PGP - DSBA

Machine Learning

Project Report – November 2022

Shruti Jha
11-16-2022

Contents

List of Tables

List of Figures

Problem 1 – Exit Poll to Predict Winning Party

Introduction

A leading news channel, CNBE, wants to analyse recent elections. A survey was conducted on 1525 voters with 9 variables. A model needs to be built, 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 of survey results

vote	Party choice
age	in years
economic.cond.national	Assessment of current national economic conditions, 1 to 5
economic.cond.household	Assessment of current household economic conditions, 1 to 5
Blair	Assessment of the Labour leader, 1 to 5
Hague	Assessment of the Conservative leader, 1 to 5
Europe	an 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Eurosceptic' sentiment
political.knowledge	Knowledge of parties' positions on European integration, 0 to 3
gender	female or male

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

1.1.1 Sample of dataset

Here are the top 5 rows (sample) of the dataset:

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

Table 1. 1: Dataset Sample

- Dataset has 9 valid variables and as we can see the first column (Unnamed: 0) only contains serial numbers which are not relevant, we can remove it from our dataset.
- Also, changed the names of columns to add meaning to them. This is how the transformed data appears now:

	Vote	Age	Economic_cond_national	Economic_cond_household	Blair_assessment	Hague_assessment	Euroscepticism	Political_knowledge	Gender
0	Labour	43	3	3	4	1	2	2	female
1	Labour	36	4	4	4	4	5	2	male
2	Labour	35	4	4	5	2	3	2	male
3	Labour	24	4	2	2	1	4	0	female
4	Labour	41	2	2	1	1	6	2	male

Table 1. 2: Transformed Dataset Sample

1.1.2 Check for Duplicate Records

```
Number of duplicate records: 8
```

We have 8 duplicate entries in our database. As these duplicate records do not add any value to the study, be it associated with different people, we have removed the duplicates.

1.1.3 Types of variables in the dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1517 entries, 0 to 1516
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Vote                    1517 non-null   object
 1   Age                     1517 non-null   int64
 2   Economic_cond_national  1517 non-null   int64
 3   Economic_cond_household 1517 non-null   int64
 4   Blair_assessment        1517 non-null   int64
 5   Hague_assessment        1517 non-null   int64
 6   Euroscepticism          1517 non-null   int64
 7   Political_knowledge     1517 non-null   int64
 8   Gender                  1517 non-null   object
dtypes: int64(7), object(2)
memory usage: 106.8+ KB
```

- 'Vote' and 'Gender' variables are 'object' datatypes and rest all are integers (int64).
- There are a total of 1517 rows and 9 columns in the dataset (*after removal of duplicates*).

1.1.4 Missing values in the dataset

