Machine Learning

PROJECT REPORT

By

HARI HARAN

9th April, 2023

CONTENTS	PAGE

Great Learning

PROBLEM 1 :

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

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

Performing EDA for the given data:

```
<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   float64
 2   economic.cond.national   1525 non-null   float64
 3   economic.cond.household  1525 non-null   float64
 4   Blair                    1525 non-null   float64
 5   Hague                    1525 non-null   float64
 6   Europe                   1525 non-null   float64
 7   political.knowledge      1525 non-null   float64
 8   gender                   1525 non-null   object
dtypes: float64(7), object(2)
memory usage: 107.4+ KB
```

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

- No Null values found

Table 1.1 - Data Info and Checking for null values.

Number of duplicate rows = 8

	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
67	Labour	35.0	4.0	4.0	5.0	2.0	3.0	2.0	male
626	Labour	39.0	3.0	4.0	4.0	2.0	5.0	2.0	male
870	Labour	38.0	2.0	4.0	2.0	2.0	4.0	3.0	male
983	Conservative	74.0	4.0	3.0	2.0	4.0	8.0	2.0	female
1154	Conservative	53.0	3.0	4.0	2.0	2.0	6.0	0.0	female
1236	Labour	36.0	3.0	3.0	2.0	2.0	6.0	2.0	female
1244	Labour	29.0	4.0	4.0	4.0	2.0	2.0	2.0	female
1438	Labour	40.0	4.0	3.0	4.0	2.0	2.0	2.0	male

Table 1.2 - Duplicate values

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

no. of rows: 1525

no. of columns: 10

- Unnamed column is unnecessary for further analysis. It can be dropped.

Table 1.3 -Skewness (right) and Shape (left)

	count	mean	std	min	25%	50%	75%	max
age	1525.0	54.0	16.0	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	3.0	1.0	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	3.0	1.0	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	3.0	1.0	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	3.0	1.0	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	7.0	3.0	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	2.0	1.0	0.0	0.0	2.0	2.0	3.0

Table 1.4 - Data Description

<u>Observations</u> -

- There are no null values in the data. Here we have dropped “Unnamed: 0 “ column.
- There are 7 float types and 2 object types variables.
- The skewness in the data is almost symmetrical. Only “Blair” has slightly higher values.
- Age -Average voters have age of 53. Minimum age is 24 and maximum age is 93.
- Most voters are from Europe.
- Political knowledge of most voters is very low.

More data can be collected to get understand the voting patterns and get more information.

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

Checking Outliers

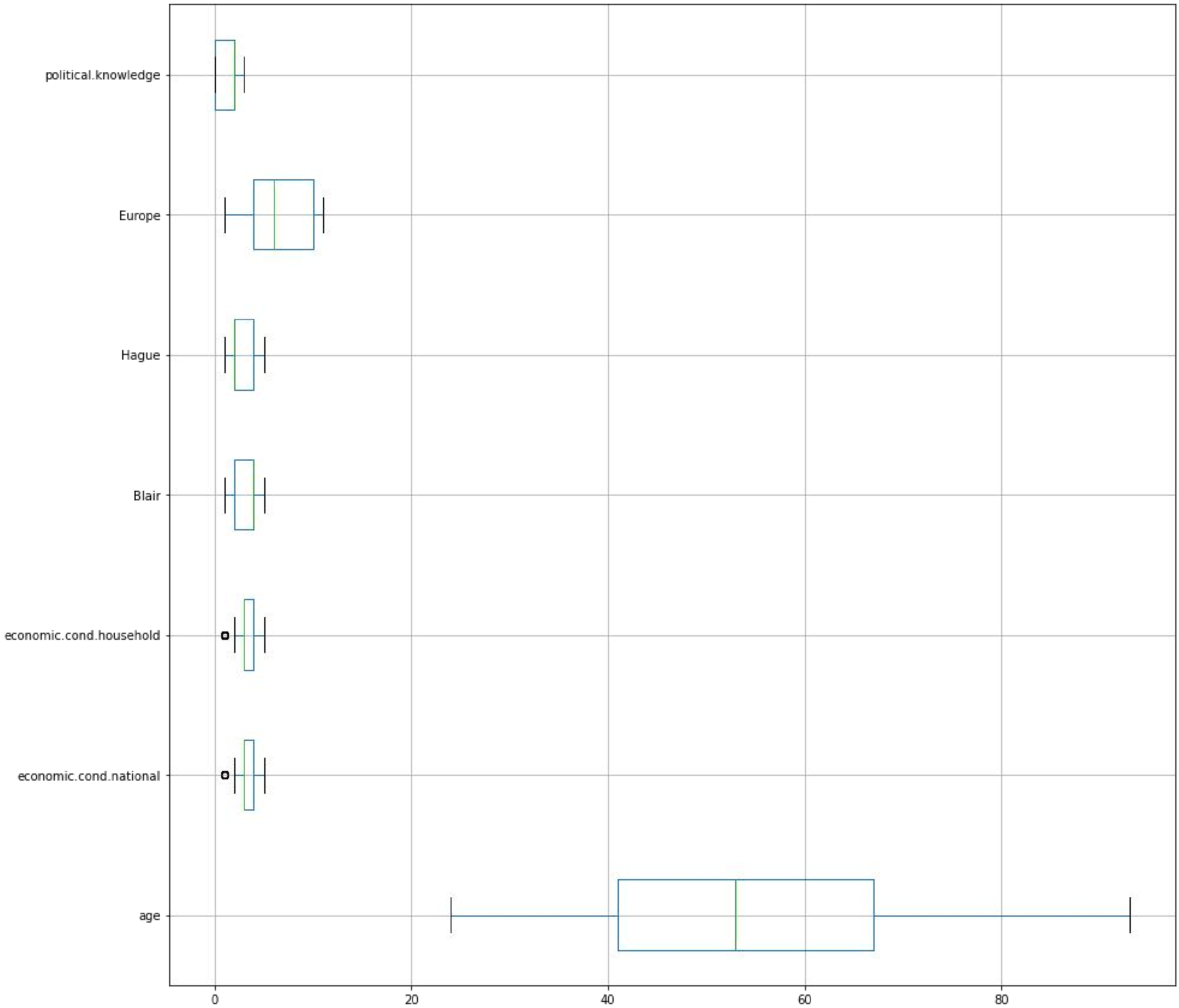

Fig 1.1 Box Plot to check Outliers

As per the above plot there are no outliers in the data and data is symmetrical.

Further Univariate analysis,

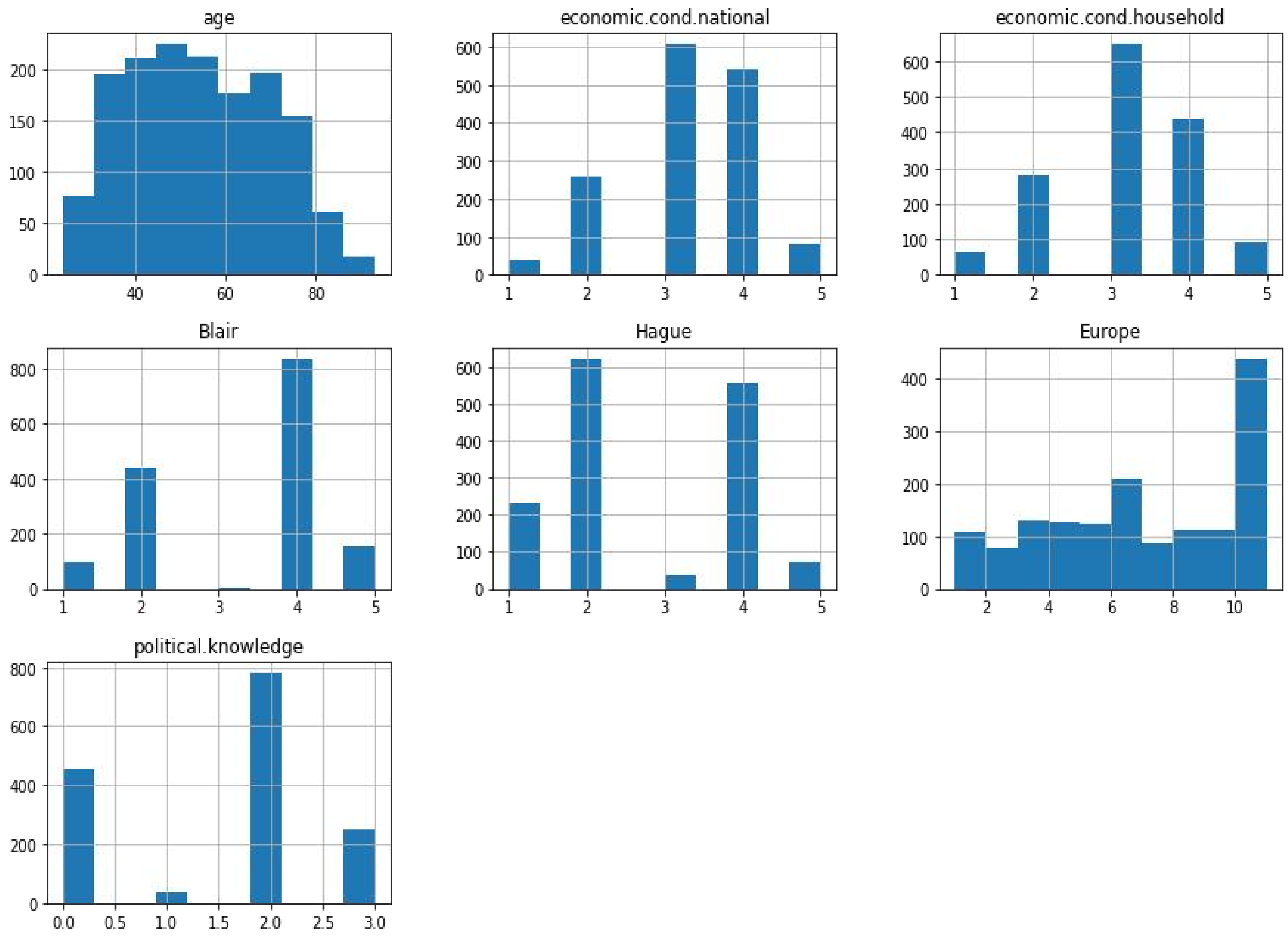

Fig 1.2 Histogram Plot for all the data.

