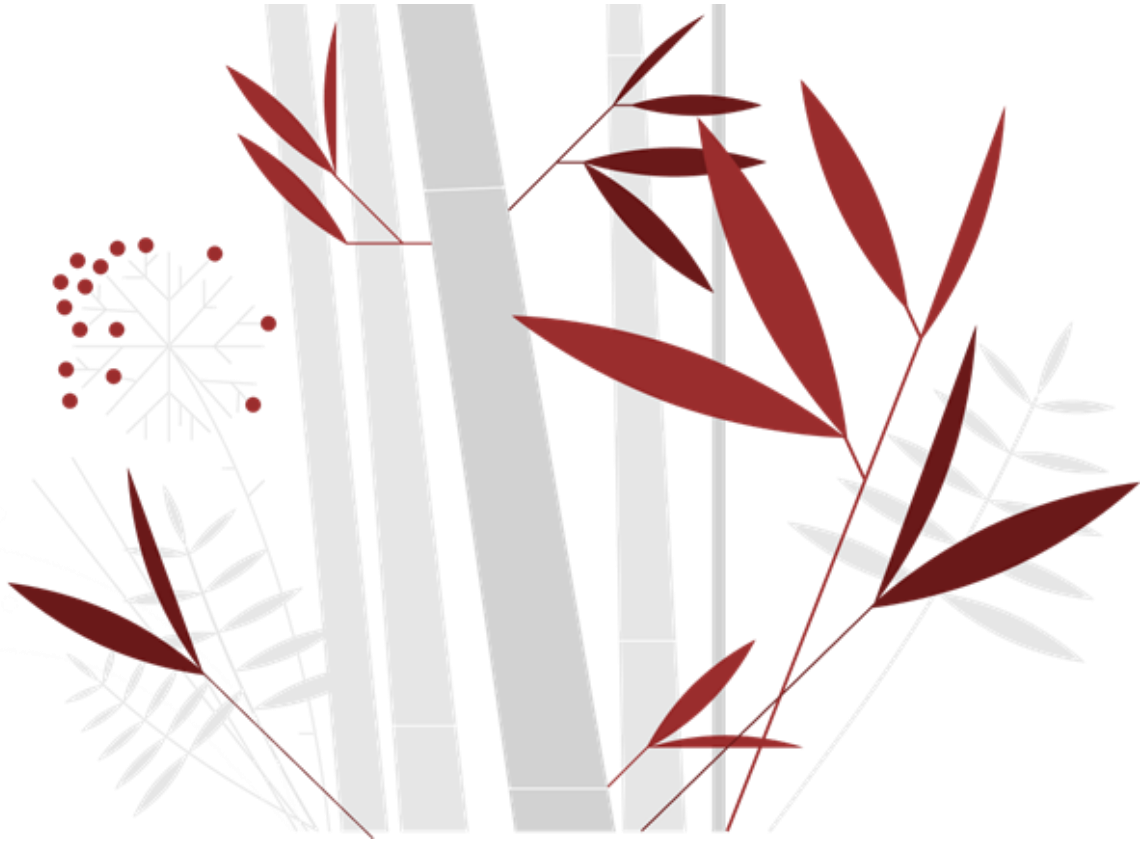
MACHINE LEARNING-2 (ML-2) PROJECT-CODED

BY

Harsh Patel

7th July 2024



Problem-1: Election Exit Poll

CNBE, a prominent news channel, is gearing up to provide insightful coverage of recent elections, recognizing the importance of data-driven analysis. A comprehensive survey has been conducted, capturing the perspectives of 1525 voters across various demographic and socio-economic factors. This dataset encompasses 9 variables, offering a rich source of information regarding voters' characteristics and preferences.

The primary objective is to leverage machine learning to build a predictive model capable of forecasting which political party a voter is likely to support. This predictive model, developed based on the provided information, will serve as the foundation for creating an exit poll. The exit poll aims to contribute to the accurate prediction of the overall election outcomes, including determining which party is likely to secure the majority of seats.

Data Variables:

- **vote**: Party choice: Conservative or Labour
- **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 Exploratory Data Analysis:

Using read excel function we load our dataset named "Election Data" and using head function we see the first three rows of the dataset to get some idea about dataset and it's variables. We can see that there is a variable called "Unnamed: 0" that is of no use for us and using the drop function we drop it and now again using head and tail function to see the first and last five rows as per fig-1 and fig-2 shown below.