```
Vote                    0
Age                     0
Economic_cond_national  0
Economic_cond_household 0
Blair_assessment        0
Hague_assessment        0
Euroscepticism          0
Political_knowledge     0
Gender                  0
dtype: int64
```

From the above results we can say that there is no missing value present in the dataset.

1.1.5 Descriptive Statistics

Describe function provides a table indicating the count of variables, mean, standard deviation and other values for the 5-point summary that includes (min, 25%, 50%, 75% and max). 50% in the table is also known as median.

	count	mean	std	min	25%	50%	75%	max
Age	1517.0	54.241266	15.701741	24.0	41.0	53.0	67.0	93.0
Economic_cond_national	1517.0	3.245221	0.881792	1.0	3.0	3.0	4.0	5.0
Economic_cond_household	1517.0	3.137772	0.931069	1.0	3.0	3.0	4.0	5.0
Blair_assessment	1517.0	3.335531	1.174772	1.0	2.0	4.0	4.0	5.0
Hague_assessment	1517.0	2.749506	1.232479	1.0	2.0	2.0	4.0	5.0
Euroscepticism	1517.0	6.740277	3.299043	1.0	4.0	6.0	10.0	11.0
Political_knowledge	1517.0	1.540541	1.084417	0.0	0.0	2.0	2.0	3.0

Table 1. 3: Data Description for integer variables

For object/categorical columns, describe function shows the observation count, unique values in each column, most frequent value and value frequency in each column.

	count	unique	top	freq
Vote	1517	2	Labour	1057
Gender	1517	2	female	808

Table 1. 4: Data Description for object variables

From the above descriptive statistics, we can infer:

- The average age of voters from the survey is 54 years, ranging from 24 till 93 years.
- The fields - Economic_cond_national, Economic_cond_household, Blair_assessment, Hague_assessment, Euroscepticism and Political_knowledge are categorical in nature, but the ratings are represented in numeric form.

- Economic_cond_national, Economic_cond_household, Blair_assessment and Hague_assessment have 5 ratings, ranging from 1 to 5, where 1 being the lowest and 5 being the highest rating.
- Euroscepticism has 11 stages, ranging from 1 to 11, where high scores represent 'Eurosceptic' sentiment.
- Political_knowledge rating ranges from 0 to 3, where 0 being the lowest and 3 being the highest.
- Vote has 2 unique values – Labour and Conservative. Labour is the most popular party choice with 1057 voters opting for it.
- Gender has 2 unique values – Female and Male. Number of female voters are more, with 808 frequencies, as compared to men.

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

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

1.1.6 Skewness

	Skewness
Age	0.14
Economic_cond_national	-0.24
Economic_cond_household	-0.14
Blair_assessment	-0.54
Hague_assessment	0.15
Euroscepticism	-0.14
Political_knowledge	-0.42

Table 1. 5: Skewness

As per the rule of thumb, if the skewness is between -0.5 and 0.5, the data are fairly symmetrical. Which holds true for our dataset, the skewness ranges between the stipulated limit. Blair_assessment is slightly higher on the scale but very close to the higher limit of 0.5.

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

1.2.1 Check for outliers

To check for outliers, box plots have been plotted:

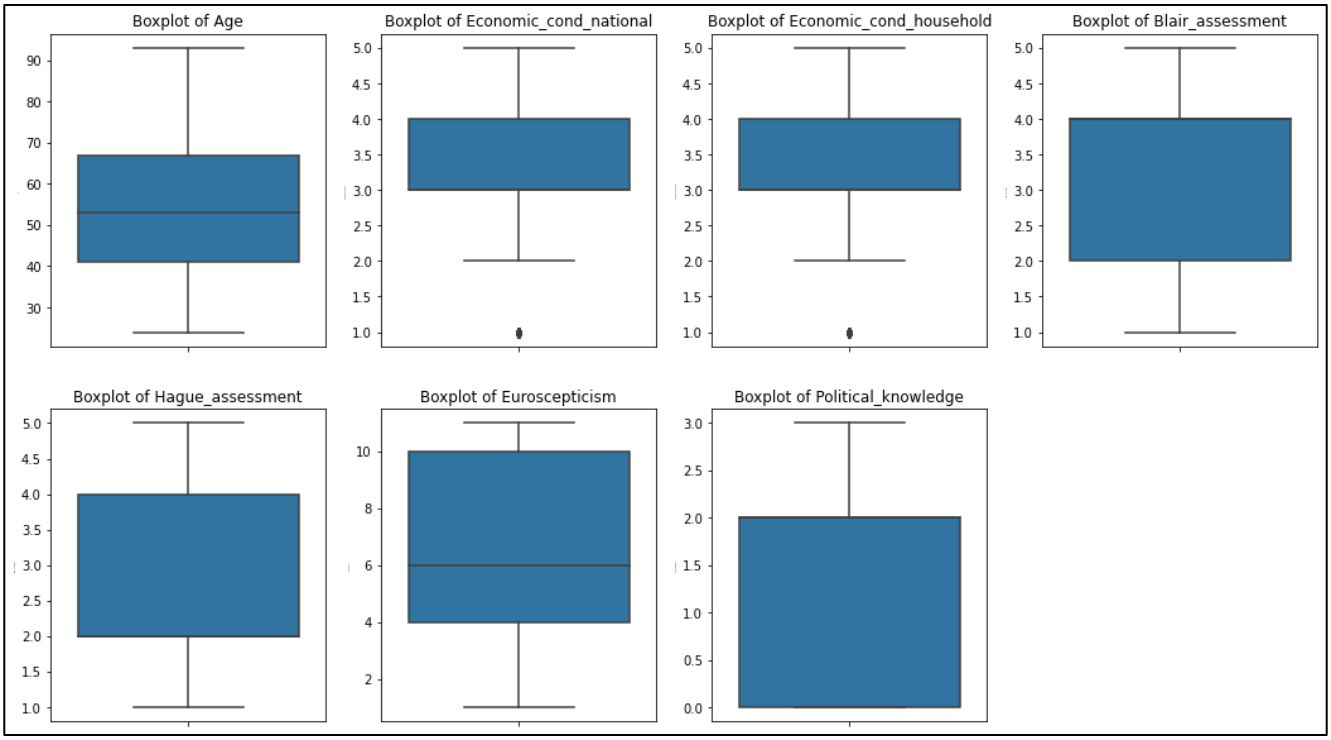
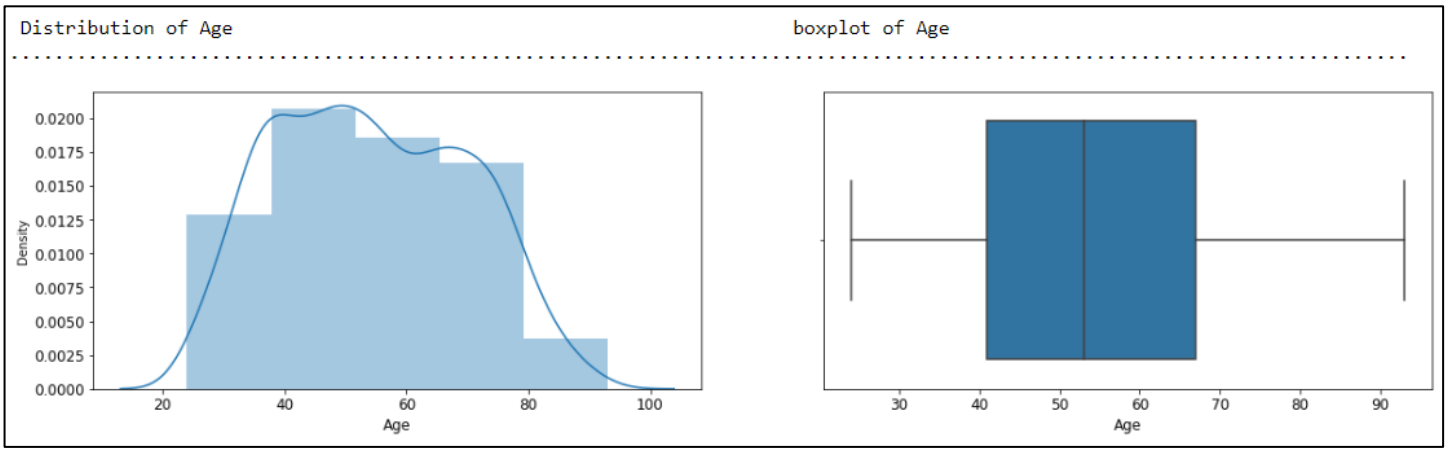


Figure 1. 1: Boxplot for Outliers

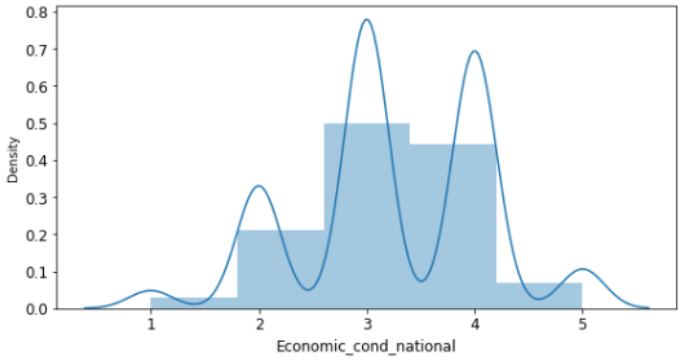
- The small dots outside the whiskers of boxplots denote outliers. As we can infer from the above plot, only 'Economic_cond_national' and 'Economic_cond_household' columns show to have outliers / extreme values present in them. But these 2 fields are on a scale of 1 to 5, hence the outliers are valid.

1.2.2 Univariate analysis

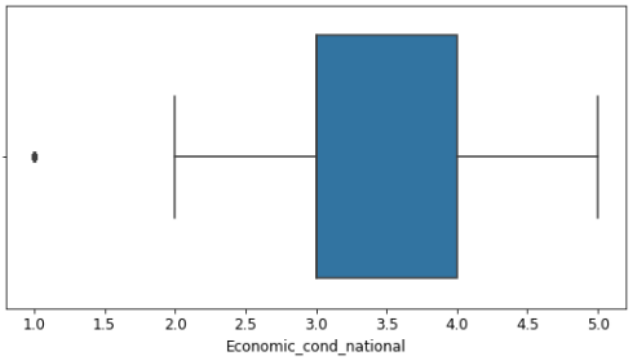
Univariate analysis is performed for all the numeric variables individually to visualize their distribution using distplot and to view 5-point summary using box plot.



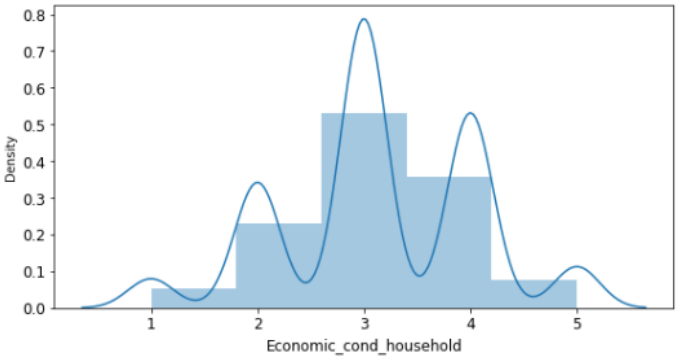
Distribution of Economic_cond_national



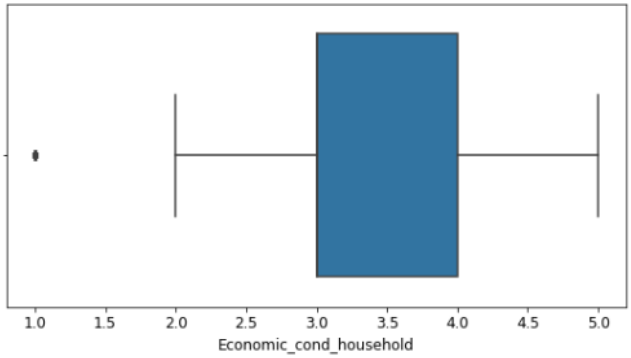
boxplot of Economic_cond_national



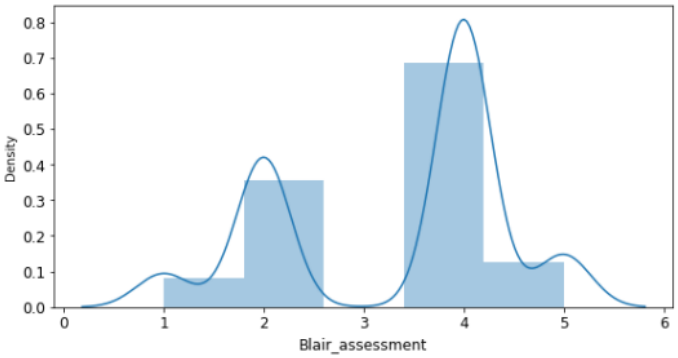
Distribution of Economic_cond_household



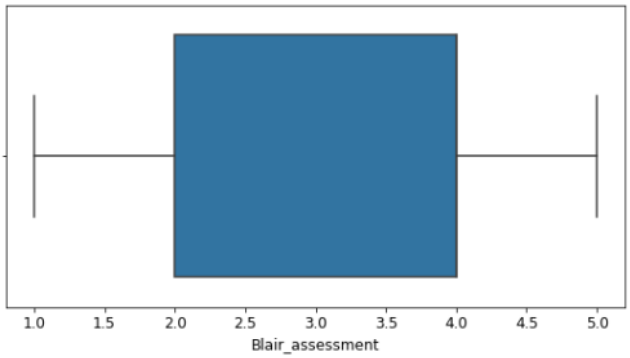
boxplot of Economic_cond_household



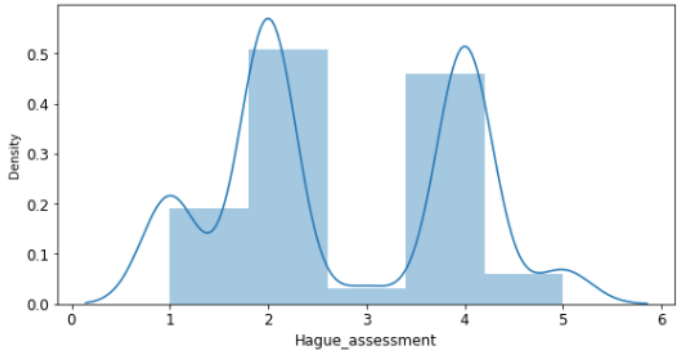
Distribution of Blair_assessment



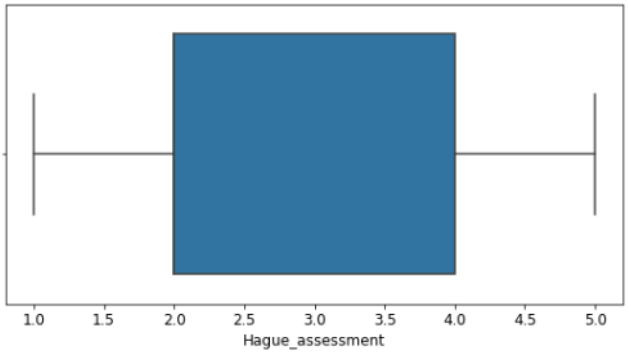
boxplot of Blair_assessment



Distribution of Hague_assessment



boxplot of Hague_assessment



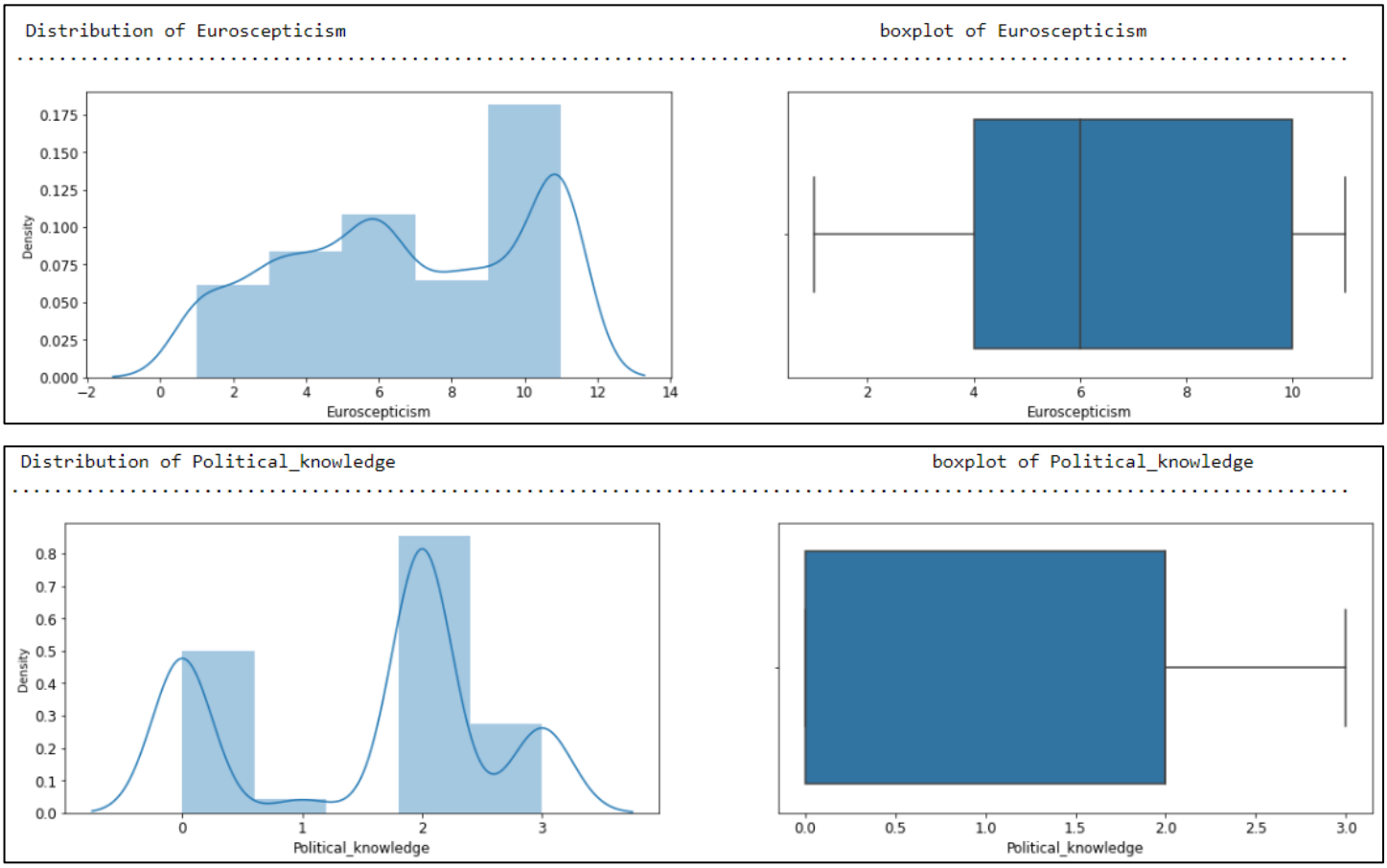


Figure 1. 2: Univariate analysis

Observations

- There are 7 numeric fields in the dataset.
- From the boxplots we can see that there are outliers present in 'Economic_cond_national' and 'Economic_cond_household', but as discussed earlier these outliers are valid.
- The distribution for all the variables is multi-modal.
- As we had seen skewness results, the skewness is very close to zero indicating data is symmetrical.
- Majority of the voters fall within the age of 40 to 68 approximately.
- The majority rating for 'Economic_cond_national' and 'Economic_cond_household' is given 3 and 4.
- The majority rating for 'Blair_assessment' and 'Hague_assessment' is given between 2 and 4.
- The majority rating for 'Euroscepticism' falls under 4 to 10, indicating the majority of the voters have moderate to high Eurosceptic sentiments, in other words withdrawalist sentiments.
- The 'Political_knowledge' of majority voters fall withing 0 to 2. And 25% voters have 0 political knowledge. 75% voters have political knowledge of rating 2. Very few voters have rating of 3.

1.2.3 Bivariate analysis

Bivariate analysis of continuous columns with target variable:

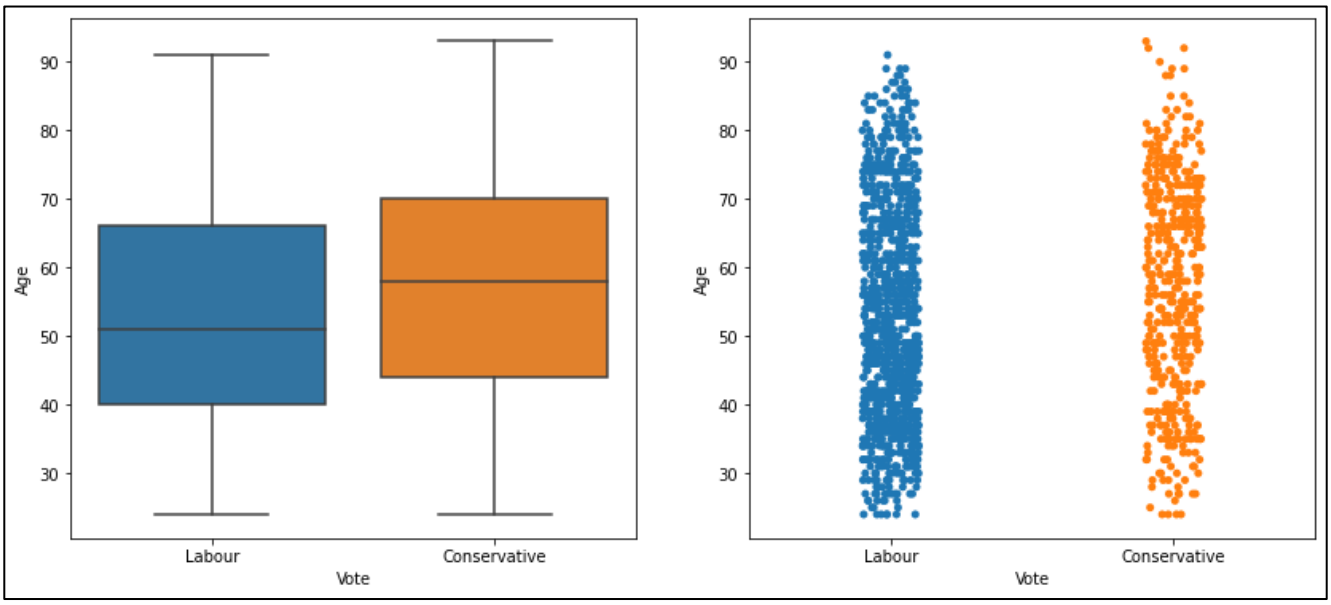