From the plot,

- Most voter's age lies between 40-60

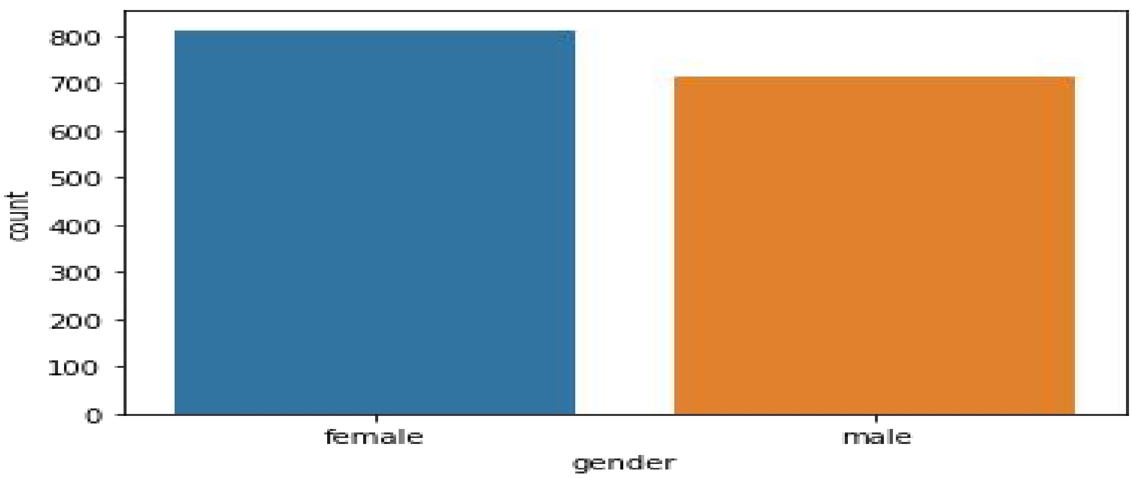

Fig 1.3 Count Plot - Gender.

Most voters are Female compared to male.

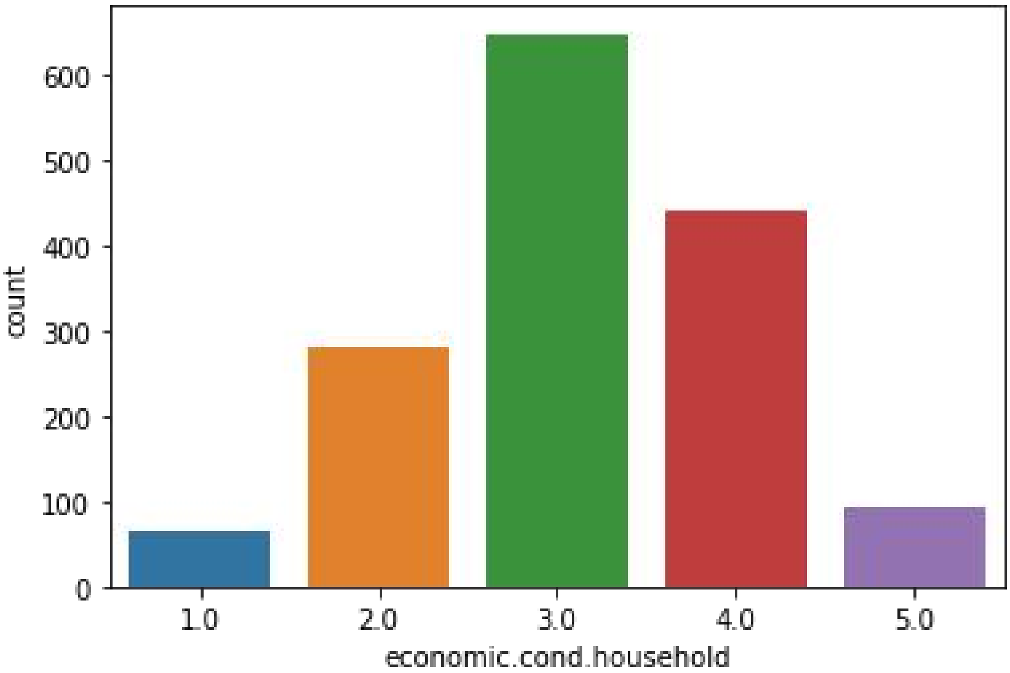

Fig 1.4 Count Plot - Household economic condition.

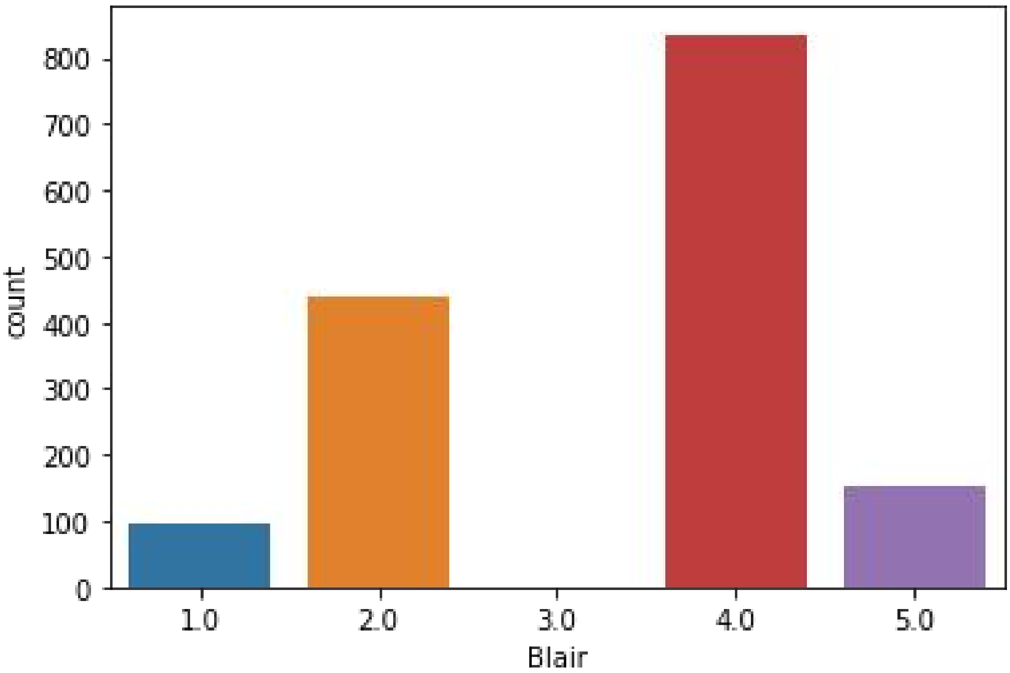

Fig 1.5 Count Plot - Blair.