	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	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 first five rows

	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
1520	Conservative	67	5	3	2	4	11	3	male
1521	Conservative	73	2	2	4	4	8	2	male
1522	Labour	37	3	3	5	4	2	2	male
1523	Conservative	61	3	3	1	4	11	2	male
1524	Conservative	74	2	3	2	4	11	0	female

Table-2 last five rows

Now, we use shape function the dataset and we get that there are 1529 row and 9 columns. Then, we use info function and found out the data type of each column and used value counts functions on the categorical variables as shown in the below table.

```
<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-3 Information table

we will check for the duplicated rows are present or not using duplicate function and found out that in the dataset there are 8 duplicated rows present and we removed those rows using drop duplicates function on the whole dataset and using shape function we can say we have 1517 rows and 9 columns.

Other, thing we can say from the above table is that there are 7 integer and 2 object data type variables present in our dataset. We also checked for any null or empty values in any variable using is null function and we can say that no empty values are present in our dataset.

Now, we will use describe function on this dataset and obtain the min, max, std, count, mean, q1, q2, q3 values for all the numerical columns as shown in the below table. We can observe that for some the numerical variables the min, max values are too larger i.e. they are far apart from each other indicating presence of outliers in them as well as there is variation between variables so we will need to scale them.

	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	1517.0	3.335531	1.174772	1.0	2.0	4.0	4.0	5.0
Hague	1517.0	2.749506	1.232479	1.0	2.0	2.0	4.0	5.0
Europe	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-4 Description of Numerical variables

Form the above table it can be said that the voters age is between 24 to 94 years and many voters have zero political knowledge. Other than that, for now no anomalies are present currently in the dataset.

We also checked the value counts for the categorical variables and can be seen in the below table.

```
economic.cond.household ---->
economic.cond.household
3    645
4    435
2    280
5     92
1     65
Name: count, dtype: int64


Blair ---->
Blair
4    833
2    434
5    152
1     97
3      1
Name: count, dtype: int64


Hague ---->
Hague
2    617
4    557
1    233
5     73
3     37
Name: count, dtype: int64
```

```
Europe ---->
Europe
11    338
6     207
3     128
4     126
5     123
9     111
8     111
1     109
10    101
7      86
2      77
Name: count, dtype: int64


political.knowledge ---->
political.knowledge
2    776
0    454
3    249
1     38
Name: count, dtype: int64
```

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


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

Table-5 Value counts of all variable

Using the skew function, we get the given below Table-6 showing the skewness of each numerical variables.