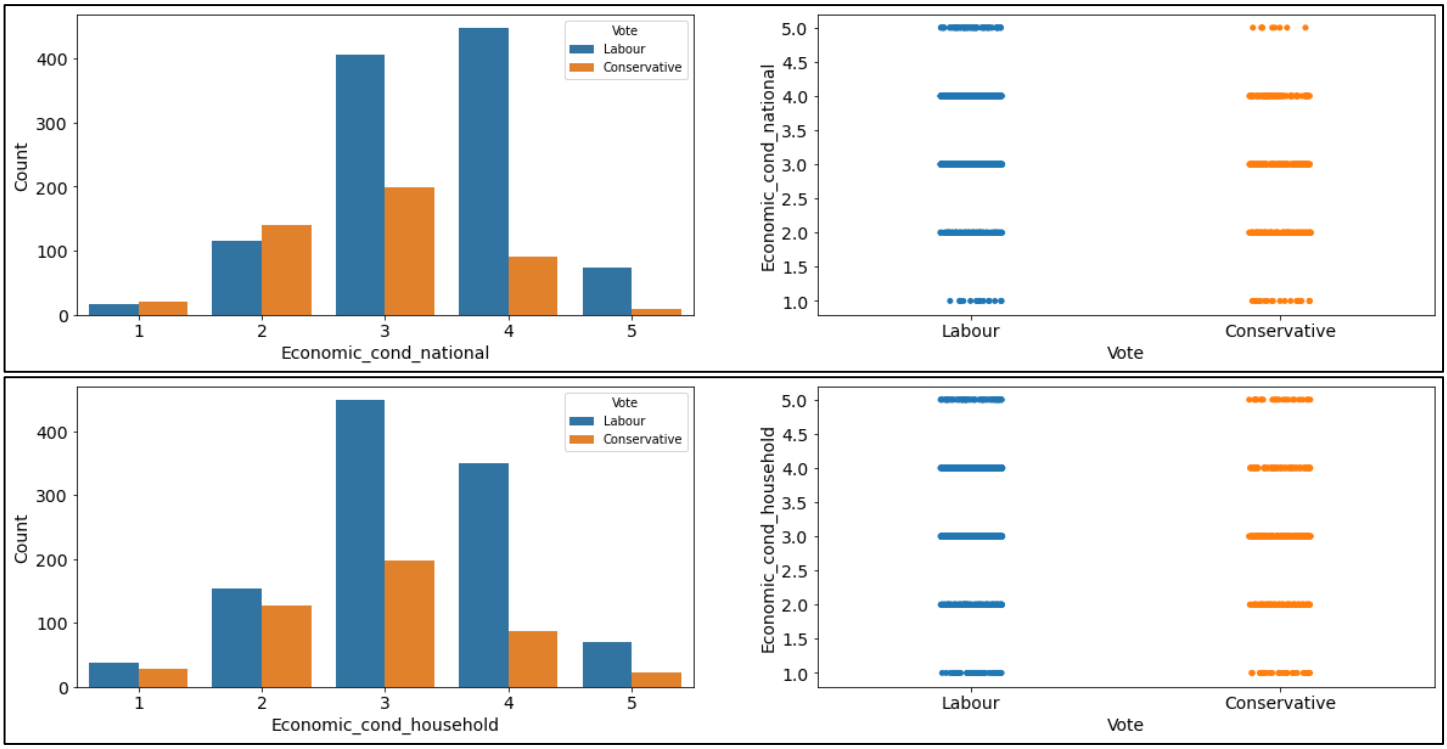


Figure 1. 3: Bivariate Analysis - Continuous

Observations

- The box plots show clear pattern that Labour party is the most popular choice among younger voters. Voters above the age of approximately 67 tend to mostly vote for Conservative party and the people below the age of 43 tend to mostly vote for Labour party.
- The strip plot of Labour party is denser than that of Conservative. This means majority of the voters chose Labour over Conservative party.

Bivariate analysis of categorical columns with target variable:



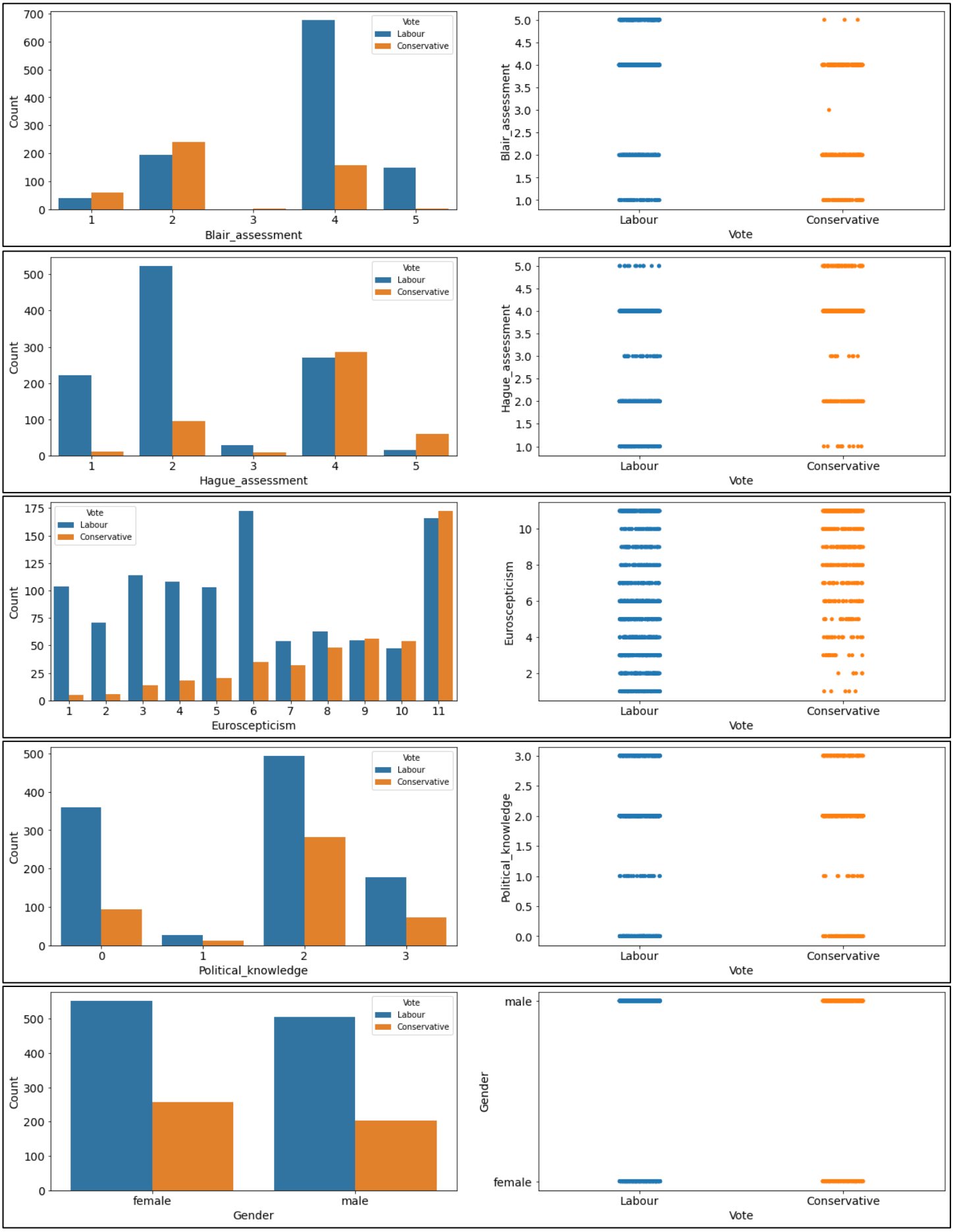


Figure 1. 4: Bivariate Analysis - Categorical

Observations

- 3 and 4 are the most popular rating given to Economic_cond_national among voters. Voters who give 5 rating to Economic_cond_national are likely to give less votes to Conservative party.
- 3 and 4 are also the most popular rating given to Economic_cond_household among voters.
- 4 is the most popular rating given to Blair_assessment by the voters. Very few voters have given 3 ratings. Voters who give 5 rating to Blair_assessment are likely to give more votes to Labour party.
- 2 is the most popular rating given to Hague_assessment by the voters. Voters who give 5 rating to Hague_assessment are likely to give more votes to Conservative party.
- Majority of the voters have moderate to high Eurosceptic sentiment. Voters with low Eurosceptic sentiments (Reformists) tend to vote more for Labour and less for Conservative party.
- Majority of the voters have Political_knowledge of 2 rating.
- In the survey, there are more of female voters as compared to male.

1.2.4 Multivariate analysis

Pair plot:

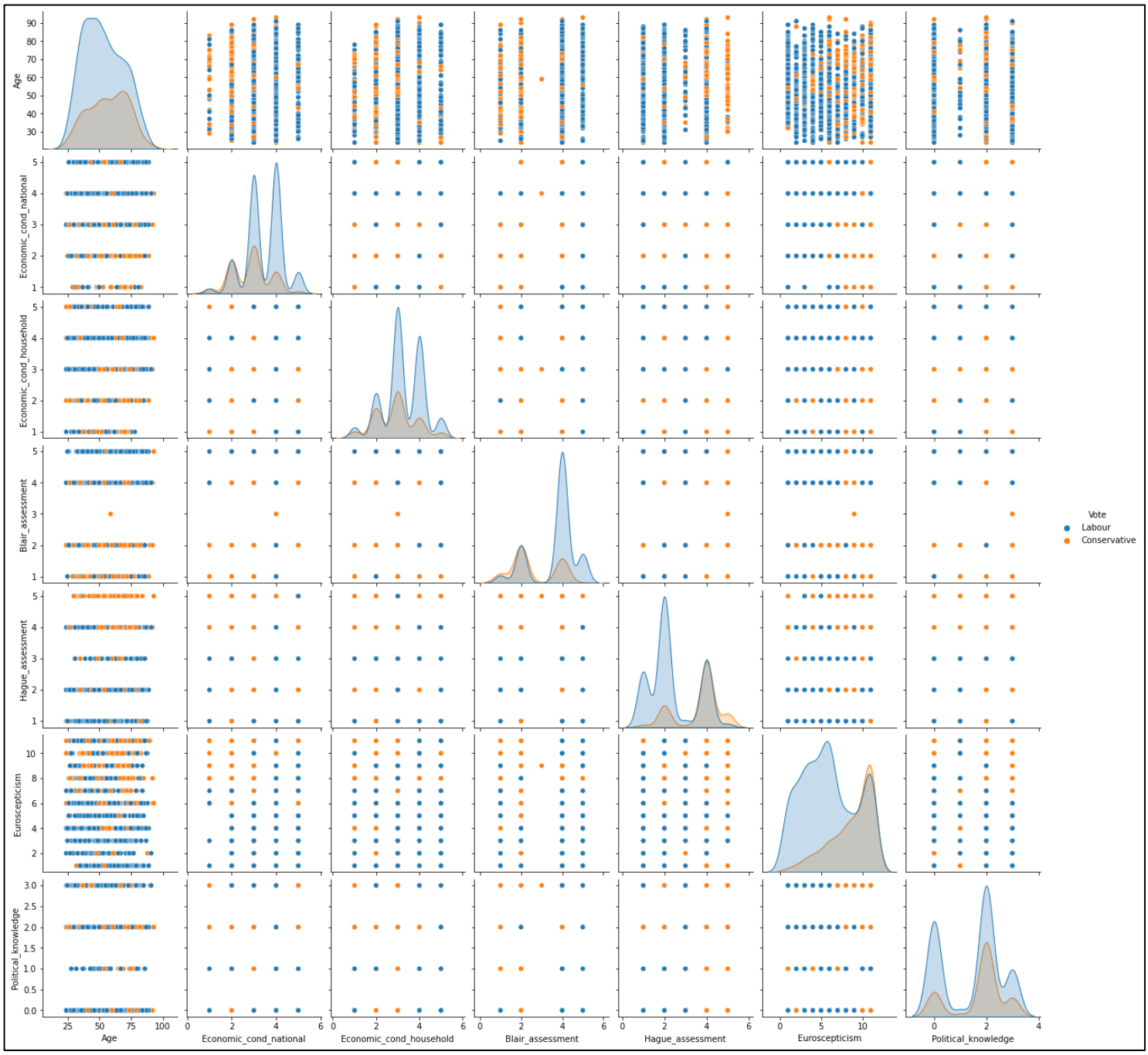


Figure 1. 5: Pair Plot

Observations

- Voters who give 5 rating to Economic_cond_national, Economic_cond_household and Blair_assessment are likely to give more votes to Labour party.
- Voters who give 5 rating to both Hauge_assessment and Economic_cond_national, tend to give vote to Labour party.
- Voters who give higher rating to Hague_assessment and Economic_cond_household, tend to cast vote for Conservative party.
- Voters who give higher rating to Blair_assessment and Economic_cond_household, tend to cast vote for Labour party.

- Voter who are high Eurosceptic (withdrawalist), rate Economic_cond_national and Hague_assessment high, tend to vote for Conservative party.
- On the other hand, voter who are high Eurosceptic (withdrawalist), rate Economic_cond_household and Blair_assessment high, tend to vote for Labour party.
- Voters with high political knowledge tend to vote for Labour party.

Correlation plot (Heatmap):

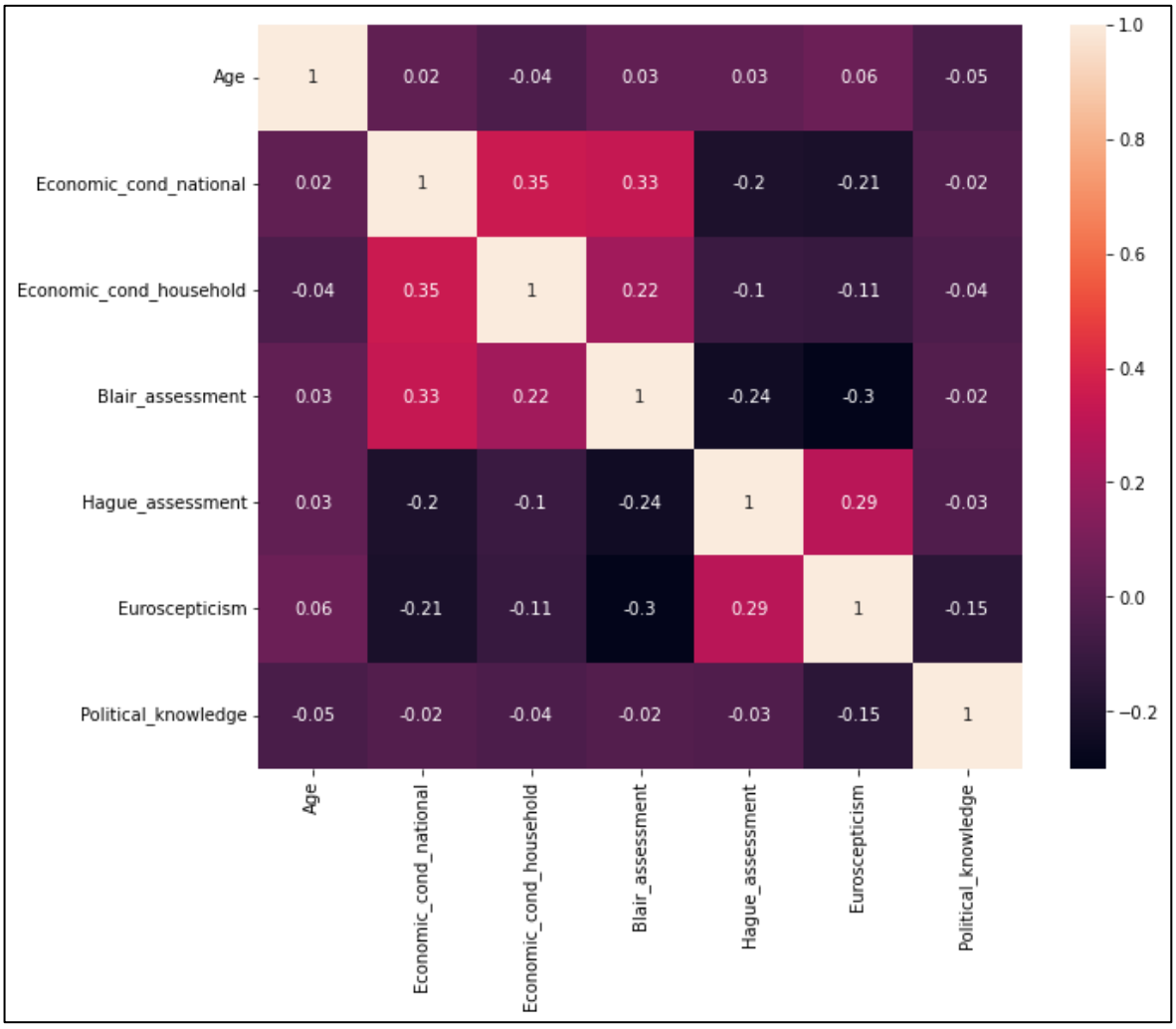


Figure 1. 6: Heatmap

Observations

- The overall correlation between all the variables are very less significant. But below are some observations bases on the even so slight correlation that exist:
- Economic_cond_national has slight positive correlation with Economic_cond_household and Blair_assessment, indicating voters who give high rating to Economic_cond_national, also tend to give high rating Economic_cond_household and Blair_assessment, and visa-versa.
- We can also observe slight negative correlation between Economic_cond_national and Euroscepticism, indicating that voters who give high rating to Economic_cond_national are generally less Eurosceptic.
- There is slight negative correlation between Blair_assessment and Hauge_assessment, indicating people who rate Blair high give less rating to Hague, and visa-versa.

- Blair_assessment is slight negatively correlated with Euroscepticism, indicating voters who are less Eurosceptic, tend to give higher rating to Labour party leader.
- Hague_assessment is slight positively correlated with Euroscepticism, indicating voters who are more Eurosceptic, tend to give higher rating to Conservative party leader.

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).

1.3.1 Data Encoding

For prediction models the data to pass should be in numeric/categorical codes format only. The variables with string values in our dataset need to be converted to integer format.

- Target variable 'Vote' has been converted into integer by replacing 'Labour' with 1 and 'Conservative' with 0, using LabelEncoder.

BEFORE	AFTER
Labour 1057 Conservative 460 Name: Vote, dtype: int64	0 460 1 1057 Name: Vote, dtype: int64

Table 1. 6: Data encoding 1

- 'Euroscepticism' column is in integer format but we can get better interpretation out of it if the data can be categorized as 'Reformist' – for 1 to 5 rating and 'Withdrawalist' – for 6 and above rating. Then performed one-hot encoding[1] (created dummy variables) to encode the data.

BEFORE	AFTER
Withdrawalist 954 Reformist 563 Name: Euroscepticism, dtype: int64	0 563 1 954 Name: Euroscepticism_Withdrawlist, dtype: int64

Table 1. 7: Data encoding 2

- Also performed one-hot encoding[2] on 'Gender' column as shown below:

BEFORE	AFTER
female 808 male 709 Name: Gender, dtype: int64	0 808 1 709 Name: Gender_male, dtype: int64

Table 1. 8: Data encoding 3

- Furthermore, for ordinal columns 'Economic_cond_national', Economic_cond_household', 'Blair_assessment', 'Hague_assessment' and 'Political_knowledge', they are already in in ordered sets and numeric format. Hence, no need to encode those.

ECONOMIC_COND_NATIONAL : 5 1 37 2 256 3 604 4 538 5 82 Name: Economic_cond_national, dtype: int64	ECONOMIC_COND_HOUSEHOLD : 5 1 65 2 280 3 645 4 435 5 92 Name: Economic_cond_household, dtype: int64

[1] Dummy variables are created using 'drop_first' parameter as True. As a result, we got n-1 columns. *(n = total number of unique values in a column)*

[2] Ibid