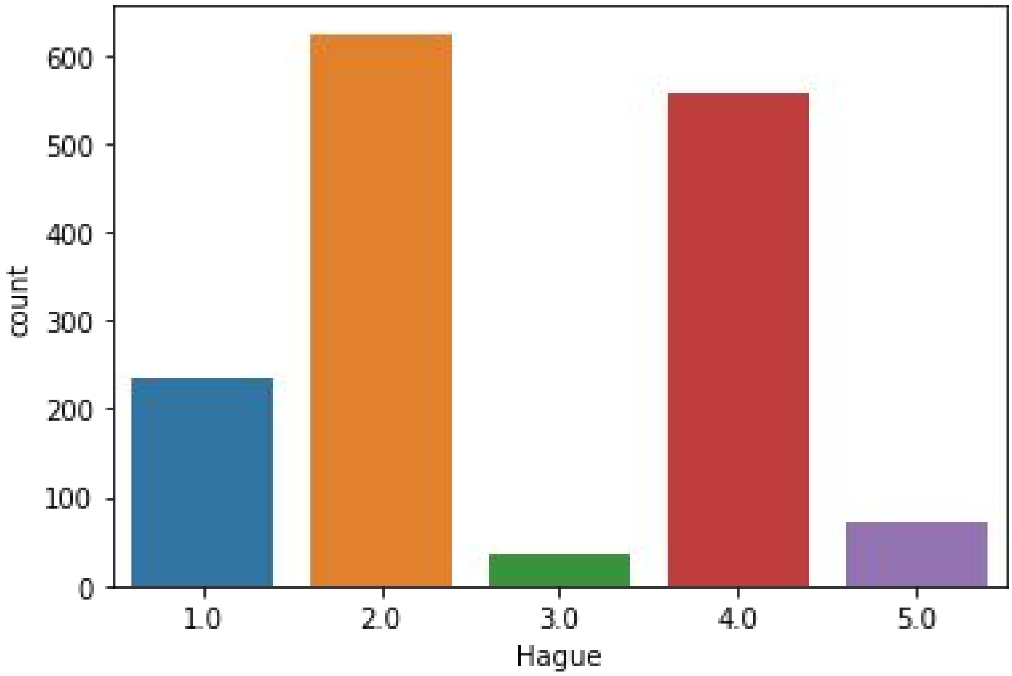

Fig 1.6 Count Plot - Hague

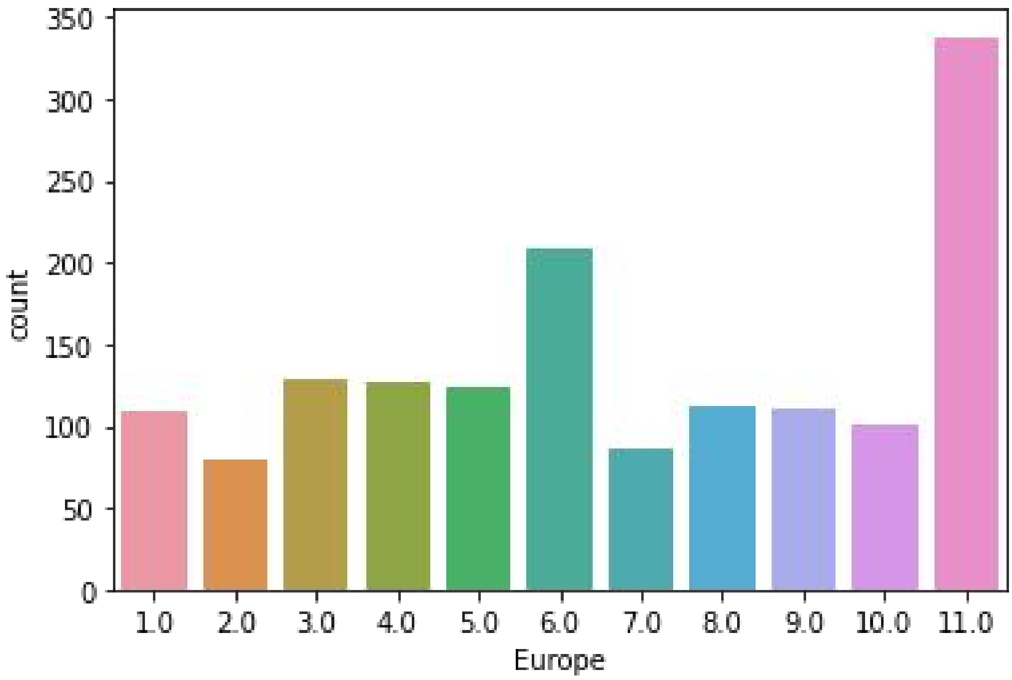

Fig 1.7 Count Plot - Europe

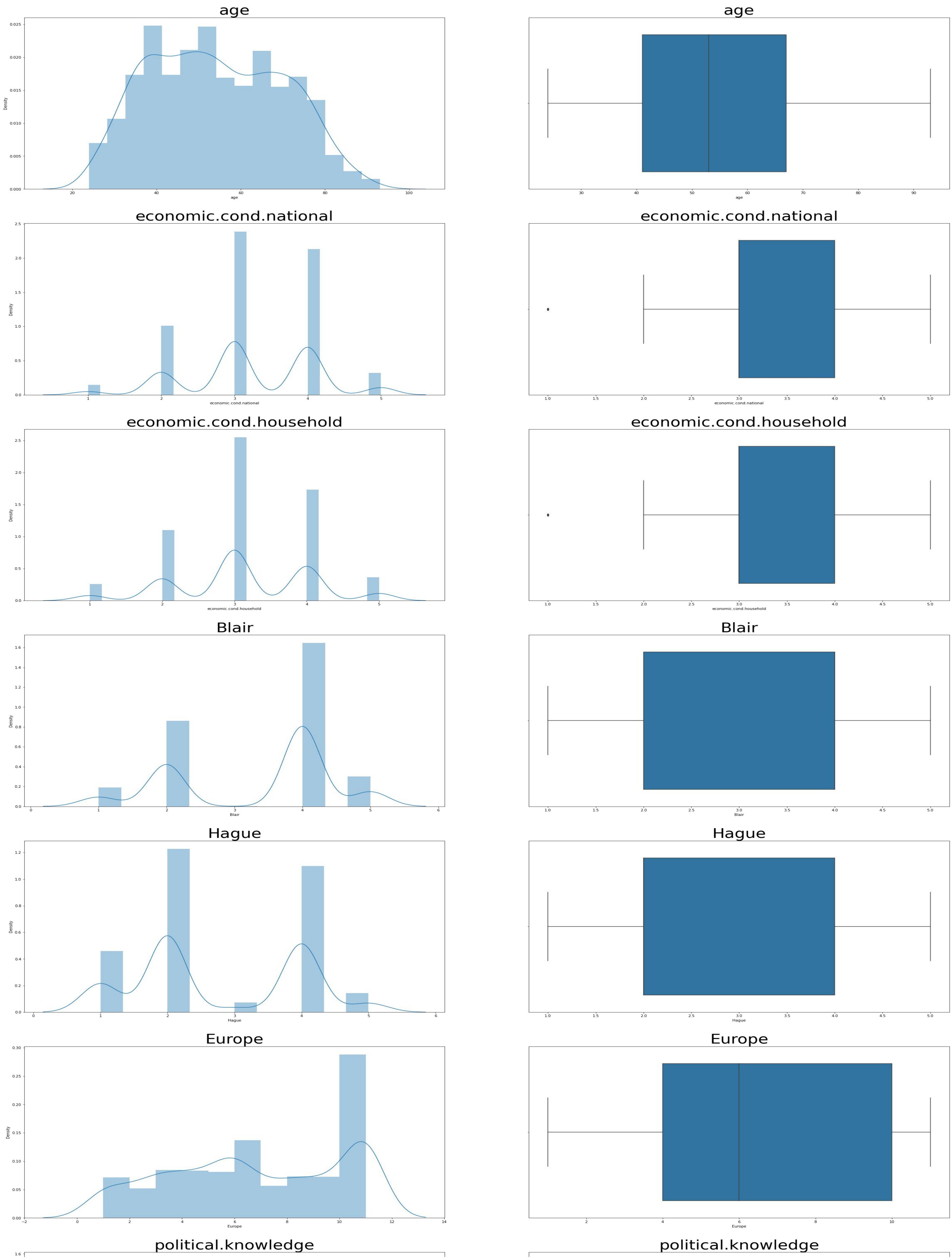

Fig 1.8 Distribution Plot

Multivariate analysis,

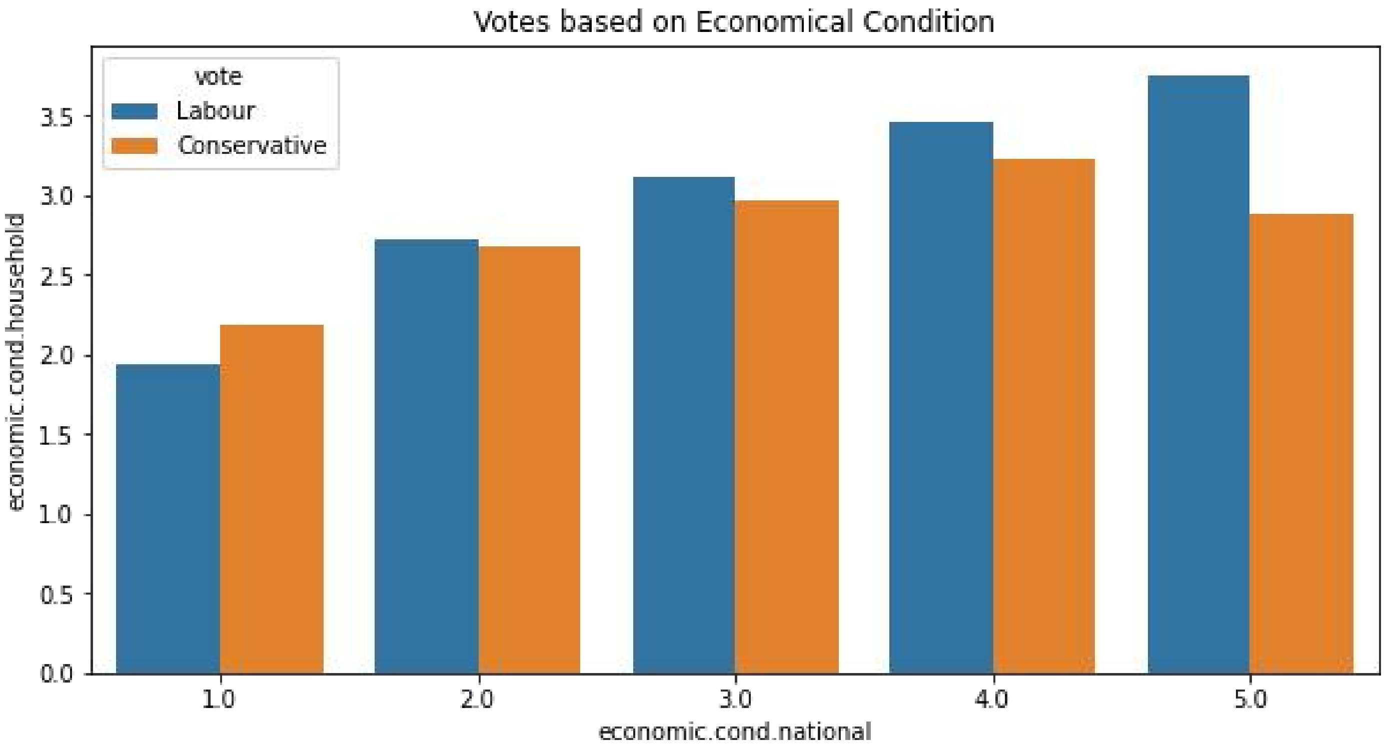

Fig 1.9 Count Plot - Votes Based on Economic Conditions.