```
age                    0.139800
economic.cond.national    -0.238474
economic.cond.household   -0.144148
Blair                  -0.539514
Hague                   0.146191
Europe                 -0.141891
political.knowledge       -0.422928
dtype: float64
```

Table-6 Skewness of integer data types

From the above it can be said that variables age and Hague shows slight right skewness and all other variables are left skewed.

1.1.1 Univariate Analysis:

First of all, we will perform univariate analysis on the numerical and categorical dataset and drawn observation from them. We will also perform outlier treatment on the numerical dataset as per requirements.

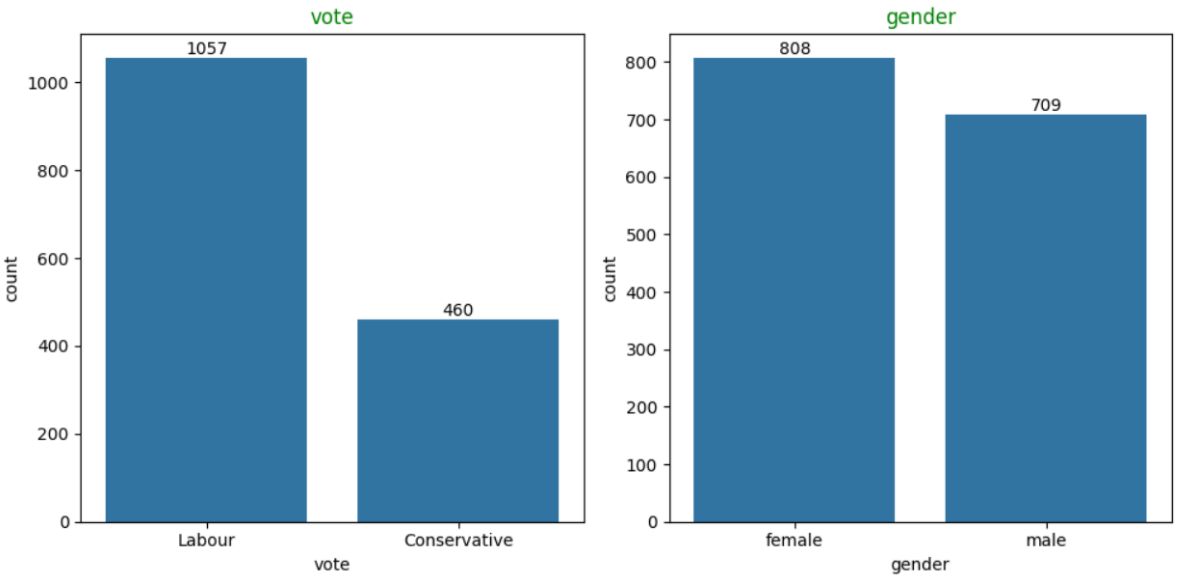


Fig-1 Count plot for Vote and Gender variables

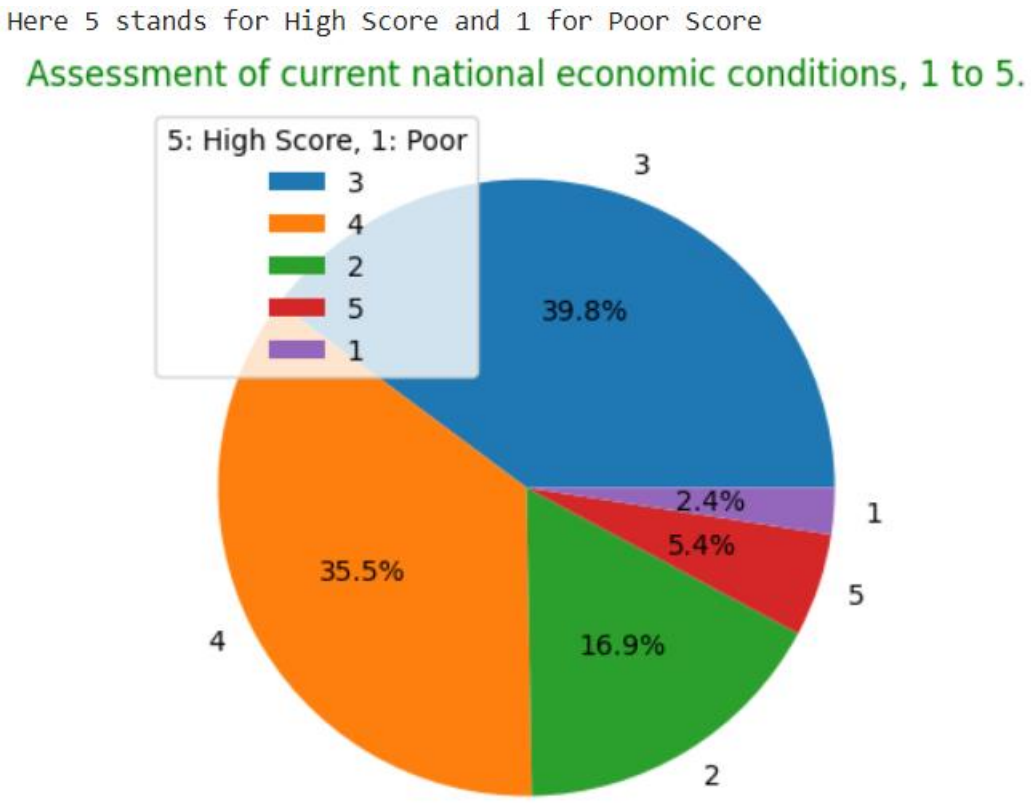


Fig-2 Pie chart for National Economic Condition variable

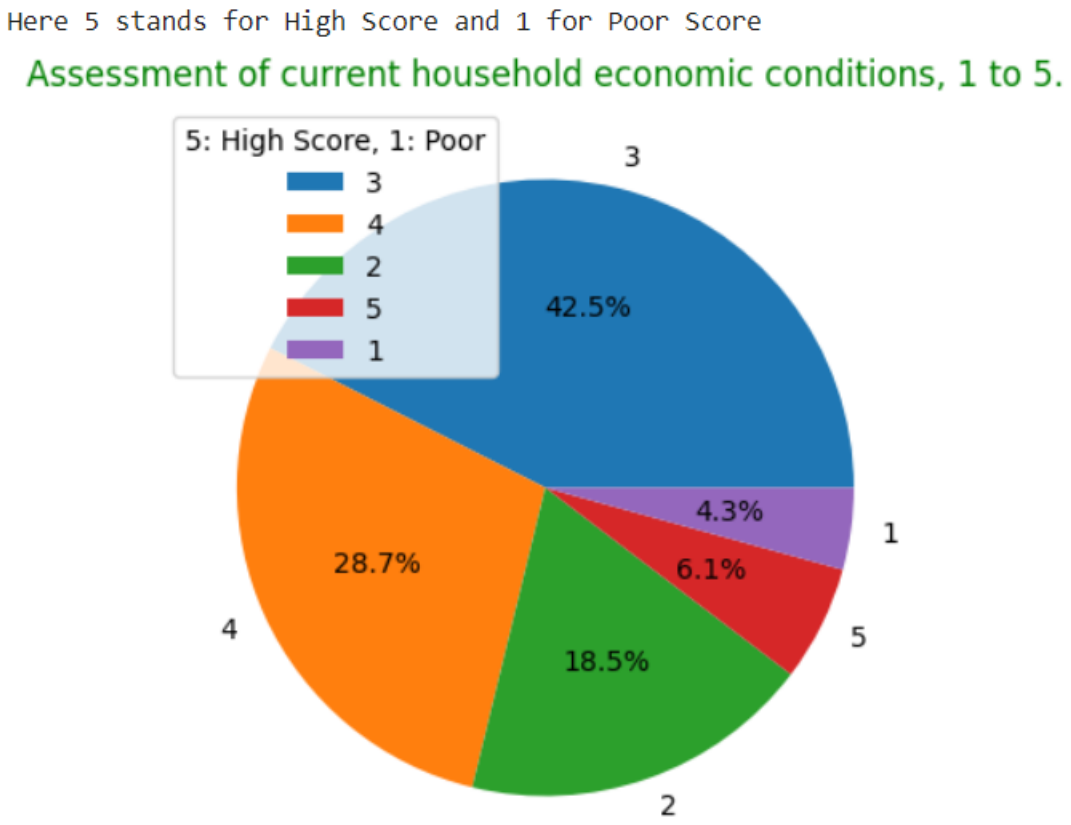


Fig-3 Pie chart for Household Economic Condition variable

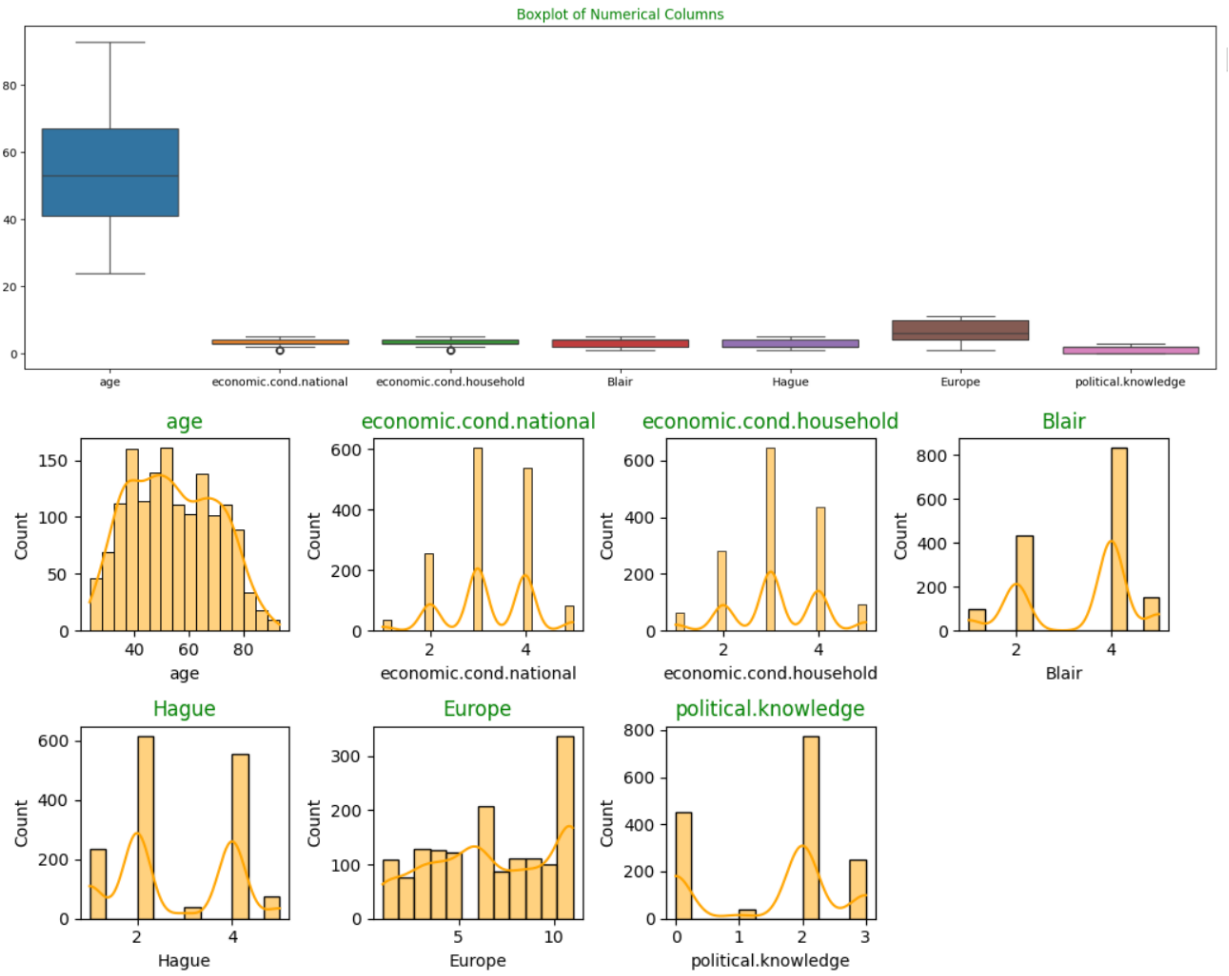