```
BLAIR_ASSESSMENT :  5
1     97
2    434
3      1
4    833
5    152
Name: Blair_assessment, dtype: int64
```

```
HAGUE_ASSESSMENT :  5
1    233
2    617
3     37
4    557
5     73
Name: Hague_assessment, dtype: int64
```

```
POLITICAL_KNOWLEDGE :  4
0    454
1     38
2    776
3    249
Name: Political_knowledge, dtype: int64
```

Table 1. 9: Data encoding 4

After encoding, this is how the dataset appear:

	Vote	Age	Economic_cond_national	Economic_cond_household	Blair_assessment	Hague_assessment	Political_knowledge	Euroscepticism_Withdrawalist	Gender_male
0	1	43	3	3	4	1	2	0	0
1	1	36	4	4	4	4	2	0	1
2	1	35	4	4	5	2	2	0	1
3	1	24	4	2	2	1	0	0	0
4	1	41	2	2	1	1	2	1	1

Table 1. 10: Modified Data Sample

Types of variables:

```
RangeIndex: 1517 entries, 0 to 1516
Data columns (total 9 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Vote                         1517 non-null   int32
 1   Age                          1517 non-null   int64
 2   Economic_cond_national       1517 non-null   int64
 3   Economic_cond_household      1517 non-null   int64
 4   Blair_assessment             1517 non-null   int64
 5   Hague_assessment             1517 non-null   int64
 6   Political_knowledge          1517 non-null   int64
 7   Euroscepticism_Withdrawalist 1517 non-null   uint8
 8   Gender_male                  1517 non-null   uint8
dtypes: int32(1), int64(6), uint8(2)
```

1.3.2 Scaling Necessity

- Scaling of the data is necessary when the variables of the dataset are of different scales, i.e. one variable is in thousands and other in only hundreds.
- Although, not all models are impacted by unscaled data, except distance-based models.
- For this problem statement, we have to apply K-Nearest Neighbours (KNN) model which is based on distance matrix. As such, we will perform scaling on the data for all the models.
- The data is scaled using MinMaxScaler to bring all the variable values down to fall between 0 and 1.
- The scaling will be performed after splitting the data into training and testing sets. MinMaxScaler should be fitted using training data and then apply the scaler on the testing data.

1.3.3 Data Split - Split the data into train and test (70:30)

The target variable in our encoded dataset is 'Vote', where 0 = Conservative and 1 = Labour. Here is the proportion of values in the target variable:

```
Proportion of 1 in target variable: 69.68 %
Proportion of 0 in target variable: 30.32 %
```

The proportion seems to be well balanced and good enough to move forward with models building.

The data has been first divided in to independent and dependent (target) variables, x and y respectively.

The data is now split into training and testing set with both sets having 70% and 30% of the data, respectively. Here is the proportion of target variable in both the sets:

```
Proportion of 1 in target variable in training set: 69.65 %
Proportion of 0 in target variable in training set: 30.35 %

Proportion of 1 in target variable in testing set: 69.74 %
Proportion of 0 in target variable in testing set: 30.26 %
```

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

1.4.1 Logistic Regression Model

In the first instance, we built the model using the default values of parameters. After observing performance of the model, we will decide the best parameters to best fit the model.

Confusion matrix and classification report:

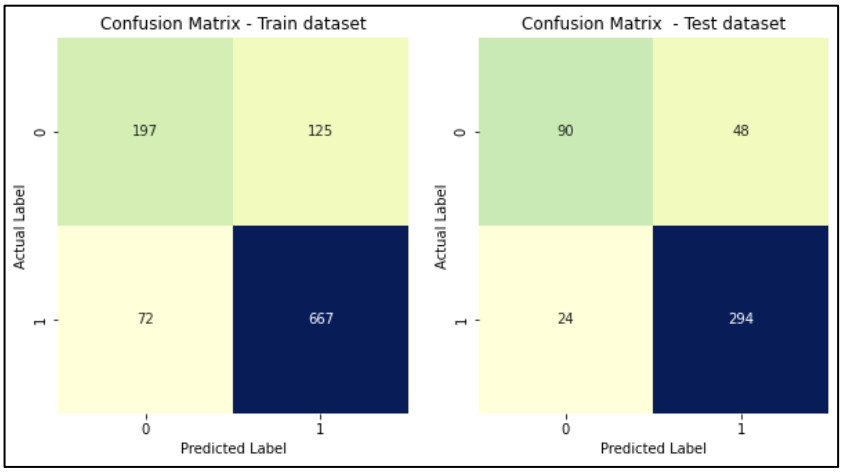


Figure 1. 7: LR Confusion Matrix 1

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.73	0.61	0.67	322
1	0.84	0.90	0.87	739
accuracy			0.81	1061
macro avg	0.79	0.76	0.77	1061
weighted avg	0.81	0.81	0.81	1061

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.79	0.65	0.71	138
1	0.86	0.92	0.89	318
accuracy			0.84	456
macro avg	0.82	0.79	0.80	456
weighted avg	0.84	0.84	0.84	456

Figure 1. 8: LR Classification Report 1

- As we can see the model is not over or under-fitted.
- Accuracy and F1 scores of train and test dataset are also pretty close, with the scores slightly higher for test dataset, indicating the model is well trained to make predictions. So, overall a good model.

	Training set	Test set
Accuracy	0.81	0.84
F1-score	0.87	0.89
Recall	0.90	0.92
Precision	0.84	0.86

Table 1. 11: LR Train & Test 1

- **Validness of the model**:
  1. Cross validation scores: After 10 folds cross validation, scores both on train and test data respectively for all 10 folds are almost same. Hence our model is valid.

```
Train data CV scores:
 [0.785  0.8491 0.8396 0.8019 0.8302 0.8113 0.783  0.7736 0.8302 0.7642]

Test data CV scores:
 [0.8043 0.913  0.8478 0.8696 0.8261 0.7391 0.7556 0.8889 0.9111 0.9111]
```

2. The error margin is low and the error rate in both train and test data is not too high. Thus, the model is not over-fitted or under-fitted.
3. This model is well trained as the accuracy of test dataset is slightly higher than the training dataset.

1.4.2 Linear Discriminant Analysis (LDA) Model

In the first instance, we built the model using the default values of parameters. After observing performance of the model, we will decide the best parameters to best fit the model.

Confusion matrix and classification report:

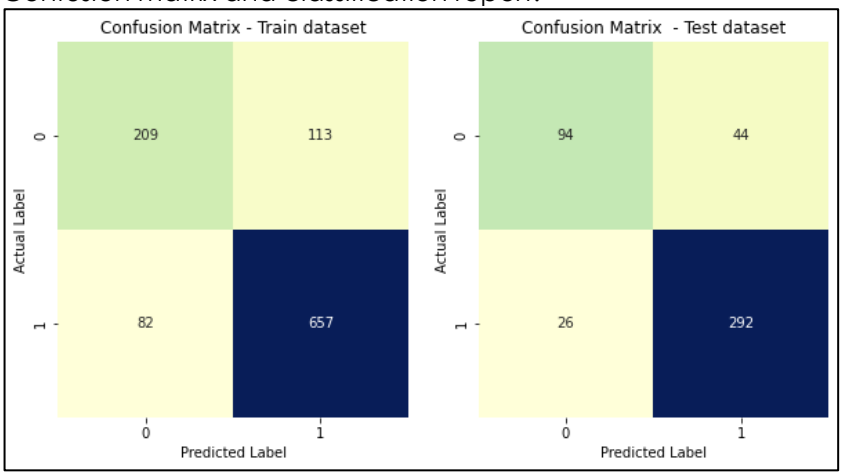


Figure 1. 9: LDA Confusion Matrix 1

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.72	0.65	0.68	322
1	0.85	0.89	0.87	739
accuracy			0.82	1061
macro avg	0.79	0.77	0.78	1061
weighted avg	0.81	0.82	0.81	1061

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.78	0.68	0.73	138
1	0.87	0.92	0.89	318
accuracy			0.85	456
macro avg	0.83	0.80	0.81	456
weighted avg	0.84	0.85	0.84	456

Figure 1. 10: LDA Classification Report 1

- As we can see the model is not over or under-fitted.
- Accuracy and F1 scores of train and test dataset are also pretty close, with the scores slightly higher for test dataset, indicating the model is well trained to make predictions. So, overall a good model.

	Training set	Test set
Accuracy	0.82	0.85
F1-score	0.87	0.89
Recall	0.89	0.92
Precision	0.85	0.87

Table 1. 12: LDA Train & Test 1

- **Validness of the model**:
    1. Cross validation scores: After 10 folds cross validation, scores both on train and test data respectively for all 10 folds are almost same. Hence our model is valid.

```
Train data CV scores:
 [0.8131 0.8491 0.8585 0.8113 0.8302 0.7925 0.7925 0.7642 0.8208 0.7642]

Test data CV scores:
 [0.8043 0.913  0.8478 0.8043 0.7826 0.7609 0.7333 0.8889 0.8667 0.8889]
```

2. The error margin is low and the error rate in both train and test data is not too high. Thus, the model is not over-fitted or under-fitted.
3. This model is well trained as the accuracy of test dataset is slightly higher than the training dataset.

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

1.5.1 K-Nearest Neighbours Model

In the first instance, we built the model using the default values of parameters. After observing performance of the model, we will decide the best parameters to best fit the model.

Confusion matrix and classification report:

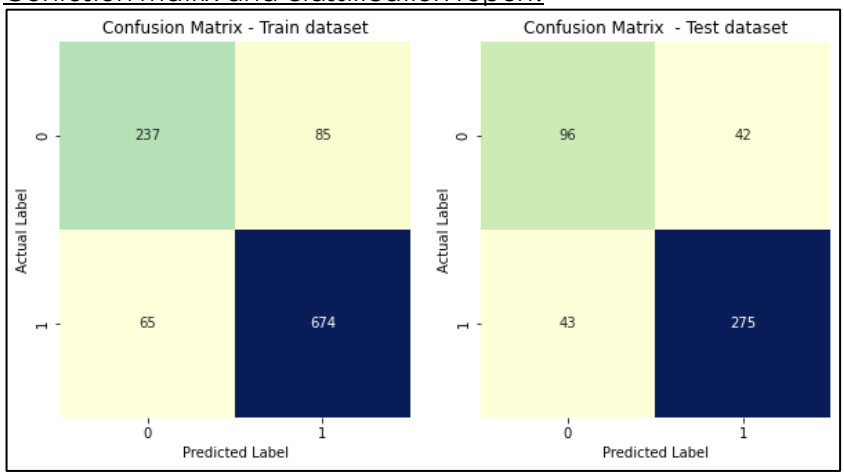


Figure 1. 11: KNN Confusion Matrix 1

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.78	0.74	0.76	322
1	0.89	0.91	0.90	739
accuracy			0.86	1061
macro avg	0.84	0.82	0.83	1061
weighted avg	0.86	0.86	0.86	1061

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.69	0.70	0.69	138
1	0.87	0.86	0.87	318
accuracy			0.81	456
macro avg	0.78	0.78	0.78	456
weighted avg	0.81	0.81	0.81	456

Figure 1. 12: KNN Classification Report 1

- As we can see the model is not over or under-fitted.
- Accuracy and F1 scores of train and test dataset are different, with test set not performing as good as the train, but the difference is not as much. We can observe any changes after model tuning.

	Training set	Test set
Accuracy	0.86	0.81
F1-score	0.90	0.87
Recall	0.91	0.86
Precision	0.89	0.87

Table 1. 13: KNN Train & Test 1

- **Validness of the model**:
  1. Cross validation scores: After 10 folds cross validation, scores both on train and test data respectively for all 10 folds are almost same. Hence our model is valid.

```
Train data CV scores:
 [0.7944 0.8491 0.8113 0.8208 0.8302 0.7358 0.7925 0.8396 0.8302 0.8396]

Test data CV scores:
 [0.7391 0.8913 0.8043 0.7391 0.8043 0.7391 0.7333 0.8889 0.8667 0.7556]
```

2. The error margin is low and the error rate in both train and test data is not too high. Thus, the model is not over-fitted or under-fitted.
3. The test set is not performing as good as the training set.

1.5.2 Naïve-Bayes Model

The model is built using the default values of parameters.

Confusion matrix and classification report:

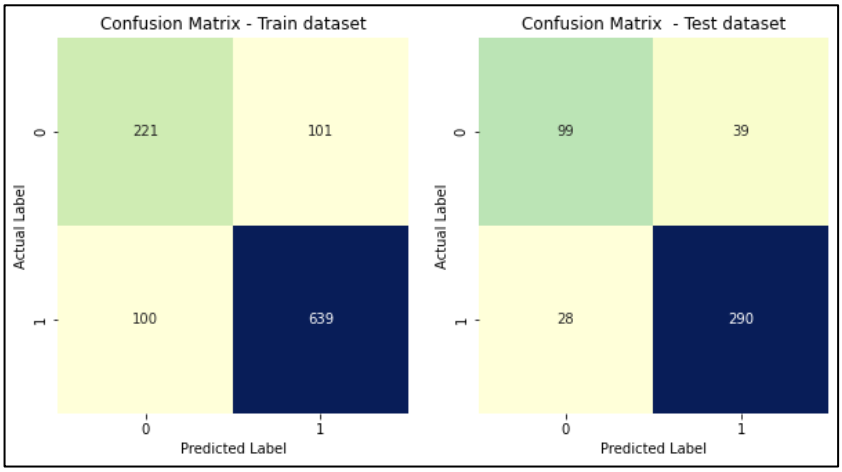


Figure 1. 13: NB Confusion Matrix

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.69	0.69	0.69	322
1	0.86	0.86	0.86	739
accuracy			0.81	1061
macro avg	0.78	0.78	0.78	1061
weighted avg	0.81	0.81	0.81	1061

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.78	0.72	0.75	138
1	0.88	0.91	0.90	318
accuracy			0.85	456
macro avg	0.83	0.81	0.82	456
weighted avg	0.85	0.85	0.85	456

Figure 1. 14: NB Classification Report

- As we can see the model is not over or under-fitted.
- Accuracy and F1 scores of train and test dataset are also more or less close, with the scores slightly higher for test dataset, indicating the model is well trained to make predictions.

	Training set	Test set
Accuracy	0.81	0.85
F1-score	0.86	0.90
Recall	0.86	0.91
Precision	0.86	0.88

Table 1. 14: NB Train & Test 1

- **Validness of the model**:
    1. Cross validation scores: After 10 folds cross validation, scores both on train and test data respectively for all 10 folds are almost same. Hence our model is valid.