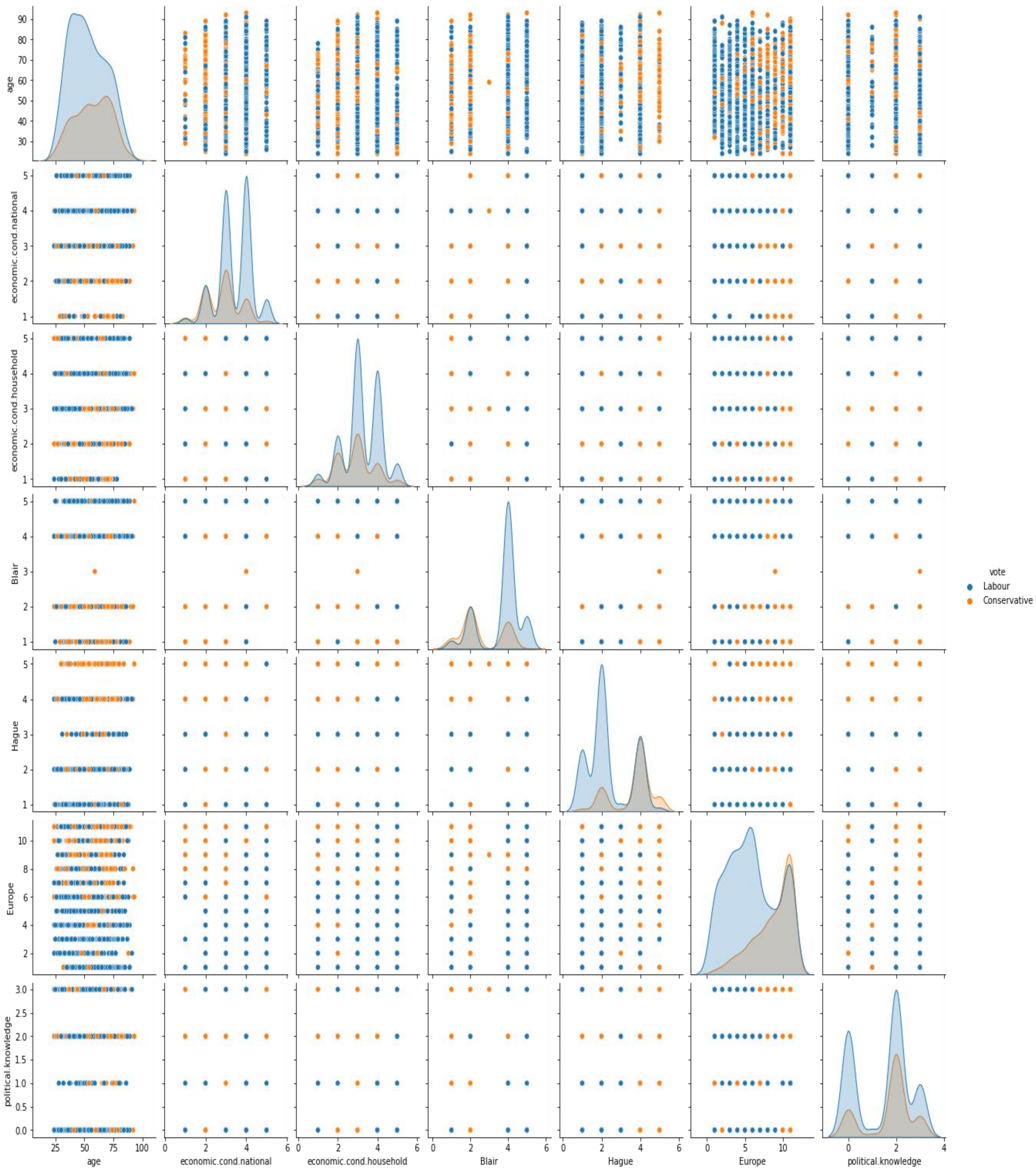

Fig 1.10 Pair Plot - Hue as Votes for Labour or Conservative parties

Fig 1.11 Correlation Plot

Based on th above plots,

- There is correlation to some extent in the data set among the variables, which is very low.
- **Blair** with **economic.cond.national** and **economic.cond.household** have a slight positive correlation.
- **Europe** with **Hague** have positive correlation.
- **Hague** with **economic.cond.national** have negative correlation.
- **Europe** with **economic.cond.national** and **Blair** have moderate negative correlation.
- **economic.cond.national** with **economic.cond.household** have positive correlation.

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
age	1.000000	0.018567	-0.041587	0.030218	0.034626	0.068880	-0.048490
economic.cond.national	0.018567	1.000000	0.346303	0.326878	-0.199766	-0.209429	-0.023624
economic.cond.household	-0.041587	0.346303	1.000000	0.215273	-0.101956	-0.114885	-0.037810
Blair	0.030218	0.326878	0.215273	1.000000	-0.243210	-0.296162	-0.020917
Hague	0.034626	-0.199766	-0.101956	-0.243210	1.000000	0.287350	-0.030354
Europe	0.068880	-0.209429	-0.114885	-0.296162	0.287350	1.000000	-0.152364
political.knowledge	-0.048490	-0.023624	-0.037810	-0.020917	-0.030354	-0.152364	1.000000

Table 1.5. Correlation

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
age	246.842075	0.256981	-0.607619	0.557762	0.669531	3.568550	-0.825301
economic.cond.national	0.256981	0.776107	0.283712	0.338314	-0.216589	-0.608397	-0.022546
economic.cond.household	-0.607619	0.283712	0.864810	0.235192	-0.116689	-0.352299	-0.038091
Blair	0.557762	0.338314	0.235192	1.380212	-0.351648	-1.147341	-0.026621
Hague	0.669531	-0.216589	-0.116689	-0.351648	1.514631	1.166149	-0.040469
Europe	3.568550	-0.608397	-0.352299	-1.147341	1.166149	10.873759	-0.544285
political.knowledge	-0.825301	-0.022546	-0.038091	-0.026621	-0.040469	-0.544285	1.173571

Table 1.6. Covariance

```
age                        246.842075
economic.cond.national       0.776107
economic.cond.household      0.864810
Blair                        1.380212
Hague                        1.514631
Europe                      10.873759
political.knowledge          1.173571
dtype: float64
```

Table 1.7. Variance

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).

<u>Encoding the data</u> :

Gender and vote needs to encoded for further Model building and analysis.

<u>Gender is encoding</u>

Female - 0

Male - 1

<u>Voter (Political Parties)</u>

Labour - 0

Conservative - 1

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

```
<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   int64
 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
 to scroll output; double click to hide  5 non-null   int64
 8   gender                   1525 non-null   int64
dtypes: int64(9)
memory usage: 107.4 KB
```

Encoding check in the data set using .head() function and .info() function

Data Split - 70:30. Shape of the training Data set.

```
(1067, 9) (458, 9) (1067,) (458,)
```

'**vote**' is the dependent variable based on which we will analyse all the other variables in the data set.

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

Logistic Regression Model :

For Train,

```
0.6807228915662651
[[668  67]
 [106 226]]
              precision    recall  f1-score   support

           0       0.86      0.91      0.89       735
           1       0.77      0.68      0.72       332

    accuracy                           0.84      1067
   macro avg       0.82      0.79      0.80      1067
weighted avg       0.83      0.84      0.83      1067
```

Accuracy - 84 %
Precision - 77%
Recall - 68 %
F1 score - 72 %

For test,

```
0.6538461538461539
[[292  36]
 [ 45  85]]
              precision    recall  f1-score   support

           0       0.87      0.89      0.88       328
           1       0.70      0.65      0.68       130

    accuracy                           0.82       458
   macro avg       0.78      0.77      0.78       458
weighted avg       0.82      0.82      0.82       458
```

Accuracy - 82 %
Precision - 70%
Recall - 65 %
F1 score - 68 %

Observation -

- Based on the above results, the model is neither under fitted nor over-fitted .

LDA Model :

For train,

```
0.7018072289156626
[[660  75]
 [ 99 233]]
              precision    recall  f1-score   support

           0       0.87      0.90      0.88       735
           1       0.76      0.70      0.73       332

    accuracy                           0.84      1067
   macro avg       0.81      0.80      0.81      1067
weighted avg       0.83      0.84      0.84      1067
```

Accuracy - 84 %
Precision - 76%
Recall - 70 %
F1 score - 73 %

For test,

```
0.6615384615384615
[[289  39]
 [ 44  86]]
              precision    recall  f1-score   support

           0       0.87      0.88      0.87       328
           1       0.69      0.66      0.67       130

    accuracy                           0.82       458
   macro avg       0.78      0.77      0.77       458
weighted avg       0.82      0.82      0.82       458
```

Accuracy - 82 %
Precision - 69%
Recall - 66 %
F1 score - 67 %

Observation -

- Based on the above results, the model is neither under fitted nor over-fitted .

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

KNN Model :

For train,

```
0.7409638554216867
[[673  62]
 [ 86 246]]
              precision    recall  f1-score   support

           0       0.89      0.92      0.90       735
           1       0.80      0.74      0.77       332

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

Accuracy - 86 %
Precision - 80%
Recall - 74 %
F1 score - 77 %

For test,

```
0.6230769230769231
[[279  49]
 [ 49  81]]
              precision    recall  f1-score   support

           0       0.85      0.85      0.85       328
           1       0.62      0.62      0.62       130

    accuracy                           0.79       458
   macro avg       0.74      0.74      0.74       458
weighted avg       0.79      0.79      0.79       458
```

Accuracy - 79 %
Precision - 62%
Recall - 62 %
F1 score - 62 %

Observation -

- Based on the above results, the model is under-fitted . Model tuning is required.

Naïve Bayes Model :

For train,

```
0.7289156626506024
[[653  82]
 [ 90 242]]
              precision    recall  f1-score   support

           0       0.88      0.89      0.88       735
           1       0.75      0.73      0.74       332

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

Accuracy - 84 %
Precision - 75%
Recall - 73 %
F1 score - 74 %

For test data,

```
0.7230769230769231
[[282  46]
 [ 36  94]]
              precision    recall  f1-score   support

           0       0.89      0.86      0.87       328
           1       0.67      0.72      0.70       130

    accuracy                           0.82       458
   macro avg       0.78      0.79      0.78       458
weighted avg       0.83      0.82      0.82       458
```

Accuracy - 84 %
Precision - 67%
Recall - 72 %
F1 score - 70%

Observation -

- Based on the above results, the model is neither under fitted nor over-fitted .