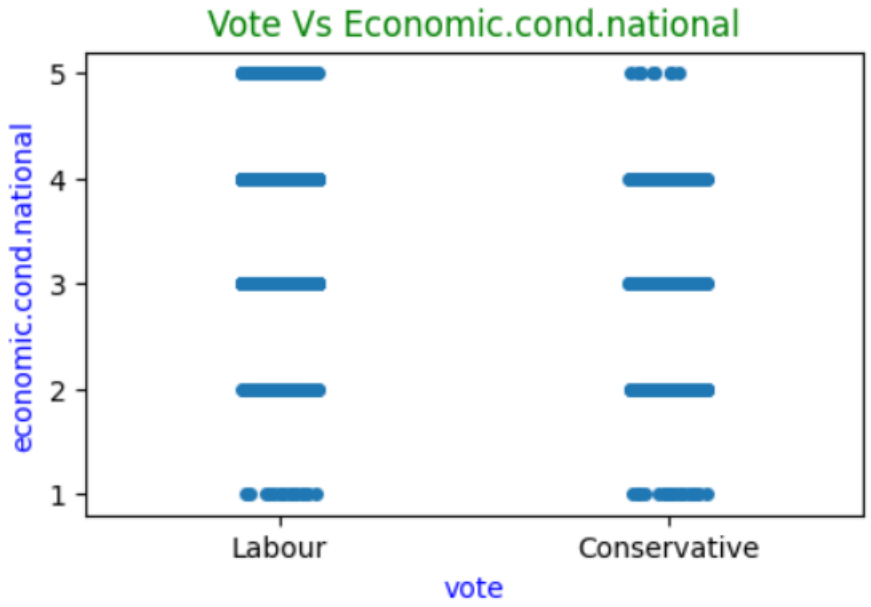
Fig-4 Histogram and boxplot for all integer Data type

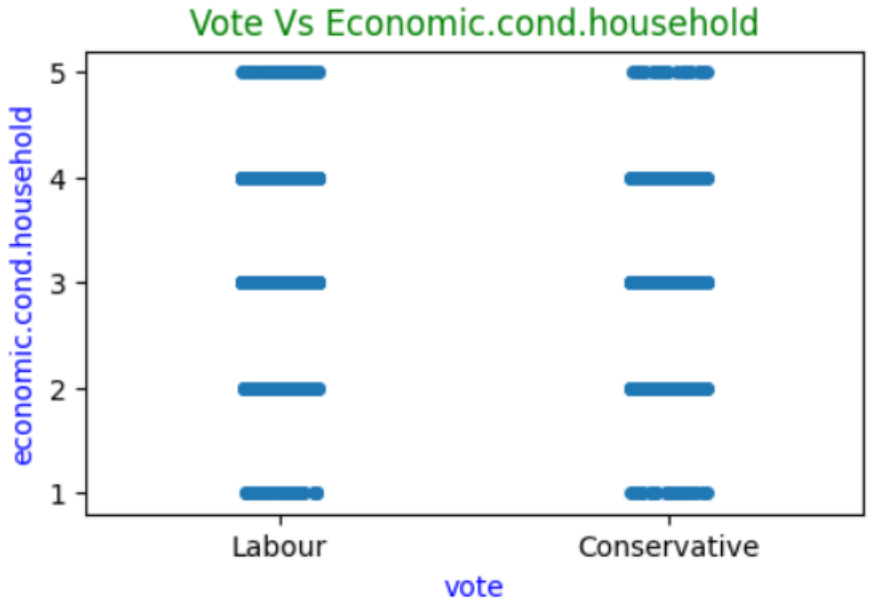
From the above Univariate analysis few insights have been obtained and they are:

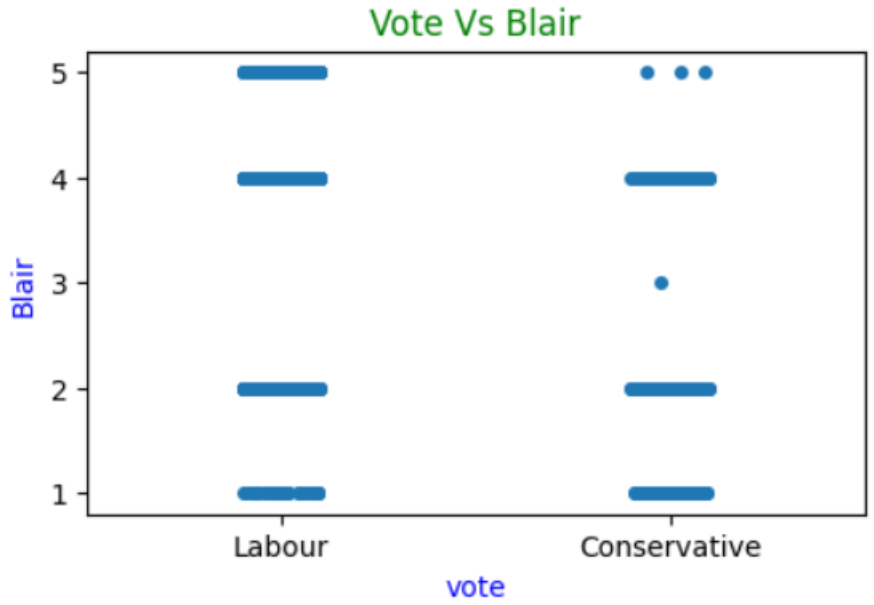
- Labour accounts for approximately 70% of the votes, indicating a clear preference among voters.
- The age distribution appears normal, with small peaks around ages 40, 50, and 70.