```
Train data CV scores:
 [0.8131 0.8208 0.8019 0.8113 0.8302 0.7642 0.7925 0.7642 0.8585 0.8113]

Test data CV scores:
 [0.7826 0.8696 0.8043 0.7609 0.8261 0.7391 0.8222 0.9556 0.8667 0.8889]
```

2. The error margin is low and the error rate in both train and test data is not too high. Thus, the model is not over-fitted or under-fitted.
3. There is very less difference in the performance of train and test sets.

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

1.6.1 Logistic Regression Model Tuning

- We tuned the Logistic Regression model by changing its hyperparameters using GridSearch crossvalidation, to find out the best parameters to build a model that performs even better.
    - Best parameters: 'penalty': 'none'; 'solver': newton-cg; 'tol': 0.001; 'max_iter': 800
- We again built the model using best parameters, obtained from GridSearch crossvalidation:

Confusion matrix and classification report:

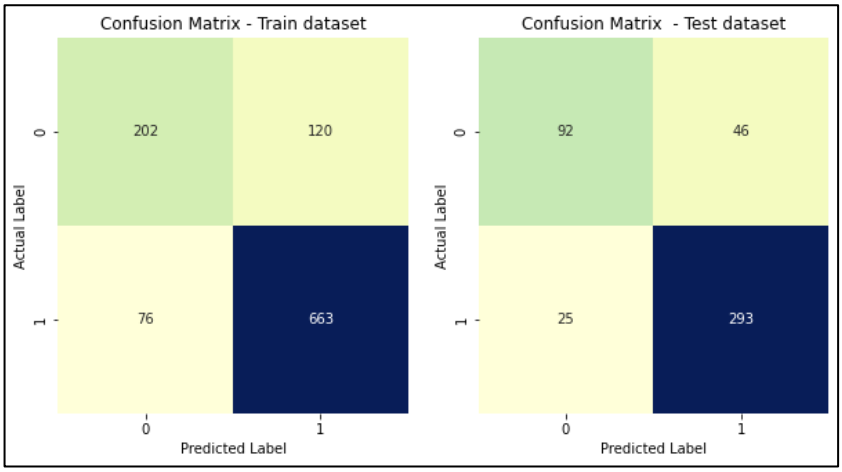


Figure 1. 15: LR Confusion Matrix 2

	precision	recall	f1-score	support
0	0.73	0.63	0.67	322
1	0.85	0.90	0.87	739
accuracy			0.82	1061
macro avg	0.79	0.76	0.77	1061
weighted avg	0.81	0.82	0.81	1061

	precision	recall	f1-score	support
0	0.79	0.67	0.72	138
1	0.86	0.92	0.89	318
accuracy			0.84	456
macro avg	0.83	0.79	0.81	456
weighted avg	0.84	0.84	0.84	456

Figure 1. 16: LR Classification Report 2

- We can observe very slight improvement in the highlighted results, as follows:

	Training set		Test set	
	Before Tuning	After Tuning	Before Tuning	After Tuning
Accuracy	0.81	0.82	0.84	0.84
F1-score	0.87	0.87	0.89	0.89
Recall	0.90	0.90	0.92	0.92
Precision	0.84	0.85	0.86	0.86

Table 1. 15: LR Train & Test 2

- The ROC-AUC score for test set is more than that of train set. Based on this observation, we can say that the testing sample is performing better than the training sample.

```
ROC - AUC score for training set is 0.87
ROC - AUC score for testing set is 0.91
```

- ROC Curve:

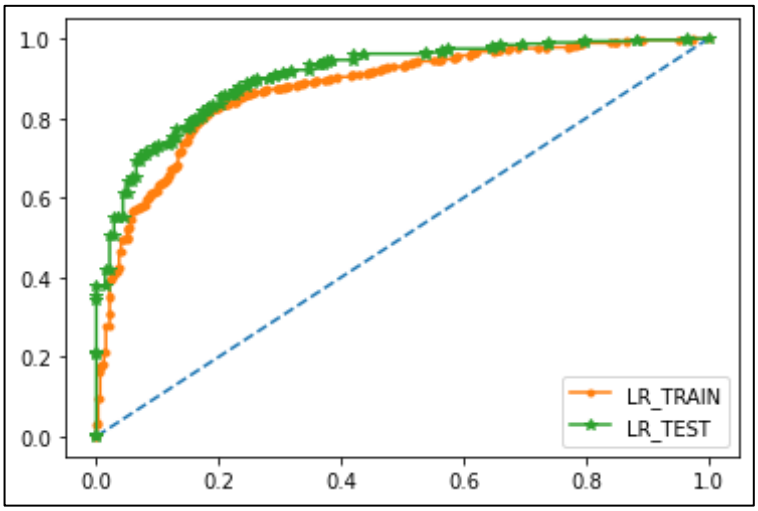


Figure 1. 17: LR ROC Curve

- The ROC curve is well fitted and test set is performing over and above the train set.

1.6.2 Linear Discriminant Analysis (LDA) Model Tuning

- Tuned the model by changing its hyperparameters using GridSearch crossvalidation, to find out the best parameters to build a model that performs even better.
    - Best parameters: 'solver': 'svd'; 'tol': 0.001
- Built the model using best parameters, obtained from GridSearch crossvalidation:

Confusion matrix and classification report:

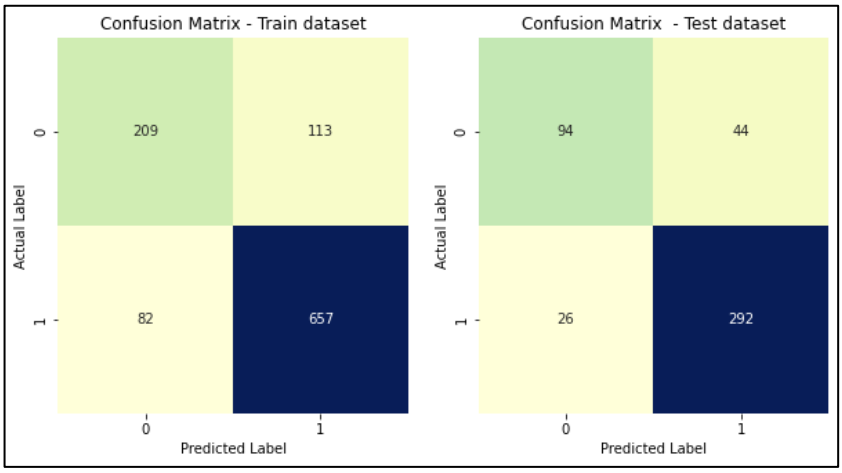


Figure 1. 18: LDA Confusion Matrix 2

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.72	0.65	0.68	322
1	0.85	0.89	0.87	739
accuracy			0.82	1061
macro avg	0.79	0.77	0.78	1061
weighted avg	0.81	0.82	0.81	1061

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.78	0.68	0.73	138
1	0.87	0.92	0.89	318
accuracy			0.85	456
macro avg	0.83	0.80	0.81	456
weighted avg	0.84	0.85	0.84	456

Figure 1. 19: LDA Classification Report 2

- We can observe no changes in the performance of the model after tuning hyperparameters, as shown below:

	Training set		Test set	
	Before Tuning	After Tuning	Before Tuning	After Tuning
Accuracy	0.82	0.82	0.85	0.85
F1-score	0.87	0.87	0.89	0.89
Recall	0.89	0.89	0.92	0.92
Precision	0.85	0.85	0.87	0.87

Table 1. 16: LDA Train & Test 2

Custom cut-off values to check for better performance:

- The default cut-off value for assigning probabilities is 0.5 (50%). In our problem statement, how it is interpreted is - People who have more than 50% probability of casting their vote for a particular party, they will actually vote for that particular party for sure.
- We assigned custom cut-off values and try to check for improvement in the performance of Linear Discriminant model. Below are the accuracy score, F1 score and confusion matrix at different cut-off values:

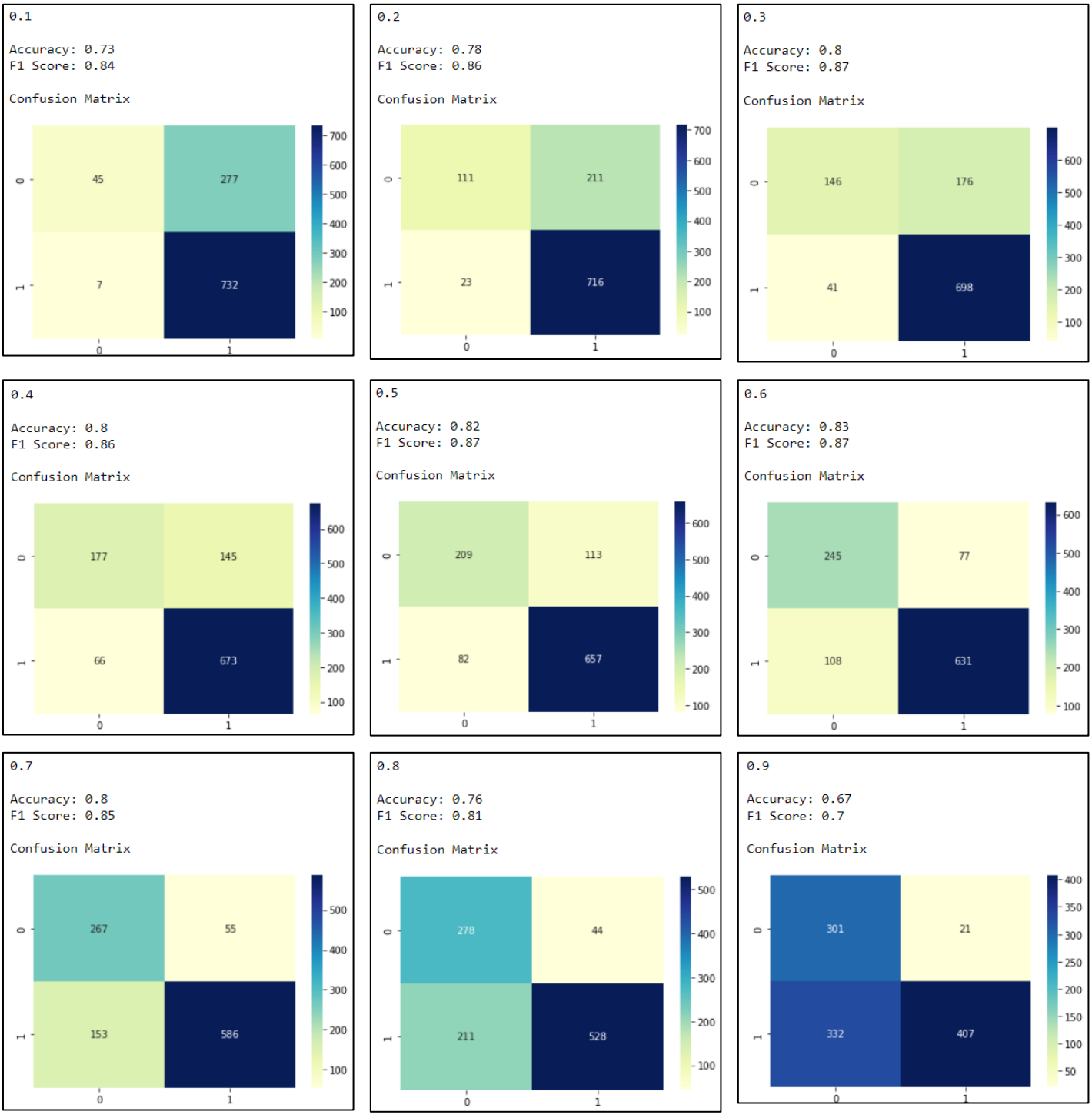


Figure 1. 20: Scores of Different Cut-off Values

- We see that 0.6 gives the best accuracy and F1 score than the rest of the custom cut-off values.

- Which indicates that voters who have more than 60% probability of casting their vote for Labour party, they will actually vote for Labour party for sure, and visa-versa.
- Hence, we can take the cut-off as 0.6 to get optimum performance of the model.
- Let's evaluate the model performance using 0.6 cut-off value:

Confusion matrix and classification report:

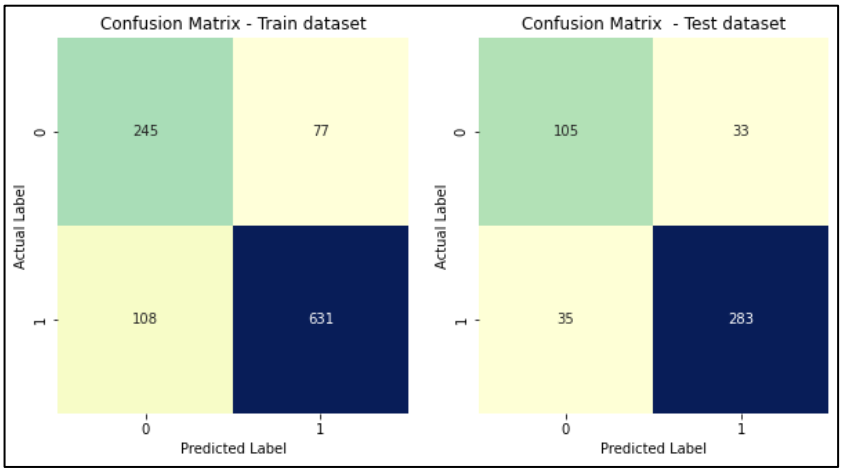


Figure 1. 21: LDA Confusion Matrix 3

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.69	0.76	0.73	322
1	0.89	0.85	0.87	739
accuracy			0.83	1061
macro avg	0.79	0.81	0.80	1061
weighted avg	0.83	0.83	0.83	1061

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.75	0.76	0.76	138
1	0.90	0.89	0.89	318
accuracy			0.85	456
macro avg	0.82	0.83	0.82	456
weighted avg	0.85	0.85	0.85	456

Figure 1. 22: LDA Classification Report 3

- We can observe very slight changes in the performance of the model. The accuracy has improved for the training set; precision has increased for both, train and test sets. Recall has decreased for both but since F1 score is unaffected, the model is well tuned.

	Training set		Test set	
	Before custom cut-off	After custom cut-off	Before custom cut-off	After custom cut-off
Accuracy	0.82	0.83	0.85	0.85
F1-score	0.87	0.87	0.89	0.89
Recall	0.89	0.85	0.92	0.89
Precision	0.85	0.89	0.87	0.90

Table 1. 17: LDA Train & Test 3

- The ROC-AUC score for test set is slightly higher than that of train set. Based on this observation, we can say that the testing sample is performing better than the training sample.

