennedy's speech in 1961 : **7618**.

Richard Nixon's speech in 1973 : **9991**.

No. of Words:

Franklin D. Roosevelt's speech in 1941 : **1536**.

John F. Kennedy's speech in 1961 : **1546**.

Richard Nixon's speech in 1973 :**2028**.

No. of Sentences:

Franklin D. Roosevelt's speech in 1941 : **68**.

John F. Kennedy's speech in 1961 : **52**.

Richard Nixon's speech in 1973 : **69**.

2.2 Remove all the stop words from all three speeches.

Forming a data frame with the President's speeches with the President's names as index.

Sample of the Dataset:

	Speech
Franklin D. Roosevelt	On each national day of inauguration since 178...
John F. Kennedy	Vice President Johnson, Mr. Speaker, Mr. Chief...
Richard Nixon	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...

Table 2.1 Sample of the Dataset

Text Pre-processing:

Before analyzing text data, we perform essential pre-processing steps to extract valuable insights. These steps include:

- Converting the text to a consistent case (lower or upper).
- Removing special symbols and punctuations.
- Removing extra white spaces (not applicable in any speech).
- Removing stop words.
- Stemming words to their original form.
- Retaining numbers (as they may be relevant, such as the year mentioned in a president's speech).

Lower case conversion:

All the speeches are converted to a lower case. The idea is to bring all of the text to one case either lower or upper because python is case sensitive and it views 'cat' & 'Cat' as different.

	Speech
Franklin D. Roosevelt	on each national day of inauguration since 178...
John F. Kennedy	vice president johnson, mr. speaker, mr. chief...
Richard Nixon	mr. vice president, mr. speaker, mr. chief jus...

Table 2.2 Dataset after lower case conversion

Special Characters & Punctuations Removal:

We utilize the regular expressions library to replace non-word and non-space characters with null. By using the pattern [^\w\s], where '\w' represents letters, numbers, and underscores, and '\s' denotes spaces, we effectively exclude special characters and punctuations (except underscores) from the text data.

	Speech
Franklin D. Roosevelt	on each national day of inauguration since 178...
John F. Kennedy	vice president johnson mr speaker mr chief jus...
Richard Nixon	mr vice president mr speaker mr chief justice ...

Table 2.3 Dataset after special characters & punctuations removal

Stop Words Removal:

Stop words are common words that do not add value or context to the data at hand. We obtain a list of stop words from the Natural Language Toolkit library and add a list of punctuations from the String library (to deal with underscore) to it, along with that we even add a few of our own stop words like let, us, etc. to it. Then we remove the complete list of stop words.

	Speech
Franklin D. Roosevelt	national day inauguration since 1789 people re...
John F. Kennedy	vice president johnson speaker chief justice p...
Richard Nixon	vice president speaker chief justice senator c...

Table 2.4 Dataset after stop words removal

A snippet of Franklin D. Roosevelt's speech after pre-processing: -

'national day inauguration since 1789 people renewed sense dedication united states washingtons day task people create weld together nation lincolns day task people preserve nation di sruption within day task people save nation institutions disruption without come time midst swift happenings pause moment take stock recall place history rediscover may risk real peri l inaction lives nations determined count years lifetime human spirit life man threescore years ten little little less life nation fullness measure live men doubt men believe democrac y form government frame life limited measured kind mystical artificial fate unexplained reason tyranny slavery become surging wave future freedom ebbing tide americans know true eight years ago life republic seemed frozen fatalistic terror proved true midst shock acted acted quickly boldly decisively later years living years fruitful years people democracy brought greater security hope better understanding lifes ideals measured material things vita...'

Fig 2.2 Sample text after pre-processing – Franklin D. Roosevelt's speech

We take a look at the word & character count after stop words removal: -

	Speech	char_count	word_count
Franklin D. Roosevelt	national day inauguration since 1789 people re...	4562	618
John F. Kennedy	vice president johnson speaker chief justice p...	4665	663
Richard Nixon	vice president speaker chief justice senator c...	5771	781

Table 2.5 Word count & character count after stop words removal

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

Year	President's Name	Sl. No.	Top words	Count
1941	Franklin D. Roosevelt	1	nation	11
		2	know	10
		3	spirit	9
1961	John F. Kennedy	1	world	8
		2	sides	8
		3	new	7
1973	Richard Nixon	1	peace	19
		2	world	16
		3	new	15

Table 2.6 Most frequent words of each President's speech – before stemming

Stemming:

Stemming means removing the words to their original stem for better analysis. Ex: - words like chopping, & chopped are now chop (original stem).