- Both national and household economic conditions are predominantly rated as "Condition 3."
- The Labour leader receives positive sentiment (score 4), just below the highest rating (5).
- The Conservative leader has less favorable sentiment (score 2) but also receives mixed opinions (score 4). And most voters express Eurosceptic sentiment
- Only 1.3% of voters had full knowledge of party positions on European integration.
- Approximately 30% lacked knowledge in this area.
- The gender variable is balanced, but the majority of voters are female

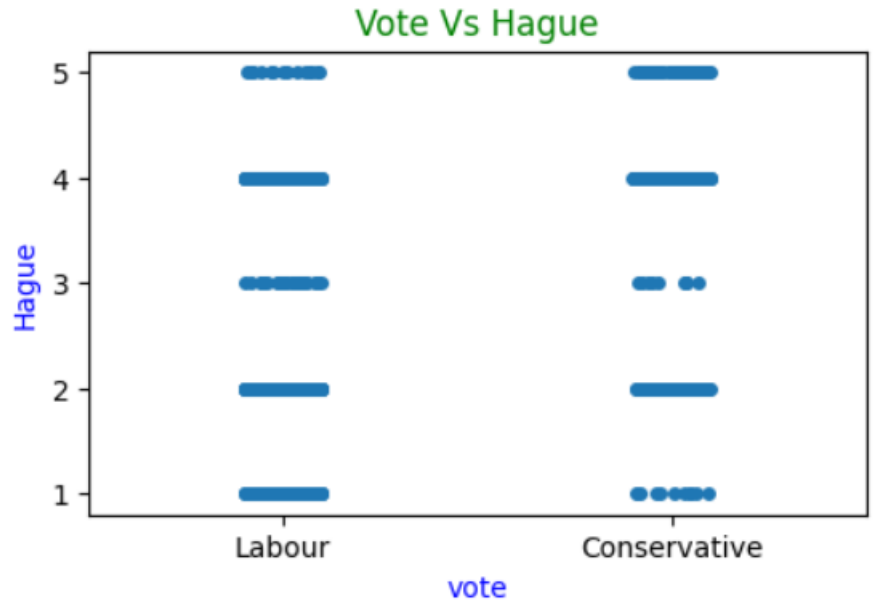
1.1.2 Multivariate Analysis:

Now, we will perform Multi-variate Analysis on the numerical data to see and find any correlations between the variables and find any hidden pattern the below figure shows the Variable vote vs all other variables plotted using strip plot.






Vote Vs Blair
Blair
5
4
3
2
1
Labour
Conservative
vote


Vote Vs Hague
Hague
5
4
3
2
1
Labour
Conservative
vote

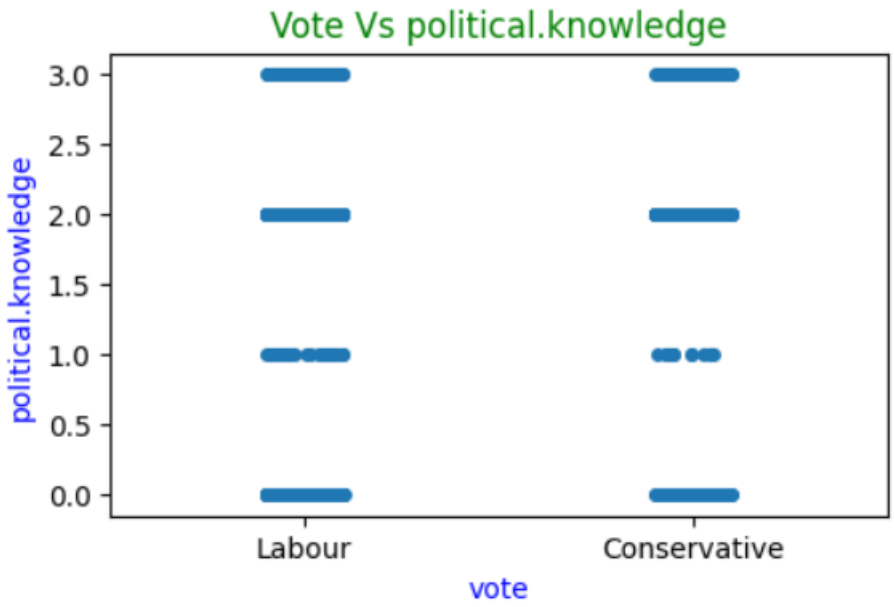