```
ROC - AUC score for training set is 0.87
ROC - AUC score for testing set is 0.91
```

- ROC Curve:

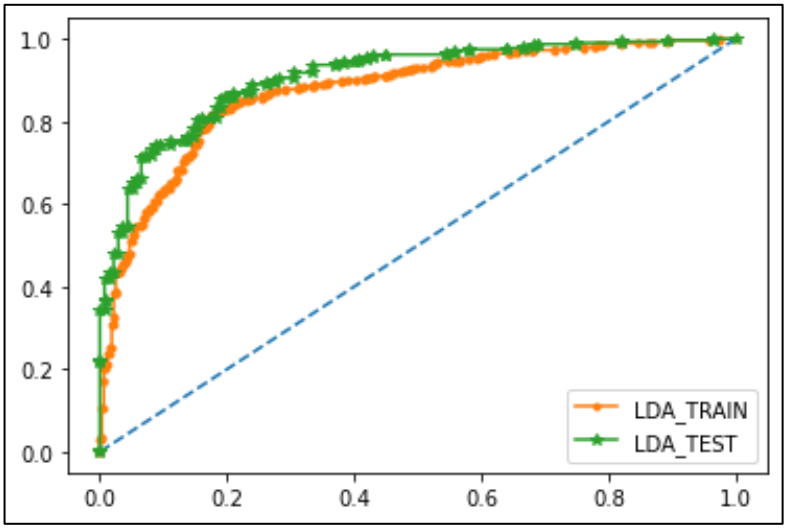


Figure 1. 23: LDA ROC Curve

- For Linear Discriminant Analysis model, the test set is performing better than the train set, hence the model is well trained.

1.6.3 K-Nearest Neighbours Model Tuning

- Ran the KNN with number of neighbours to be 1,3,5, till 23 to find the optimal number of neighbours from K=1,3,5,7....23 using the Misclassification error.
- Misclassification error (MCE) = 1 - Test accuracy score. Calculated MCE for each model with neighbours = 1,3,5...23 and find the model with lowest MCE.
- MCE Plotted:

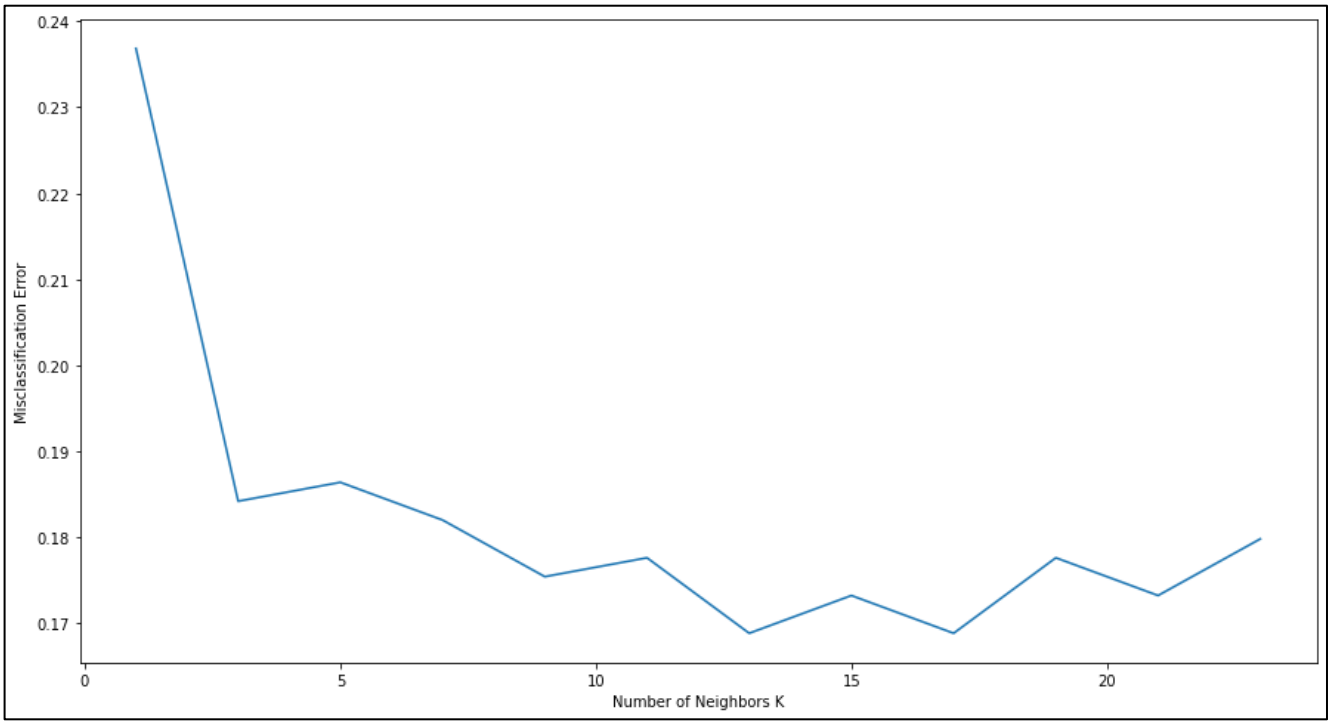


Figure 1. 24: KNN Misclassification error plot

- We can observe that the miss-classification error is the lowest (0.1689) at k = 13.
- Tuned the model with the best parameter as 'n_neighbors = 13'. This is how the tuned model performed:

Confusion matrix and classification report:

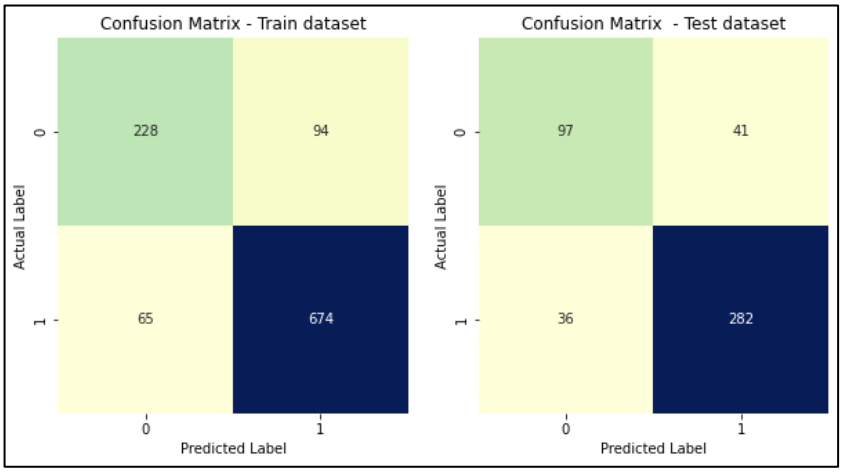


Figure 1. 25: KNN Confusion Matrix 2

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.78	0.71	0.74	322
1	0.88	0.91	0.89	739
accuracy			0.85	1061
macro avg	0.83	0.81	0.82	1061
weighted avg	0.85	0.85	0.85	1061

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.73	0.70	0.72	138
1	0.87	0.89	0.88	318
accuracy			0.83	456
macro avg	0.80	0.79	0.80	456
weighted avg	0.83	0.83	0.83	456

Figure 1. 26: KNN Classification Report 2

- We can observe the changes in the performance of the model after tuning. The performance of training set has dropped slightly. But the performance of test set has improved more. Hence, rather than the default of K as '5', we can see that '13' is the optimum number of neighbors through which our model is more consistent in terms of performance and recommendations.

	Training set		Test set	
	Before Tuning	After Tuning	Before Tuning	After Tuning
Accuracy	0.86	0.85	0.81	0.83
F1-score	0.90	0.89	0.87	0.88
Recall	0.91	0.91	0.86	0.89
Precision	0.89	0.88	0.87	0.87

Table 1. 18: KNN Train & Test 2

- The ROC-AUC score for both data sets are very close, indicating the model is well trained.

```
ROC - AUC score for training set is 0.9
ROC - AUC score for testing set is 0.88
```

- ROC Curve:

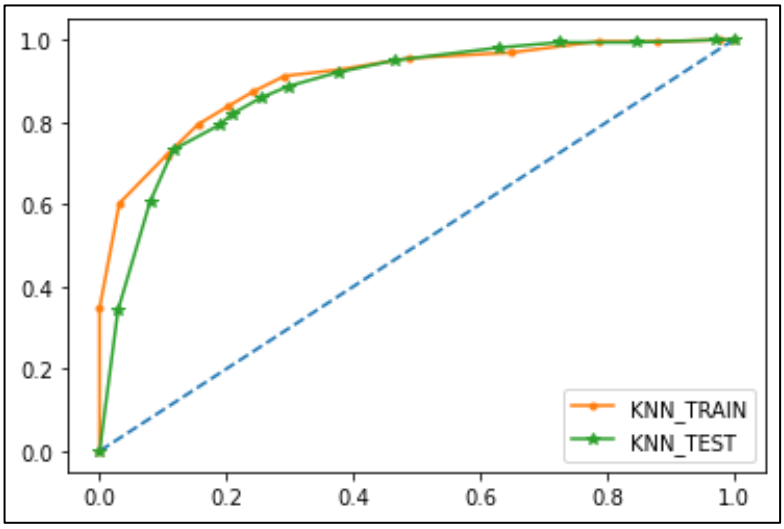


Figure 1. 27: KNN ROC Curve

- This is overall a well fitted model, where test set is performing pretty close as to train set.

1.6.4 Naïve-Bayes Model Tuning

There are no hyper-parameters to tune in Naïve-Bayes model. So, tuning cannot be performed on this model. The model with default parameters performed well and is consistent in terms of performance and recommendations.

	Training set	Test set
Accuracy	0.81	0.85
F1-score	0.86	0.90
Recall	0.86	0.91
Precision	0.86	0.88

Table 1. 19: NB Train & Test 2

- The ROC-AUC score for both data sets are more or less close, with test data having slightly higher score than that of train data. It can be inferred that the model is well trained and performing well to make predictions.

```
ROC - AUC score for training set is 0.87
ROC - AUC score for testing set is 0.9
```

- ROC Curve:

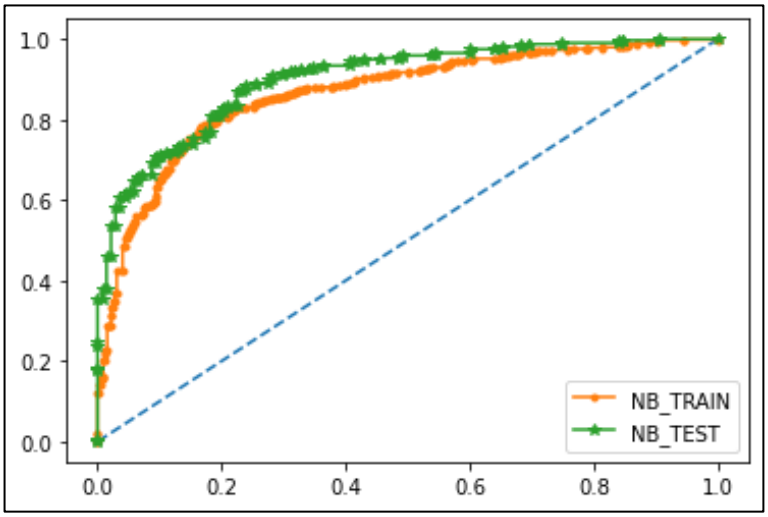


Figure 1. 28: NB ROC Curve

1.6.5 Bagging – Random Forest

In the first instance, we built the model using the default values of parameters and base_estimator as RandomForestClassifier(). After observing performance of the model, we will decide the best parameters to best fit the model.

Confusion matrix and classification report:

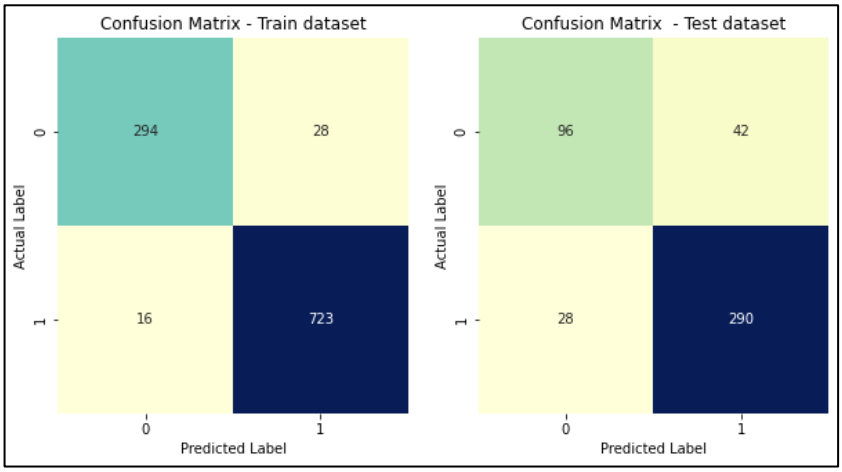


Figure 1. 29: BAG Confusion Matrix 1

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.95	0.91	0.93	322
1	0.96	0.98	0.97	739
accuracy			0.96	1061
macro avg	0.96	0.95	0.95	1061
weighted avg	0.96	0.96	0.96	1061

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.77	0.70	0.73	138
1	0.87	0.91	0.89	318
accuracy			0.85	456
macro avg	0.82	0.80	0.81	456
weighted avg	0.84	0.85	0.84	456

Figure 1. 30: BAG Classification Report 1

- As we can see the model seems to be a little over-fitted.
- There is a difference in the performance measures of both the sets, train set results being higher. This indicates that the test set is not performing as good as the train set.

	Training set	Test set
Accuracy	0.96	0.85
F1-score	0.97	0.89
Recall	0.98	0.91
Precision	0.96	0.87

Table 1. 20: Bagging Train & Test 1

- **Validness of the model**:
  1. Cross validation scores: After 10 folds cross validation, scores both on train and test data respectively for all 10 folds are almost same. Hence our model is valid.

```
Train data CV scores:
 [0.8131 0.8302 0.8585 0.8302 0.8774 0.783  0.7736 0.8302 0.8491 0.8019]

Test data CV scores:
 [0.8261 0.8478 0.8043 0.8696 0.8261 0.7391 0.7778 0.8889 0.8889 0.8889]
```

- We tuned the BaggingClassifier model by changing its hyperparameters using GridSearch crossvalidation, to find out the best parameters to build a model that performs even better.
    - Best parameters: 'n_estimators': 100; max_samples: 8; 'max_features': 4
- We again built the model using best parameters, obtained from GridSearch crossvalidation:

Confusion matrix and classification report:

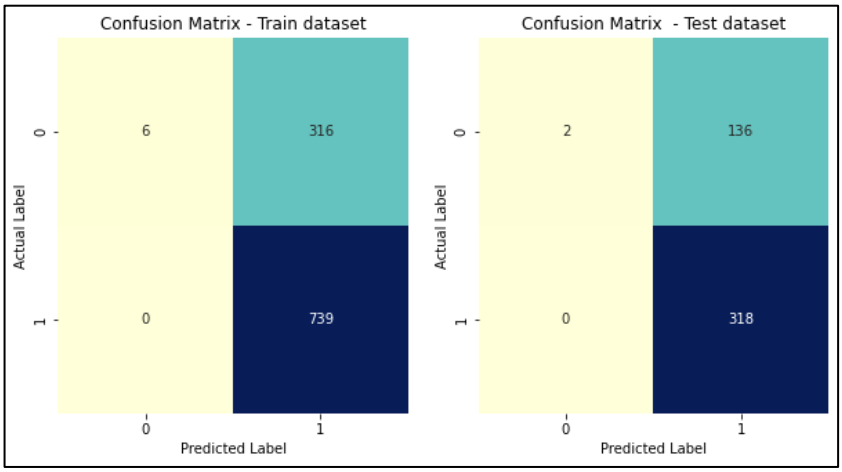


Figure 1. 31: BAG Confusion Matrix 2

	precision	recall	f1-score	support
0	1.00	0.02	0.04	322
1	0.70	1.00	0.82	739
accuracy			0.70	1061
macro avg	0.85	0.51	0.43	1061
weighted avg	0.79	0.70	0.58	1061

	precision	recall	f1-score	support
0	1.00	0.01	0.03	138
1	0.70	1.00	0.82	318
accuracy			0.70	456
macro avg	0.85	0.51	0.43	456
weighted avg	0.79	0.70	0.58	456

Figure 1. 32: BAG Classification Report 2

- We can observe after tuning the performance has dropped for both the sets. And model is not able to calculate True Negative and False Negative values properly, due to which model seems imbalanced.
- However, the performance scores for train and test sets are exactly same, as such we can say that the model in itself has been trained well, but doesn't seem optimum for making predictions properly.

	Training set		Test set	
	Before Tuning	After Tuning	Before Tuning	After Tuning
Accuracy	0.96	0.70	0.85	0.70
F1-score	0.97	0.82	0.89	0.82
Recall	0.98	1.00	0.91	1.00
Precision	0.96	0.70	0.87	0.70

Table 1. 21: Bagging Train & Test 2

- There is a difference in the ROC-AUC score for both sets, with test set performing better than train set.