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

Here we have applied grid search with MinMax Scaler

Random Forest Model :

For train set,

```
0.6867469879518072
[[679  56]
 [104 228]]
              precision    recall  f1-score   support

           0       0.87      0.92      0.89       735
           1       0.80      0.69      0.74       332

    accuracy                           0.85      1067
   macro avg       0.83      0.81      0.82      1067
weighted avg       0.85      0.85      0.85      1067
```

Accuracy - 85 %
Precision - 80%
Recall - 69 %
F1 score - 74%

For test,

```
0.6538461538461539
[[294  34]
 [ 45  85]]
              precision    recall  f1-score   support

           0       0.87      0.90      0.88       328
           1       0.71      0.65      0.68       130

    accuracy                           0.83       458
   macro avg       0.79      0.78      0.78       458
weighted avg       0.82      0.83      0.83       458
```

Accuracy - 83 %
Precision - 71%
Recall - 65 %
F1 score - 68%

Observation -

- Based on the above results, the model is neither under fitted nor over-fitted .

Bagging Model (Random Forest is applied) :

For train,

```
0.9156626506024096
[[726   9]
 [ 28 304]]
```

	precision	recall	f1-score	support
0	0.96	0.99	0.98	735
1	0.97	0.92	0.94	332
accuracy			0.97	1067
macro avg	0.97	0.95	0.96	1067
weighted avg	0.97	0.97	0.97	1067

Accuracy - 97 %
Precision - 97%
Recall - 92 %
F1 score - 94%

For test,

```
0.7076923076923077
[[291  37]
 [ 38  92]]
```

	precision	recall	f1-score	support
0	0.88	0.89	0.89	328
1	0.71	0.71	0.71	130
accuracy			0.84	458
macro avg	0.80	0.80	0.80	458
weighted avg	0.84	0.84	0.84	458

Accuracy - 84 %
Precision - 71%
Recall - 71 %
F1 score - 71%

Observation -

- Based on the above results, the model is not a good model .
- Model tuning required.

<u>ADA Boosting Model</u> :

For train,

```
0.6536144578313253
[[674  61]
 [115 217]]
              precision    recall  f1-score   support

           0       0.85      0.92      0.88       735
           1       0.78      0.65      0.71       332

    accuracy                           0.84      1067
   macro avg       0.82      0.79      0.80      1067
weighted avg       0.83      0.84      0.83      1067
```

Accuracy - 84 %
Precision - 77%
Recall - 65 %
F1 score - 71%

For test,

```
0.6538461538461539
[[296  32]
 [ 45  85]]
              precision    recall  f1-score   support

           0       0.87      0.90      0.88       328
           1       0.73      0.65      0.69       130

    accuracy                           0.83       458
   macro avg       0.80      0.78      0.79       458
weighted avg       0.83      0.83      0.83       458
```

Accuracy - 83 %
Precision - 73%
Recall - 65 %
F1 score - 69%

<u>Observation -</u>

- Based on the above results, the model is neither under fitted nor over-fitted .

<u>Gradient Boosting Model</u> :

For Train,

```
0.7891566265060241
[[684  51]
 [ 70 262]]
              precision    recall  f1-score   support

           0       0.91      0.93      0.92       735
           1       0.84      0.79      0.81       332

    accuracy                           0.89      1067
   macro avg       0.87      0.86      0.87      1067
weighted avg       0.89      0.89      0.89      1067
```

Accuracy - 89 %
Precision - 84%
Recall - 79 %
F1 score - 81%

For test,

```
0.7384615384615385
[[285  43]
 [ 34  96]]
              precision    recall  f1-score   support

           0       0.89      0.87      0.88       328
           1       0.69      0.74      0.71       130

    accuracy                           0.83       458
   macro avg       0.79      0.80      0.80       458
weighted avg       0.84      0.83      0.83       458
```

Accuracy - 83 %
Precision - 69%
Recall - 74 %
F1 score - 71%

Observation -

- Based on the above results, the model is slightly under 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.

<u>Confusion Matrix</u> :