Applying stemming to words before generating word clouds, and use a lemmatizer to retain meaningful dictionary format. Check word count for any changes.

We use a lemmatizer called word net lemmatizer to stem the words. The lemmatizer is not extreme while stemming like other stemmers like porter or snow ball. It retains the words in a dictionary format so that it is more meaningful and it also considers the context before converting the word.

	Speech
Franklin D. Roosevelt	national day inauguration since 1789 people re...
John F. Kennedy	vice president johnson speaker chief justice p...
Richard Nixon	vice president speaker chief justice senator c...

Table 2.7 Dataset after stemming

The most frequent words for each president after stemming are: -

Year	President's Name	Sl. No.	Top words	Count
1941	Franklin D. Roosevelt	1	nation	15
		2	life	11
		3	know	10
1961	John F. Kennedy	1	world	8
		2	side	8
		3	power	7
1973	Richard Nixon	1	america	21
		2	peace	19
		3	world	18

Table 2.8 Most frequent words of each President's speech – post stemming

2.4 Plot the word cloud of each of the speeches of the variable.

Franklin D. Roosevelt:

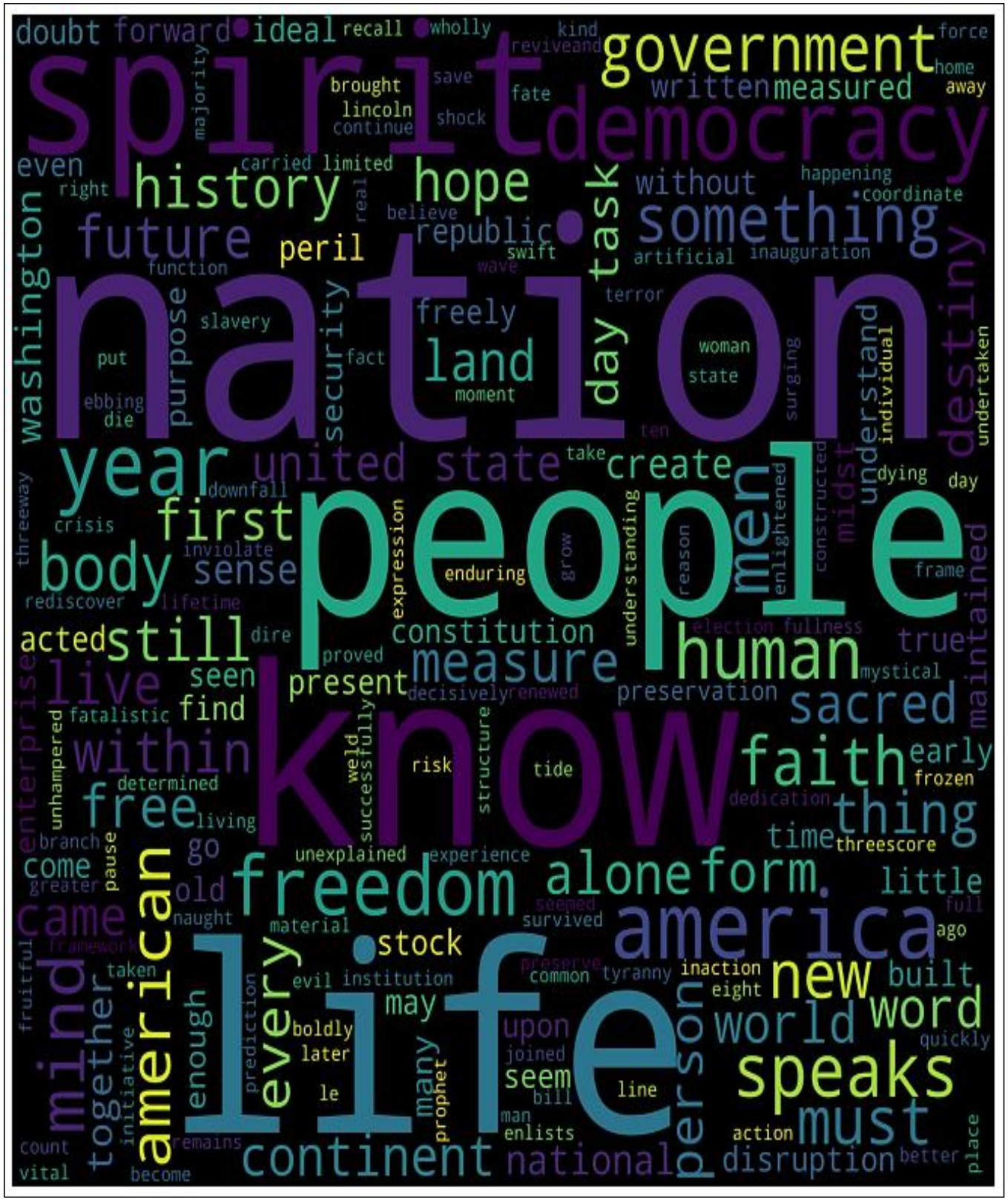

Fig 2.3 Word Cloud – Franklin D. Roosevelt's speech

John F. Kennedy:

Fig 2.4 Word Cloud – John F. Kennedy's speech

Richard Nixon:

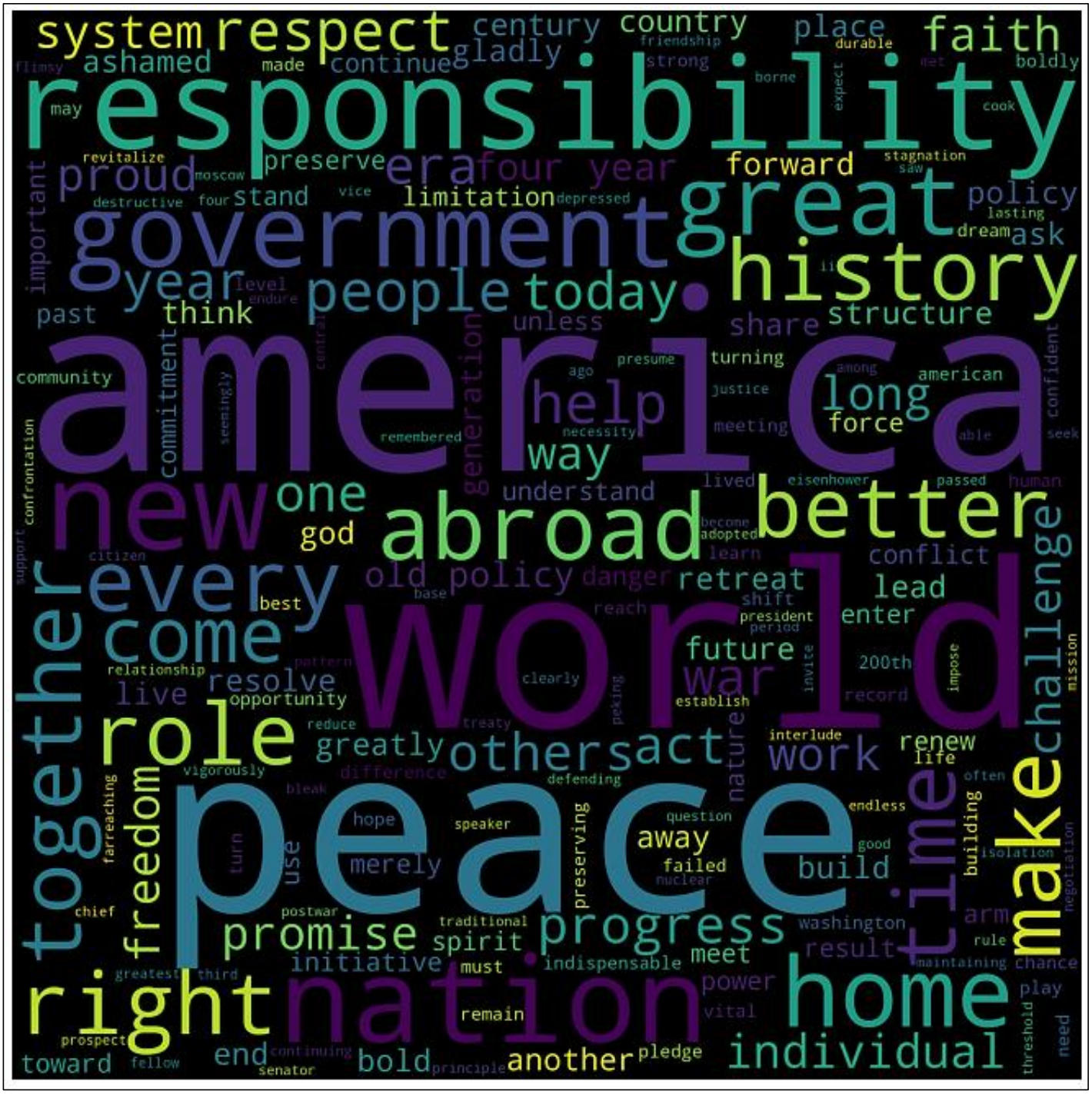

Fig 2.5 Word Cloud – Richard Nixon's speech