```
ROC - AUC score for training set is 0.8636
ROC - AUC score for testing set is 0.9106
```

- ROC Curve:

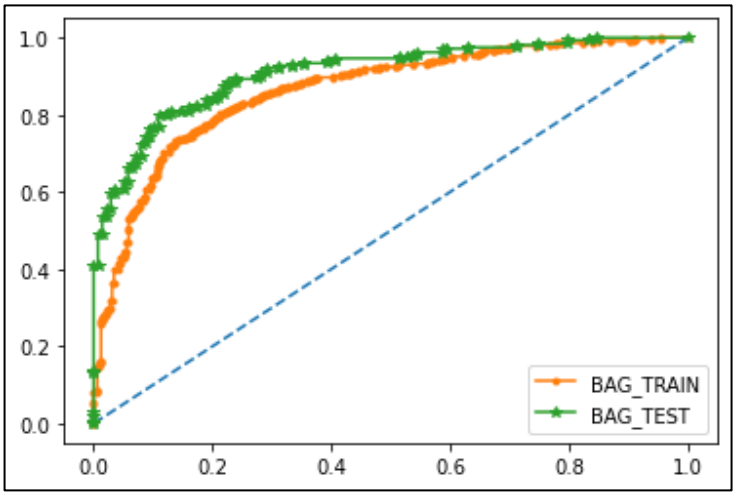


Figure 1. 33: BAG ROC Curve

- The test set is performing better than training set.

1.6.6 Adaptive Boosting

In the first instance, we built the model using the default values of parameters. After observing performance of the model, we will decide the best parameters to best fit the model.

Confusion matrix and classification report:

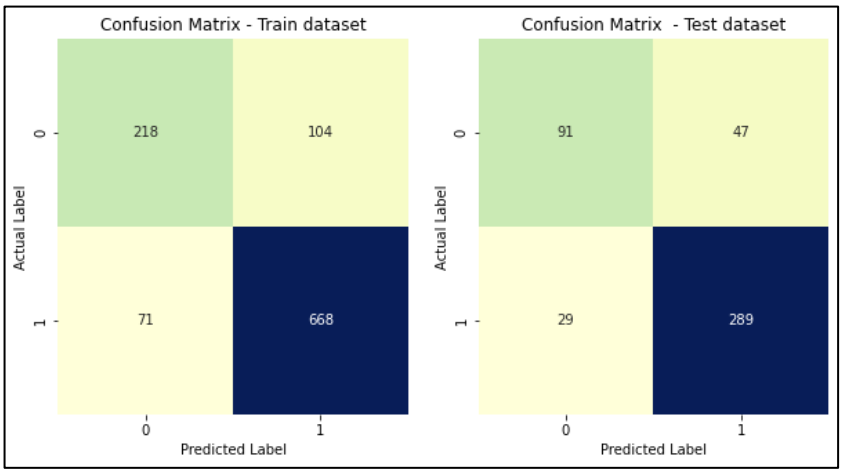


Figure 1. 34: ADB Confusion Matrix 1

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.75	0.68	0.71	322
1	0.87	0.90	0.88	739
accuracy			0.84	1061
macro avg	0.81	0.79	0.80	1061
weighted avg	0.83	0.84	0.83	1061

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.76	0.66	0.71	138
1	0.86	0.91	0.88	318
accuracy			0.83	456
macro avg	0.81	0.78	0.79	456
weighted avg	0.83	0.83	0.83	456

Figure 1. 35: ADB Classification Report 1

- As we can see the model is not over or under-fitted.
- Accuracy score of train and test dataset is very close and F1 scores is same for both. Indicating the model is well trained to make predictions.

	Training set	Test set
Accuracy	0.84	0.83
F1-score	0.88	0.88
Recall	0.90	0.91
Precision	0.87	0.86

Table 1. 22: Ada Boost Train & Test 1

- **Validness of the model**:
  1. Cross validation scores: After 10 folds cross validation, scores both on train and test data respectively for all 10 folds are almost same. Hence our model is valid.

```
Train data CV scores:
 [0.8131 0.8396 0.8113 0.8113 0.8396 0.7736 0.8208 0.7925 0.783  0.7642]

Test data CV scores:
 [0.8043 0.8478 0.8261 0.8696 0.7391 0.7609 0.7778 0.9111 0.7778 0.9111]
```

  2. The error margin is low and the error rate in both train and test data is not too high. Thus, the model is not over-fitted or under-fitted.

3. This model is well trained as the accuracy of test dataset is slightly higher than the training dataset.

- We tuned the Adaptive Boosting model by changing its hyperparameters using GridSearch crossvalidation, to find out the best parameters to build a model that performs even better.
    - Best parameters: 'n_estimators': 100
- We again built the model using best parameters, obtained from GridSearch crossvalidation:

Confusion matrix and classification report:

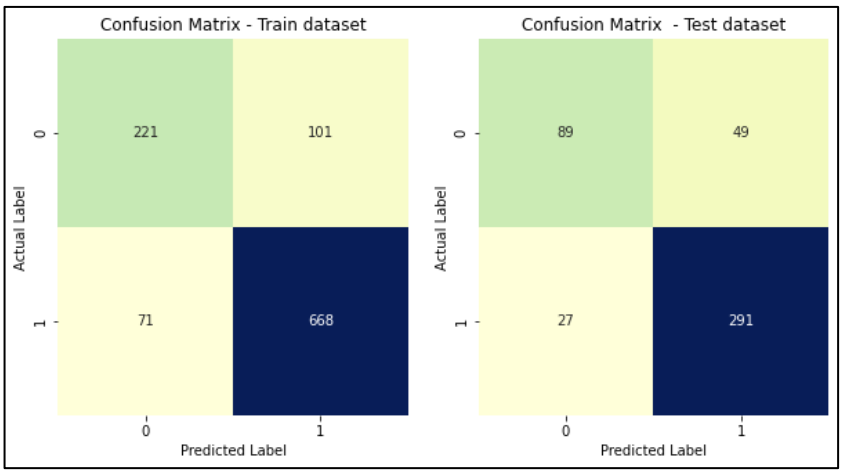


Figure 1. 36: ADB Confusion Matrix 2

	precision	recall	f1-score	support
0	0.76	0.69	0.72	322
1	0.87	0.90	0.89	739
accuracy			0.84	1061
macro avg	0.81	0.80	0.80	1061
weighted avg	0.83	0.84	0.84	1061

	precision	recall	f1-score	support
0	0.77	0.64	0.70	138
1	0.86	0.92	0.88	318
accuracy			0.83	456
macro avg	0.81	0.78	0.79	456
weighted avg	0.83	0.83	0.83	456

Figure 1. 37: ADB Classification Report 2

- We can observe very slight improvement in the F1 score of train set and Recall score of test set, as follows:

	Training set		Test set	
	Before Tuning	After Tuning	Before Tuning	After Tuning
Accuracy	0.84	0.84	0.83	0.83
F1-score	0.88	0.89	0.88	0.88
Recall	0.90	0.90	0.91	0.92
Precision	0.87	0.87	0.86	0.86

Table 1. 23: Ada Boost Train & Test 2

- The ROC-AUC scores for both sets are same. Based on this observation, we can say that the testing sample is performing as good as the training sample.

```
ROC - AUC score for training set is 0.8984
ROC - AUC score for testing set is 0.8987
```

- ROC Curve:

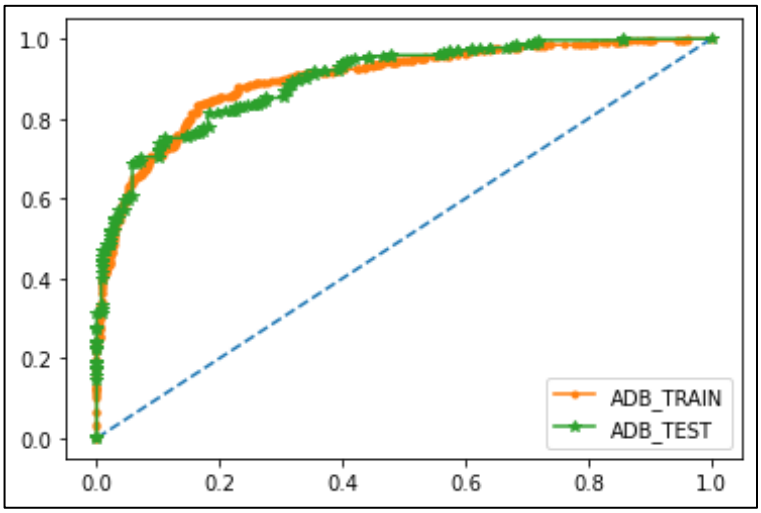


Figure 1. 38: ADB ROC Curve

- The train and test set are performing very similar. Model is well trained for prediction.

1.6.7 Gradient Boosting

In the first instance, we built the model using the default values of parameters. After observing performance of the model, we will decide the best parameters to best fit the model.

Confusion matrix and classification report:

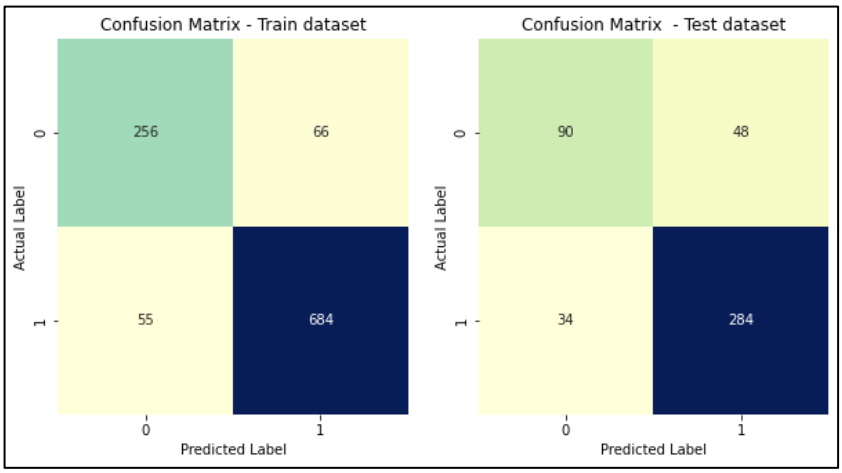


Figure 1. 39: GB Confusion Matrix 1

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.82	0.80	0.81	322
1	0.91	0.93	0.92	739
accuracy			0.89	1061
macro avg	0.87	0.86	0.86	1061
weighted avg	0.89	0.89	0.89	1061

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.73	0.65	0.69	138
1	0.86	0.89	0.87	318
accuracy			0.82	456
macro avg	0.79	0.77	0.78	456
weighted avg	0.82	0.82	0.82	456

Figure 1. 40: GB Classification Report 1

- As we can see the model seems to be over-fitted.
- There is a difference in the performance measures of both the sets, train set results being higher. This indicates that the test model is not performing as good as the train model.

	Training set	Test set
Accuracy	0.89	0.82
F1-score	0.92	0.87
Recall	0.93	0.89
Precision	0.91	0.86

Table 1. 24: Grad Boost Train & Test 1

- **Validness of the model**:
  1. Cross validation scores: After 10 folds cross validation, scores both on train and test data respectively for all 10 folds are almost same. Hence our model is valid.

```
Train data CV scores:
 [0.8131 0.8019 0.8491 0.8396 0.8679 0.8208 0.783  0.8019 0.7925 0.7925]

Test data CV scores:
 [0.8043 0.8478 0.7826 0.8043 0.8261 0.7826 0.7778 0.8889 0.8667 0.8444]
```

- We tuned the Gradient Boosting model by changing its hyperparameters using GridSearch crossvalidation, to find out the best parameters to build a model that performs even better.
    - Best parameters: 'n_estimators': 100; 'min_samples_split': 8; 'min_samples_leaf': 6; 'max_depth': 4; 'max_features': 4
- We again built the model using best parameters, obtained from GridSearch crossvalidation:

Confusion matrix and classification report:

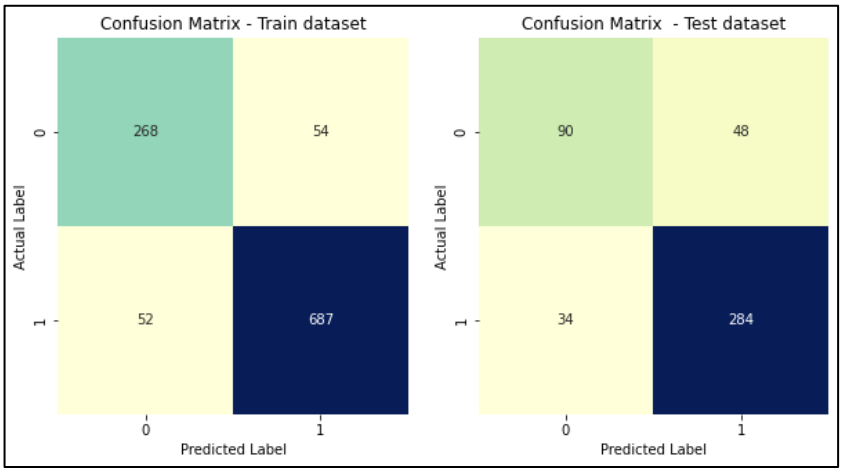


Figure 1. 41: GB Confusion Matrix 2

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.84	0.83	0.83	322
1	0.93	0.93	0.93	739
accuracy			0.90	1061
macro avg	0.88	0.88	0.88	1061
weighted avg	0.90	0.90	0.90	1061

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.73	0.65	0.69	138
1	0.86	0.89	0.87	318
accuracy			0.82	456
macro avg	0.79	0.77	0.78	456
weighted avg	0.82	0.82	0.82	456

Figure 1. 42: GB Classification Report 2

- We can observe very slight improvement in the train set results but test set results remain same:

	Training set		Test set	
	Before Tuning	After Tuning	Before Tuning	After Tuning
Accuracy	0.89	0.90	0.82	0.82
F1-score	0.92	0.93	0.87	0.87
Recall	0.93	0.93	0.89	0.89
Precision	0.91	0.93	0.86	0.86

Table 1. 25: Grad Boost Train & Test 2

- There is a significant difference in the ROC-AUC score for both sets. Based on this observation, we can say that even after tuning, the model remains over-fitted and testing sample is not performing as good as the training sample.