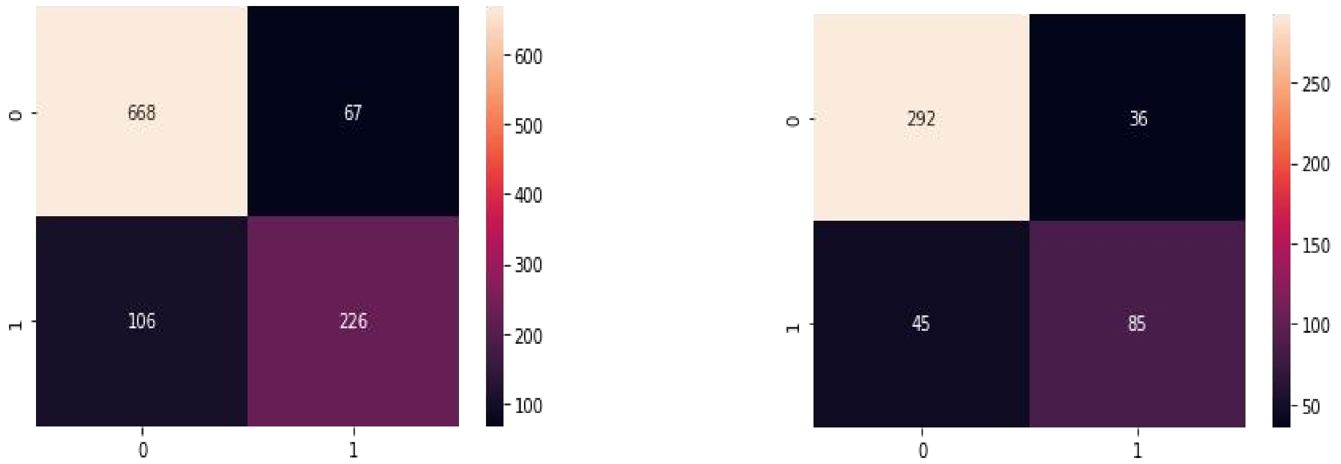

Fig 1.12 LOGREG - Train and Test

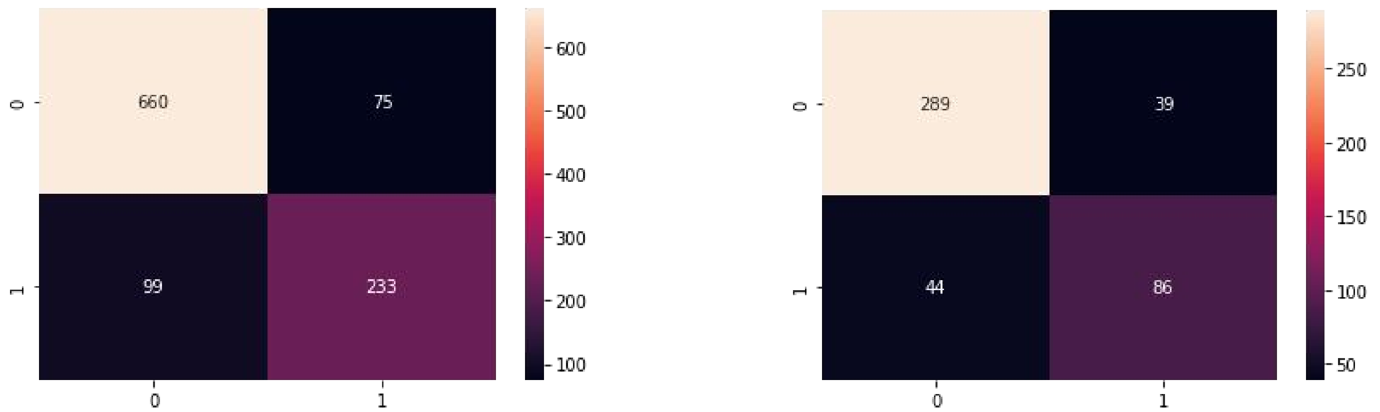

Fig 1.13 LDA - Train and Test

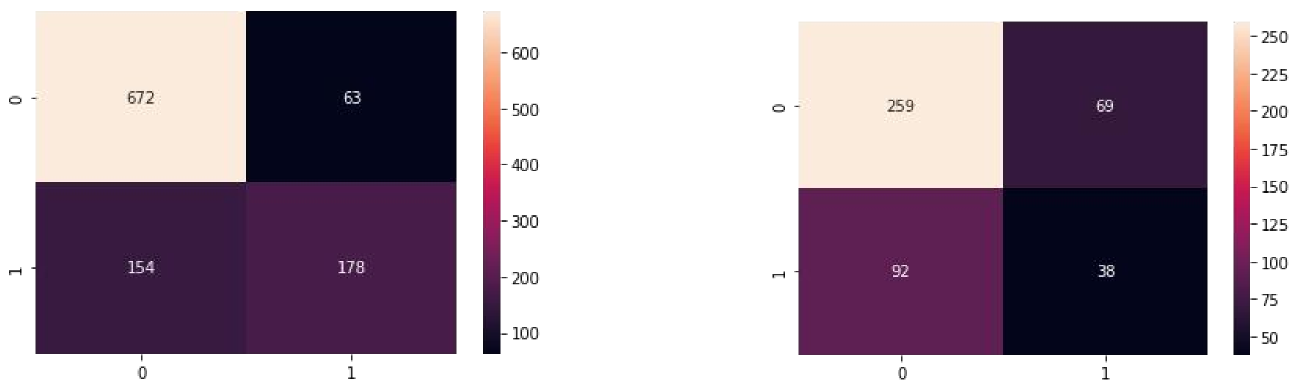

Fig 1.14 KNN - Train and Test

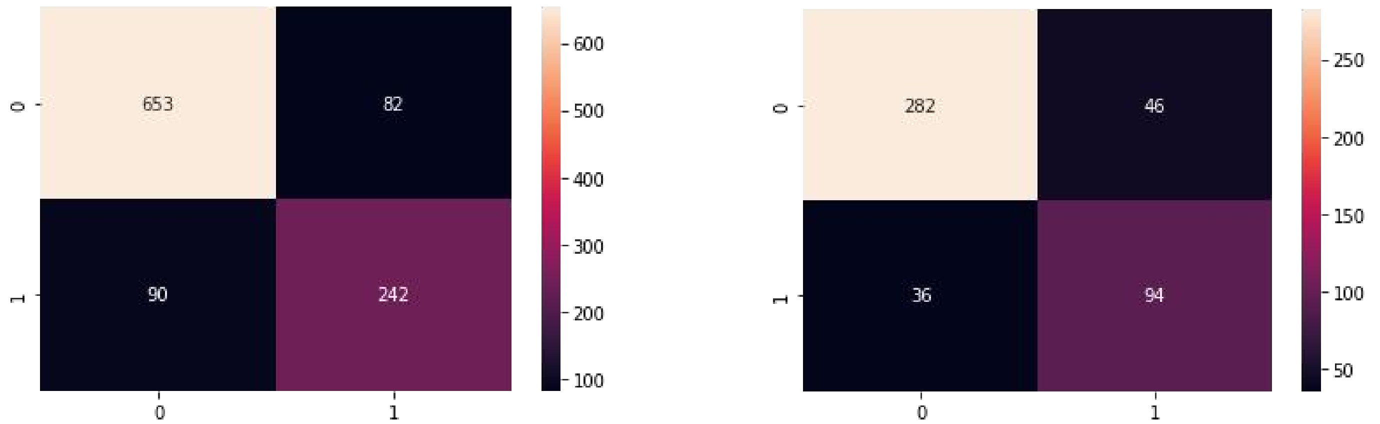

Fig 1.15 Naive Bayes Model - Train and Test

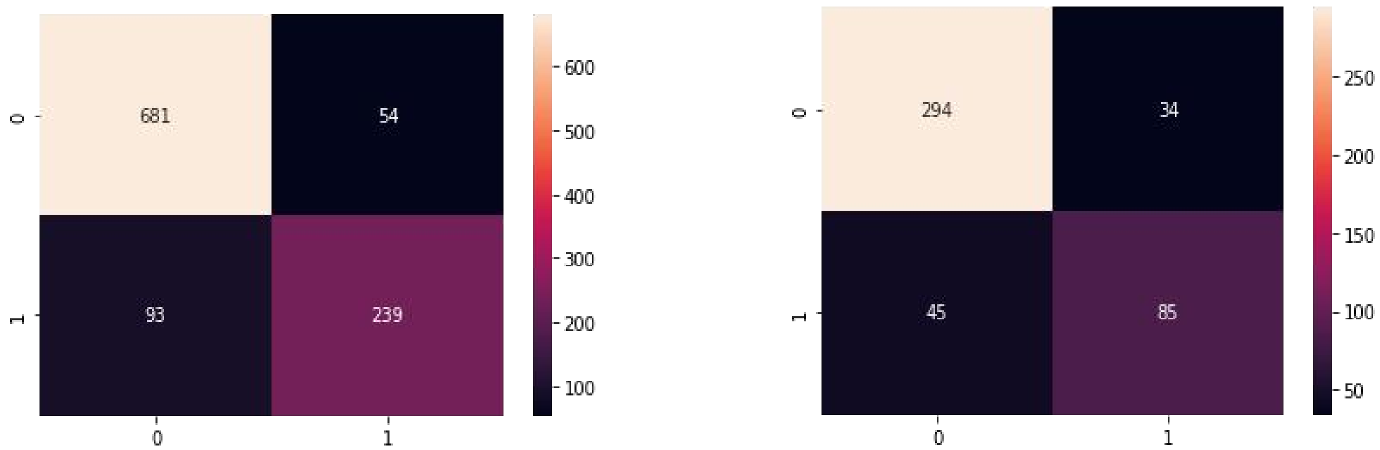

Fig 1.16 RF Model - Train and Test

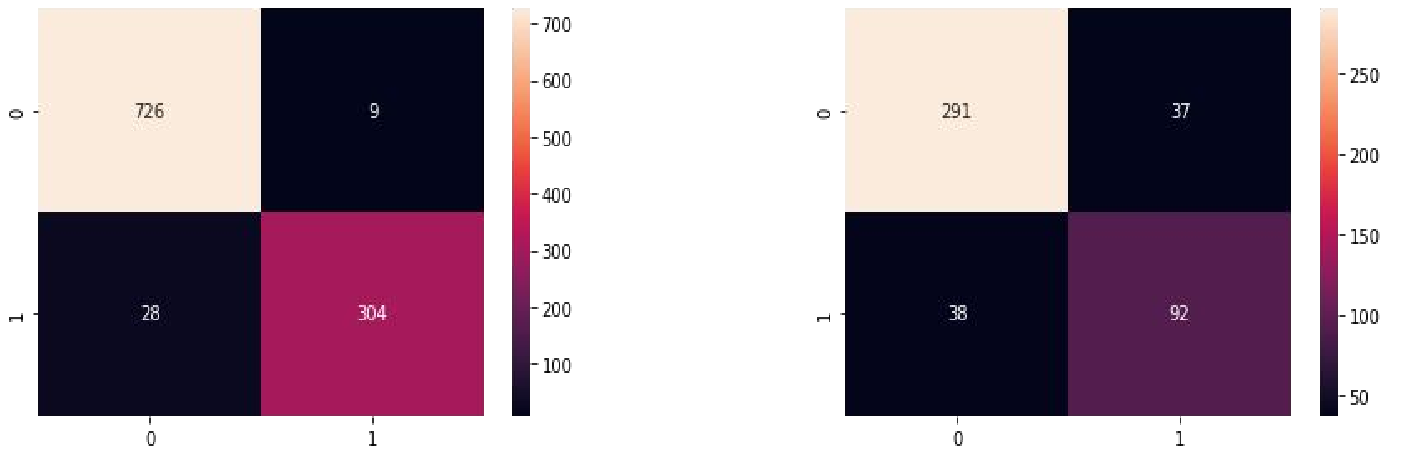

Fig 1.17 Bagging Model - Train and Test

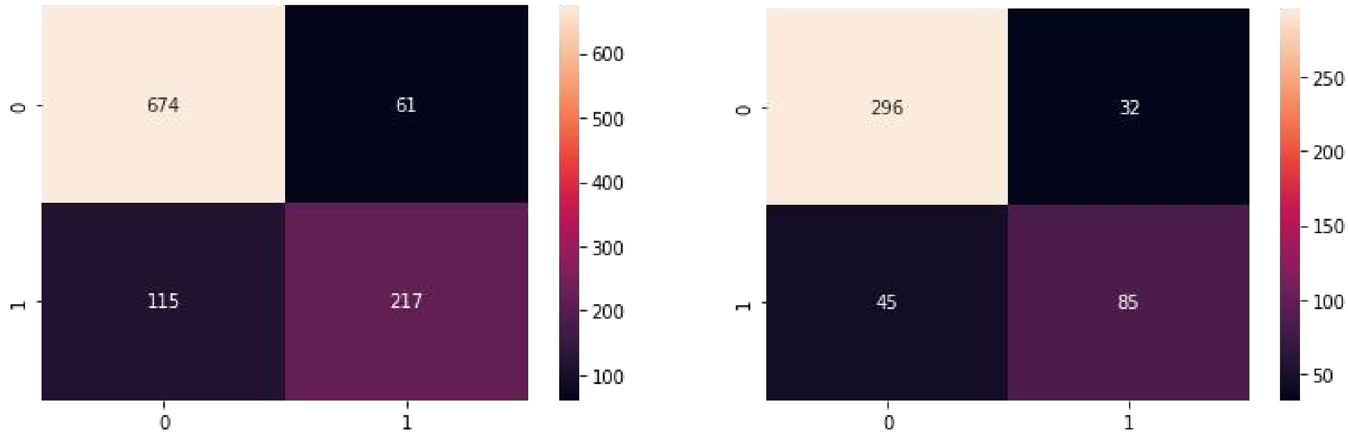

Fig 1.17 ADB Model - Train and Test

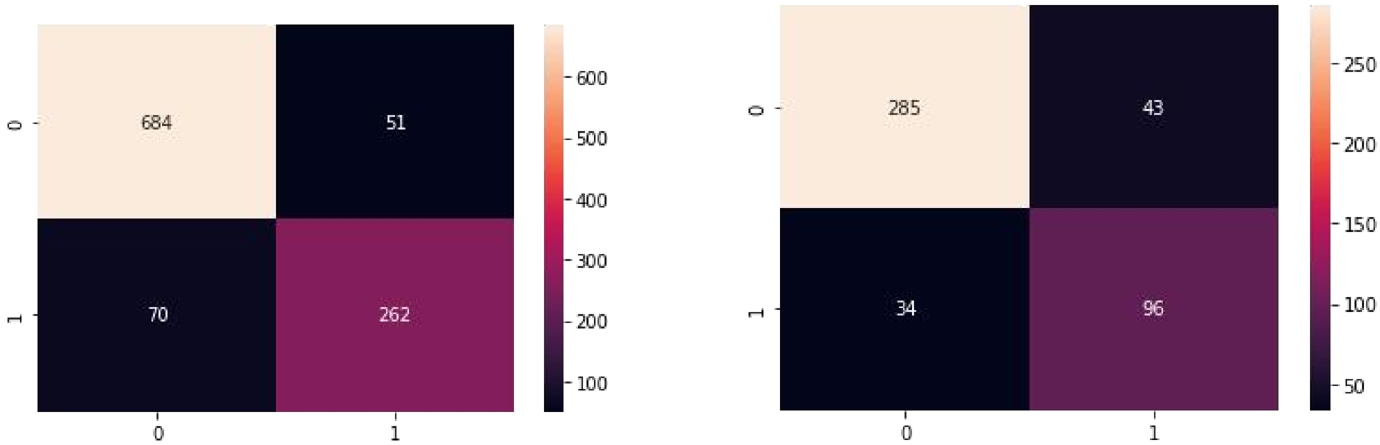

Fig 1.18 Gradient boosting Model - Train and Test

AUC - ROC Plot :

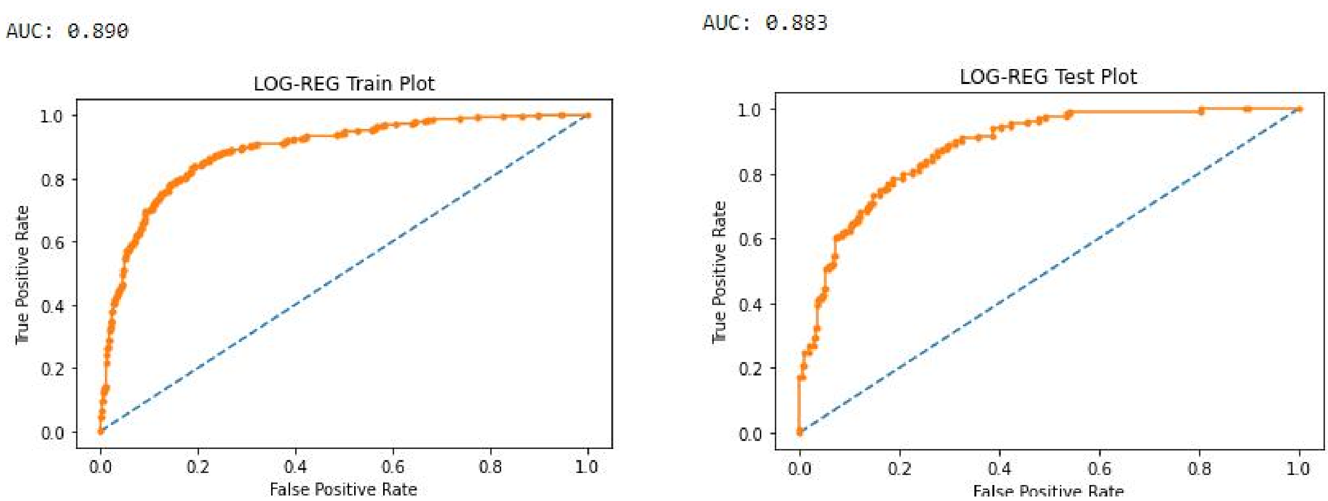

Fig 1.19. Logistic Regression - Train and test

AUC: 0.889

AUC: 0.884

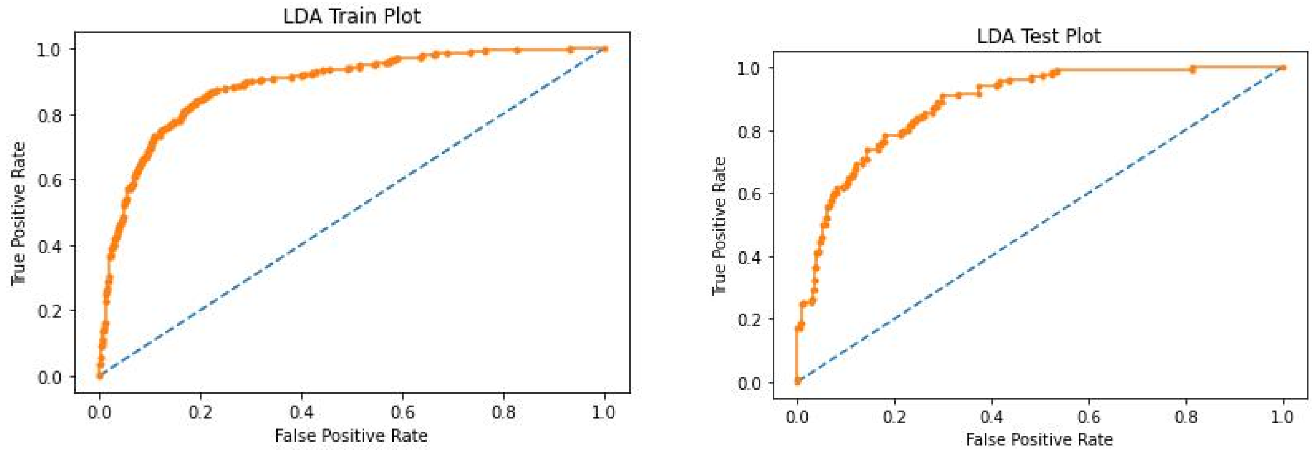

Fig 1.20. LDA - Train and test

AUC: 0.924

AUC: 0.832

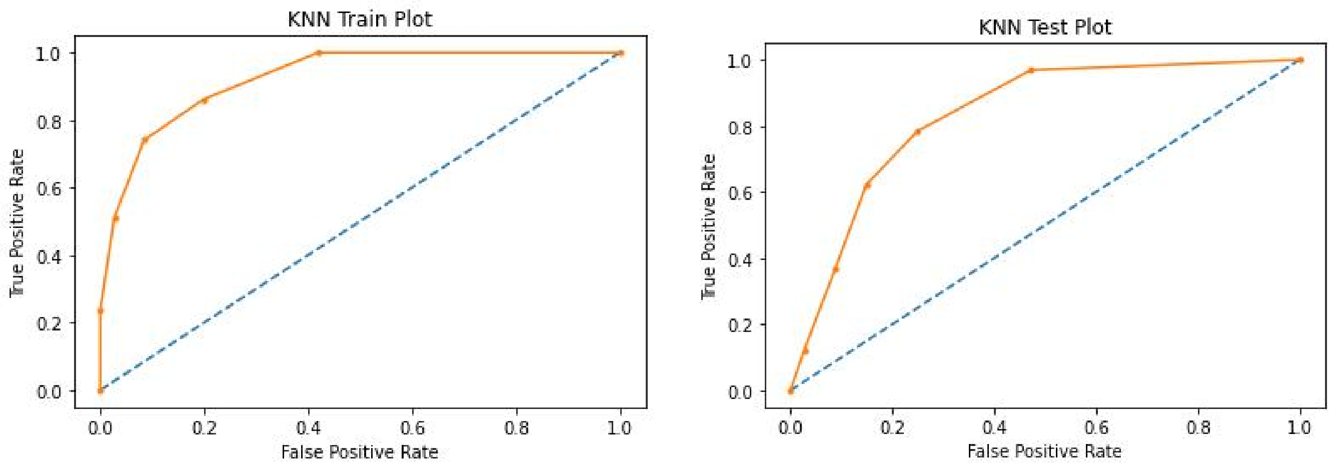

Fig 1.21. KNN - Train and test

AUC: 0.888

AUC: 0.886

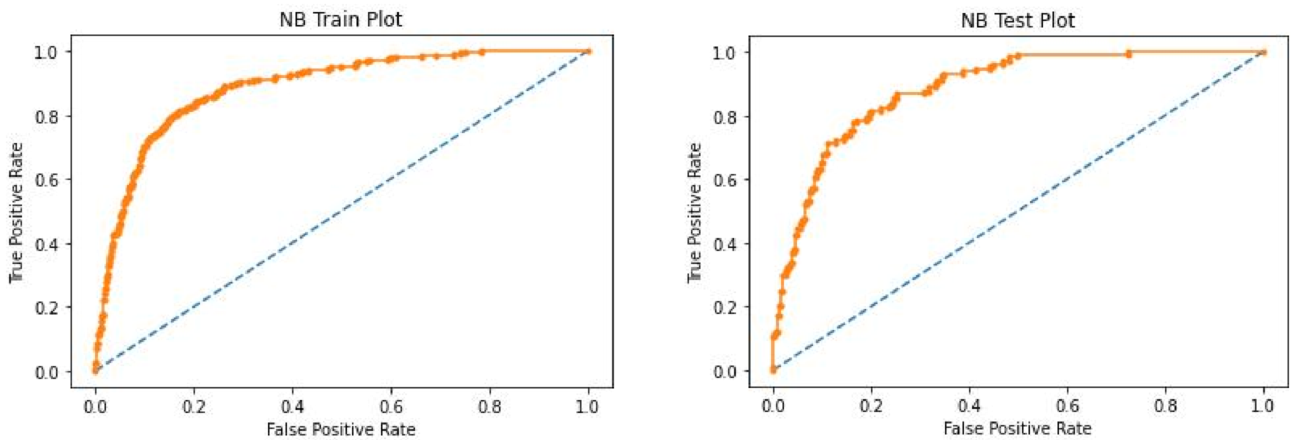

Fig 1.22. Naive Bayes Model - Train and test

AUC: 0.917

AUC: 0.897

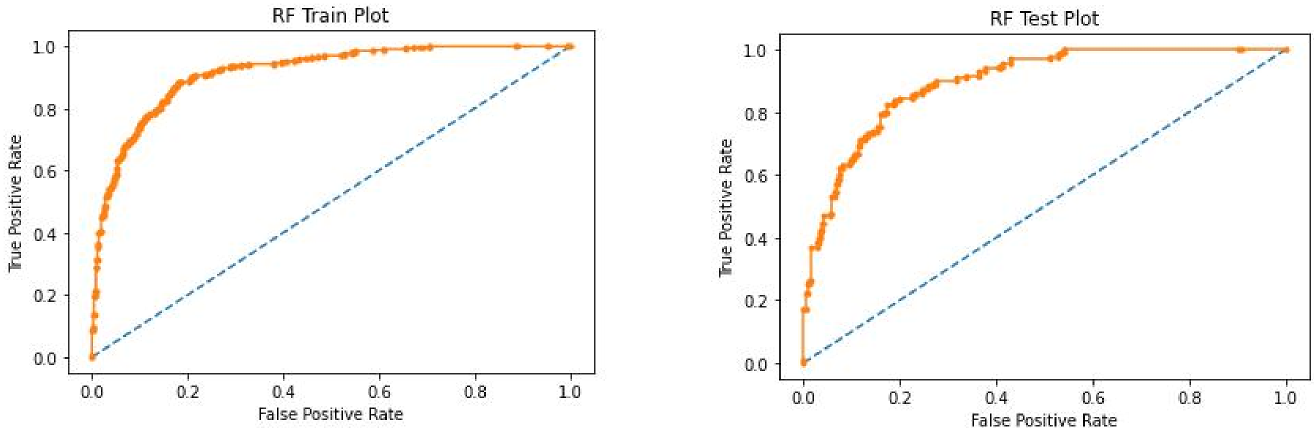

Fig 1.23. Random Forest Model - Train and test

AUC: 0.997

AUC: 0.897

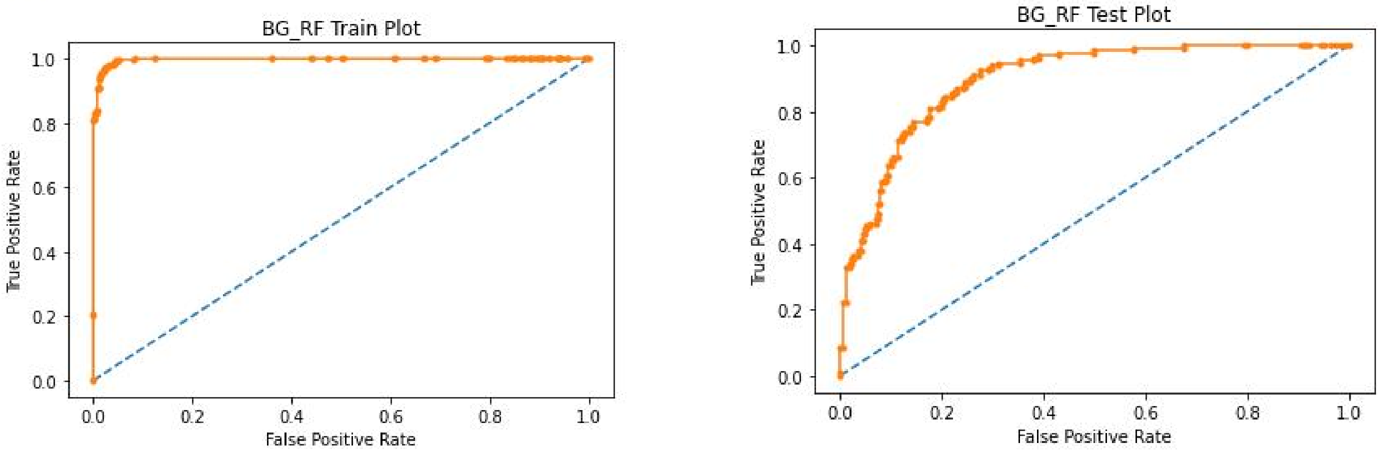

Fig 1.24. Bagging Model - Train and test

AUC: 0.913

AUC: 0.879

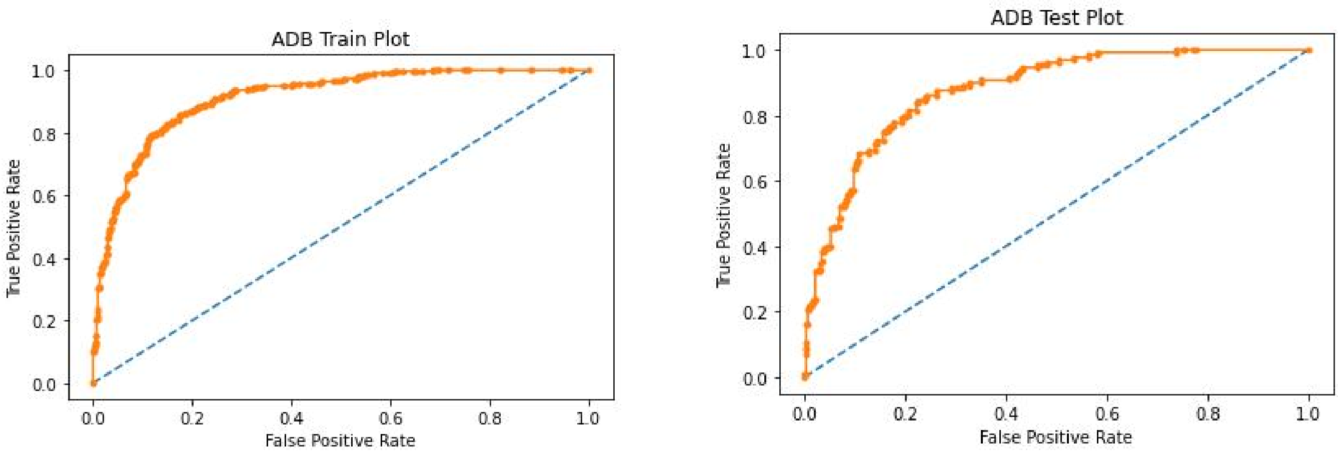

Fig 1.25. ADA Boosting Model - Train and test

AUC: 0.950

AUC: 0.900

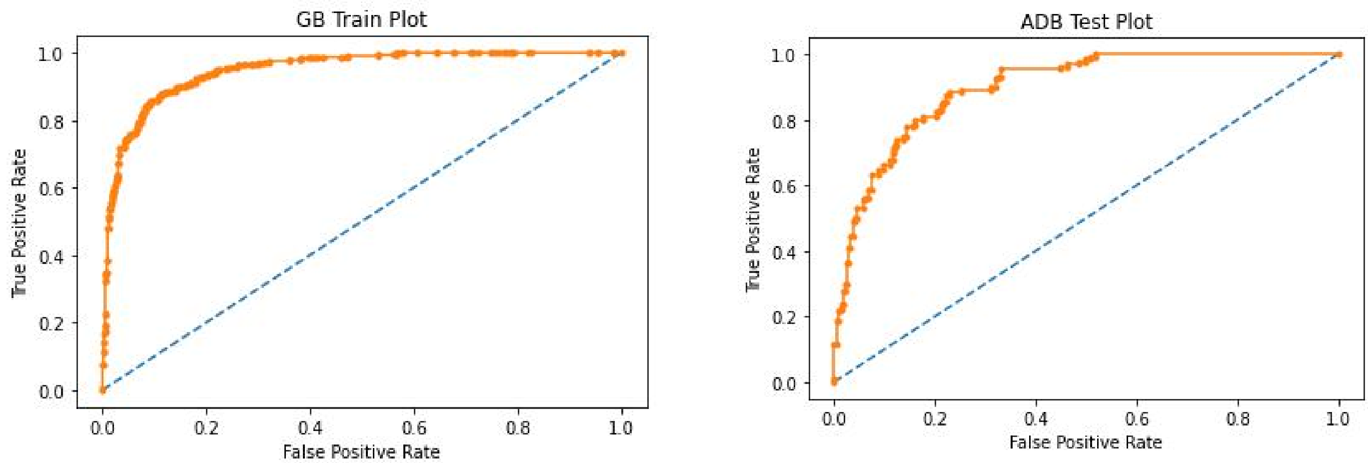

Fig 1.25. Gradient Boosting Model - Train and test

	Train Recall	Test Recall
LogReg_Model	0.680723	0.646154
LDA_Model	0.701807	0.661538
KNN_Model	0.740964	0.623077
NB_Model	0.722892	0.723077
RF_Model	0.686747	0.653846
RF_Bag_Model	0.915663	0.707692
ADB_Model	0.653614	0.653846
GB_Model	0.789157	0.738462

Table1.8 - Table for all Models - Recall values

Observations -

- Based on above plots, Gradient boosting is the best model observed.
- Although further tuning can reduce the under fitting or over fitting in the models.
- Bagging model and Random forest needs tuning.

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

Insights :-

- Labour party has more than double the votes than the conservative party.
- The average of national economic condition rating is 3.25.
- The average of economic household condition rating is 3.14.
- Blair has higher number of votes than the Hague.
- Europe has highest number of voters.
- Most of the voters have very low political knowledge.
- All the models have performed well in training data set as well as in test dat set.
- There is no over-fitting or under-fitting for most models, except in Random Forest and Bagging model. Here, model tuning is required.
- Gradient Boosting model tuned is the best model without tuning.

Business Intelligence:

- Gathering more data will help in improving the various ratings
- More data of other regions maybe collected like region wise, religion-wise, based on jobs and employments- business,private, ethnicity, etc.
- These data will help us understand the mindset of voters towards both parties.
- There are lot more analysis or visualization plots that can be created for various variables based on the requirement.

Problem 2:

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

1. **President Franklin D. Roosevelt in 1941**
2. **President John F. Kennedy in 1961**
3. **President Richard Nixon in 1973**

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

	Words_count	Sent_count	Char_count
Preseident Roosevelt	1526	68	7571
President Kennedy	1543	52	7618
President Nixon	2006	68	9991

Tabe 2.1. Total number of words, sentences and characters