```
ROC - AUC score for training set is 0.9547
ROC - AUC score for testing set is 0.8843
```

- ROC Curve:

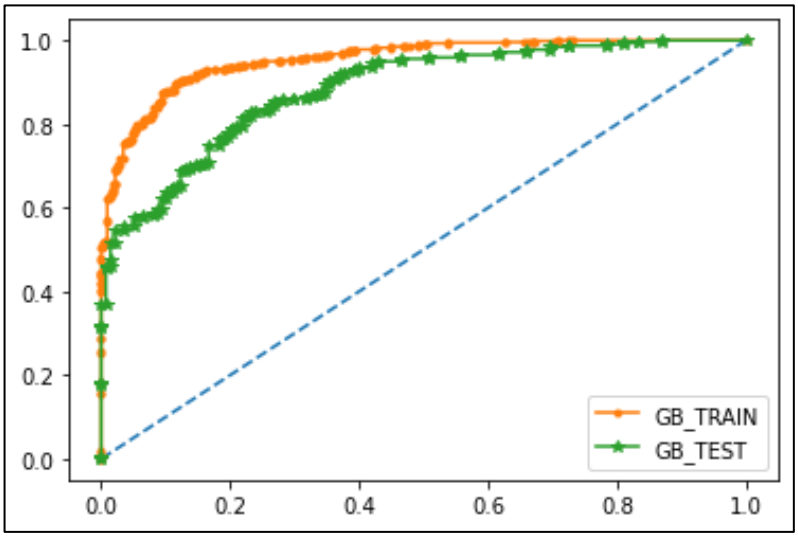


Figure 1. 43: GB ROC Curve

- The train data is performing way better than the test data; hence the model is over-fitted.

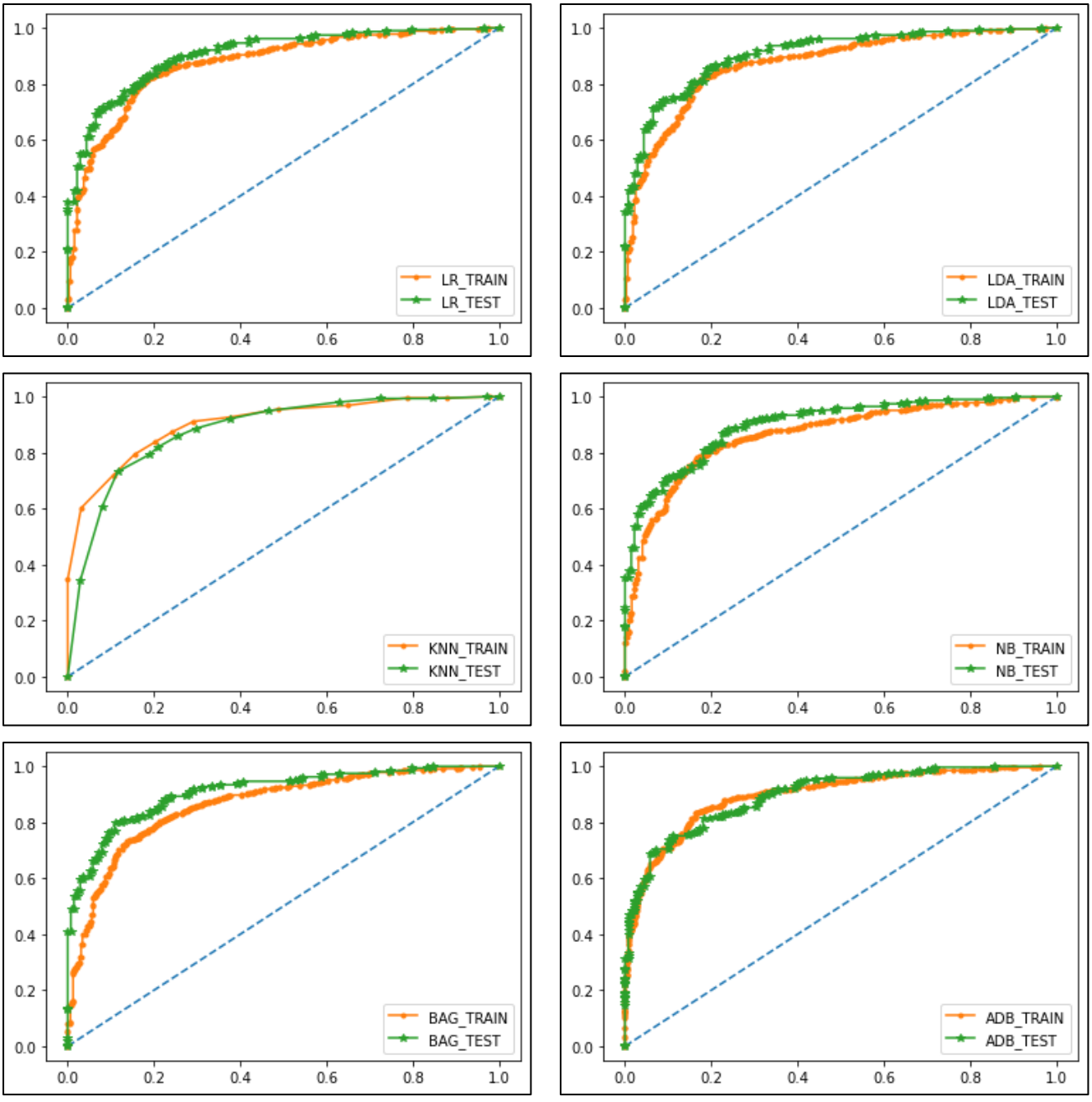
1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized.

Performance Metrics:

	LR Train	LR Test	LDA Train	LDA Test	KNN Train	KNN Test	NB Train	NB Test	Bag Train	Bag Test	ADB Train	ADB Test	GB Train	GB Test
Accuracy	0.82	0.84	0.83	0.85	0.85	0.83	0.81	0.85	0.70	0.70	0.84	0.83	0.90	0.82
F1 score	0.87	0.89	0.87	0.89	0.89	0.88	0.86	0.90	0.82	0.82	0.89	0.88	0.93	0.87
Recall	0.90	0.92	0.85	0.89	0.91	0.89	0.86	0.91	1.00	1.00	0.90	0.92	0.93	0.89
Precision	0.85	0.86	0.89	0.90	0.88	0.87	0.86	0.88	0.70	0.70	0.87	0.86	0.93	0.86
AUC	0.87	0.91	0.87	0.91	0.90	0.88	0.87	0.90	0.86	0.91	0.89	0.89	0.95	0.88

Table 1. 26: Performance Metrics

ROC Curve Comparision:



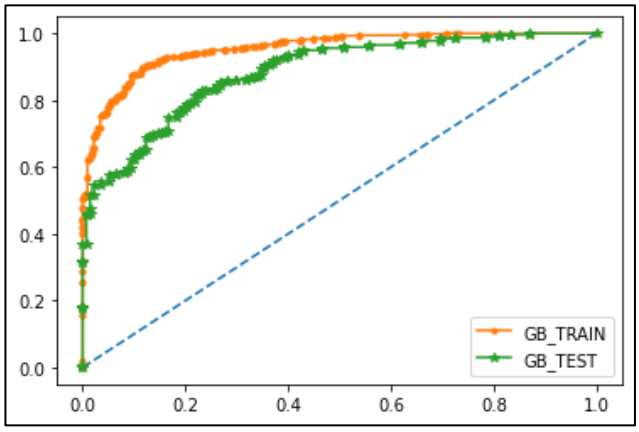


Figure 1. 44: ROC Curve Comparison

- Looking at the above table values, Gradient Boosing (GB) model has the highest scores for train dataset. But the scores for test datasets are comparatively less. The model is overfitted and isn't optimum for making predictions.
- The Bagging (Bag) model is not able to calculate True Negative and False Negative values properly, due to which model seems imbalanced. Moreover, the performance scores are less than that of other models we prepared. Hence, Bagging model also doesn't seem be fitted well enough for prediction purposes.
- From the remaining models, K-Nearest Neighbours (KNN) model has the highest accuracy and F1 score, combined. But the test set of that model is not performing as good as the train set, there is a slight difference in the scores of both the sets. Although, the model is optimized, but other models are better trained than KNN. As such, KNN is not a well optimized model among all.
- Rest of the models – Logistic Regression (LR), Linear Discriminant Analysis (LDA), Naïve-Bayse (NB) and Adaptive Boosting (AB) are well fitted and consistent in terms of performance and recommendations.
- The scenario of **False Negative, where "prediction is that voter will pick conservative party but actually voter picked labour party"** as well as **False Positive, where "prediction is that voter will pick labour party and actually picked conservative party"** need to be of main focus. As such, the accuracy and F1 score becomes of utmost importance for this case study.
- The model with highest accuracy and F1 score, along with test set performing better than train set is Linear Discriminant Analysis model.
- AUC score is also higher for test dataset for Linear Discriminant Model.
- The ROC curve seems to be best fit for the Linear Discriminant Analysis model, where testing set performing slightly better than training set.
- After evaluating all above factors, we can conclude that **Linear Discriminant Analysis model is the best optimized** to create an exit poll that will help in predicting overall win and seats covered by a particular party.

1.8 Based on these predictions, what are the insights?

- As per the LDA model accuracy score, there is a 83% chance that the model will accurately predict the election results.

Exit polls are conducted on random or systematic sampling of voters' sentiments. It happens with the help of media and helps in gauging the political trend of a particular constituency. From

- Majority of the voters have moderate to high Eurosceptic sentiments, in other words withdrawalist sentiments. Withdrawalist voters tend to vote more for Labour and less for Conservative party.
- Voters above the age of approximately 67 tend to mostly vote for Conservative party and the people below the age of 43 tend to mostly vote for Labour party.
- Voters who give high rating to Blair_assessment, are more likely to votes for Labour party.
- Voters who give high rating to Hague_assessment are more likely to votes for Conservative party.
- Voter who are withdrawalists and rate Hague_assessment high, tend to vote for Conservative party.
- On the other hand, voter who are withdrawalists, rate Economic_cond_household and Blair_assessment high, tend to vote for Labour party.
- Voters with high political knowledge tend to vote for Labour party.

Based on above observations, if the demography and sentiment of the voters remains same as the sample voters' data we have:

- More number of voters who give high rating to Blair_assessment and Economic_cond_household; as well as majority voters are withdrawalist.
- Based on these predictions the Labour party will win the upcoming elections.

Problem 2 – Text Mining

Introduction

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.

	Franklin D. Roosevelt	John F. Kennedy	Richard Nixon
Characters count	7571	7618	9991
Words count	1536	1546	2028
Sentences count	68	52	69

Table 2. 1: Character, Words, Sentence count

2.2 Remove all the stopwords from all three speeches.

- This is how the text looks before cleaning:

```
First 10 words from Franklin D. Roosevelt's speech:
 ['on', 'each', 'national', 'day', 'of', 'inauguration', 'since', '1789', ',', 'the']

First 10 words from John F. Kennedy's speech:
 ['vice', 'president', 'johnson', ',', 'mr', '.', 'speaker', ',', 'mr', '.']

First 10 words from Richard Nixon's speech:
 ['mr', '.', 'vice', 'president', ',', 'mr', '.', 'speaker', ',', 'mr']
```

- Stopwords are the filler words which are essential for sentence structure, but do not add any value on their own and would not be useful while assessing the sentiments of the text. Hence, it is a good practice to remove the stopwords from the text, so that focus can be on more sentimental words.
- 'stopwords' is a collection which has been built into 'nltk' (natural language tool kit) in python, which includes 179 words which do not convey any actual meaning.
- We also removed any punctuation marks present in the text, for the same reason, as they do not convey any meaning.
- We also performed stemming, to convert words to their root form, for example from 'spirited' to 'spirit'. So that all forms of the same words get counted as one and not different words.
- These are the steps followed to clean the text:
    1. Converted letters into lower case, as python is a case sensitive language and all the defined stopwords are in lower case.
    2. Removed stopwords and punctuation marks.
    3. Extended 'stopwords' list to also remove additional unimportant words from our text like '--', 'know', 'let', 'us', 'life', 'sides', 'new'.
    4. Performed stemming.
- This is how the text looks after cleaning:

```
First 10 words from Franklin D. Roosevelt's speech after cleaning:
 ['national', 'day', 'inauguration', 'since', '1789', 'people', 'renewed', 'sense', 'dedication', 'united']

First 10 words from John F. Kennedy's speech after cleaning:
 ['vice', 'president', 'johnson', 'mr', 'speaker', 'mr', 'chief', 'justice', 'president', 'eisenhower']

First 10 words from Richard Nixon's speech after cleaning:
 ['mr', 'vice', 'president', 'mr', 'speaker', 'mr', 'chief', 'justice', 'senator', 'cook']
```

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).

After removing stopwords & punctuations and stemming, here are the results:

- Top 3 words in Franklin D. Roosevelt's speech:

Before cleaning		After cleaning	
Words	**Count**	**Words**	**Count**
'the'	114	'nation'	12
'of'	81	'spirit'	9
','	77	'democracy'	9

Table 2. 2: Speech Cleaning - Roosevelt

- Top 3 words in John F. Kennedy's speech:

Before cleaning		After cleaning	
Words	**Count**	**Words**	**Count**
'the'	86	'world'	8
','	85	'pledge'	7
'of'	65	'citizens'	5

Table 2. 3: Speech Cleaning - Kennedy

- Top 3 words in Richard Nixon's speech:

Before cleaning		After cleaning	
Words	**Count**	**Words**	**Count**
','	96	'america'	21
'the'	83	'peace'	19
'.'	68	'world'	18

Table 2. 4: Speech Cleaning - Nixon

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

Word Cloud gives a graphical representation of all the important keywords present in the text after cleaning. The keywords in bigger font means that they are been used in the text more often than the ones in smaller fonts. It over all helps in grasping the sentiment of the text.

As it can be seen in the below word clouds from the inaugural speeches of 3 different US Presidents, some keywords like 'nation', 'america', 'spirit', 'democracy', 'pledge, 'citizen', 'peace', 'responsibility', 'world', revolve around their intention towards their nation, democracy, citizens and responsibilities to maintain peace in America as well as the world. So overall positive sentiments of these speeches.

- Word Cloud for Franklin D. Roosevelt's speech:

Figure 2. 1: Word Cloud - Roosevelt

- Word Cloud for John F. Kennedy's speech:

Figure 2. 2: Word Cloud - Kennedy

- Word Cloud for Richard Nixon's speech:

Figure 2. 3: Word Cloud - Nixon

End of Report