Fig-5 Strip plot for Vote vs all other variables

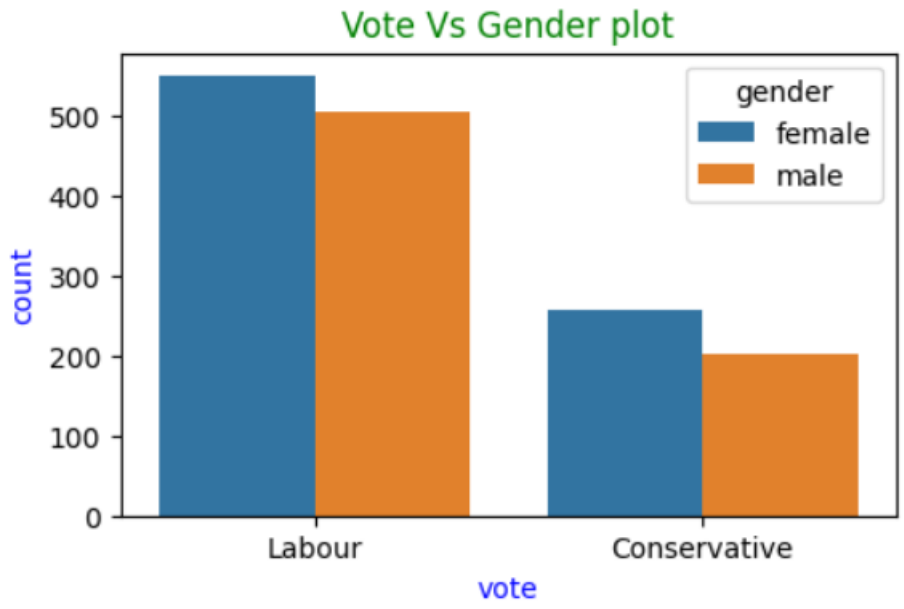


Fig-6 Count plot for Vote vs Gender

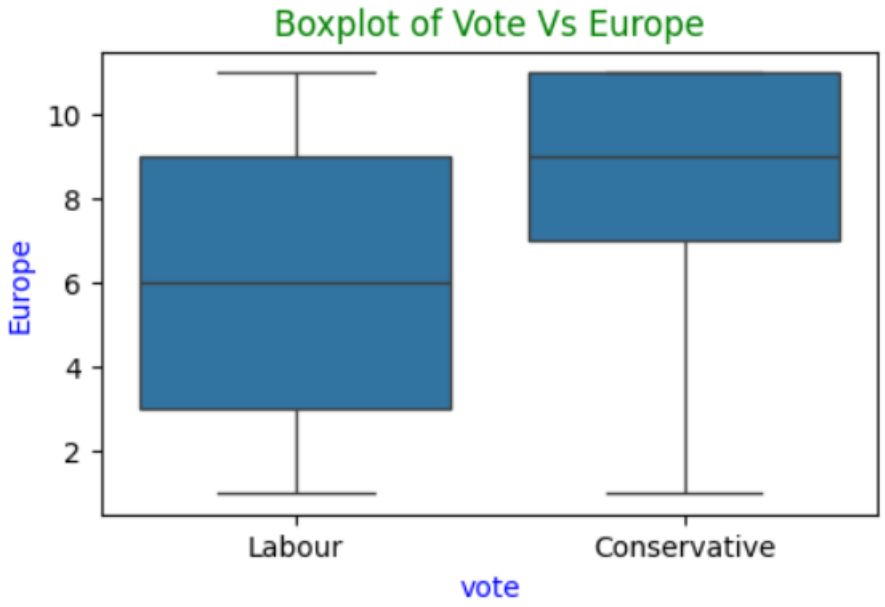


Fig-7 Boxplot for Vote vs Europe

Now we, we create a pair plot and a heatmap for numerical variables using seaborn library as shown below figure-8 and figure-9.

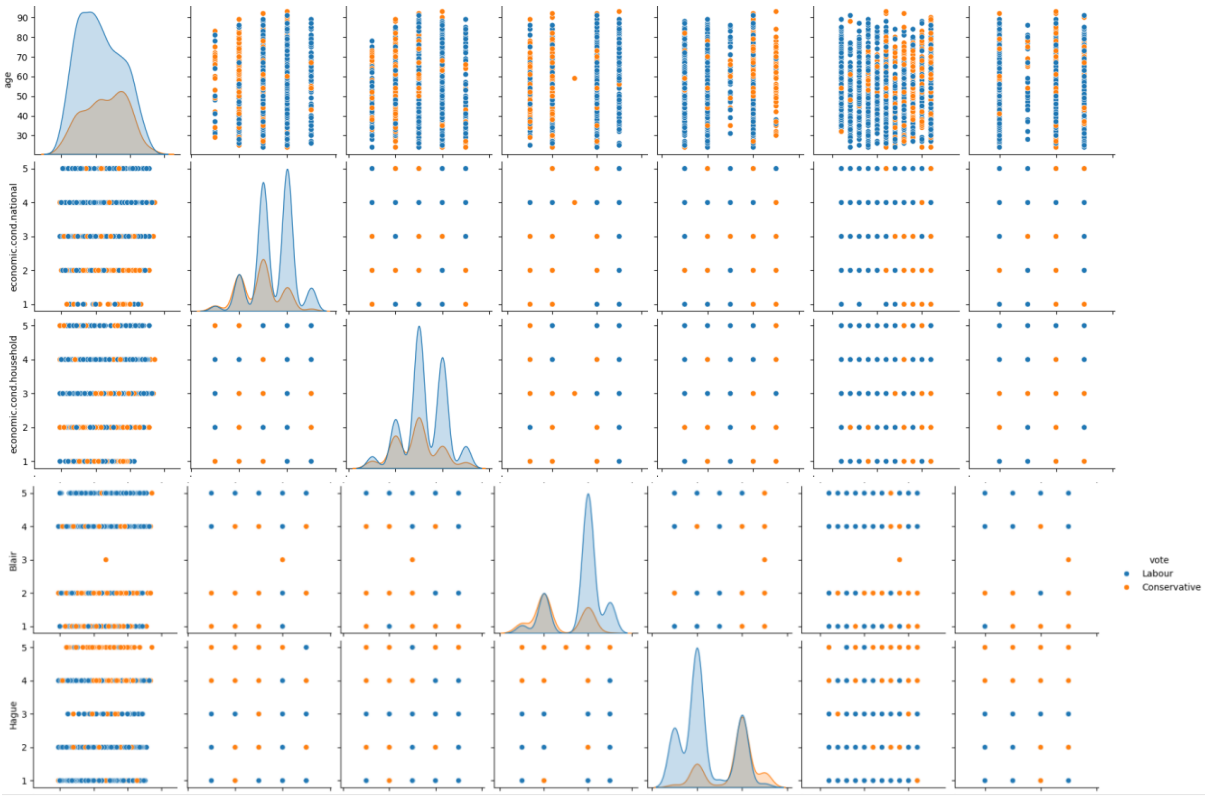


Fig-8 Pair plot for Numerical variable

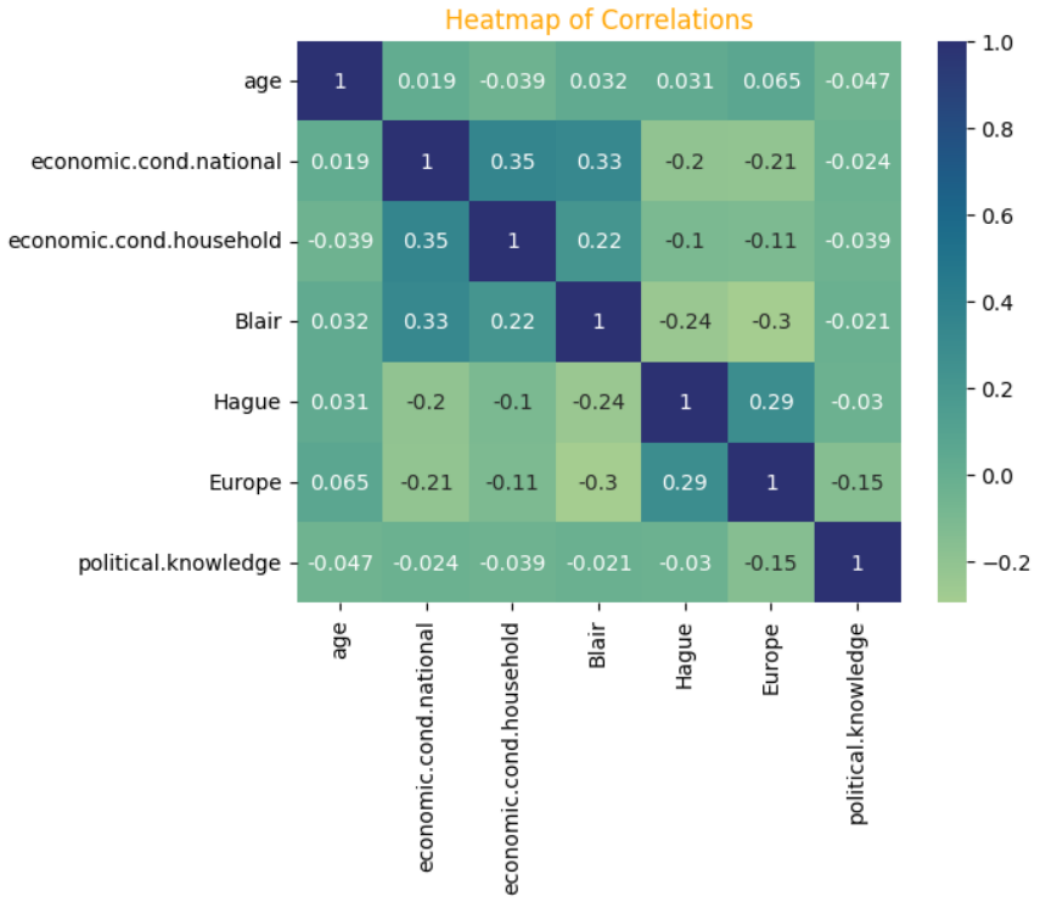


Fig-9 Heatmap for Numerical variable

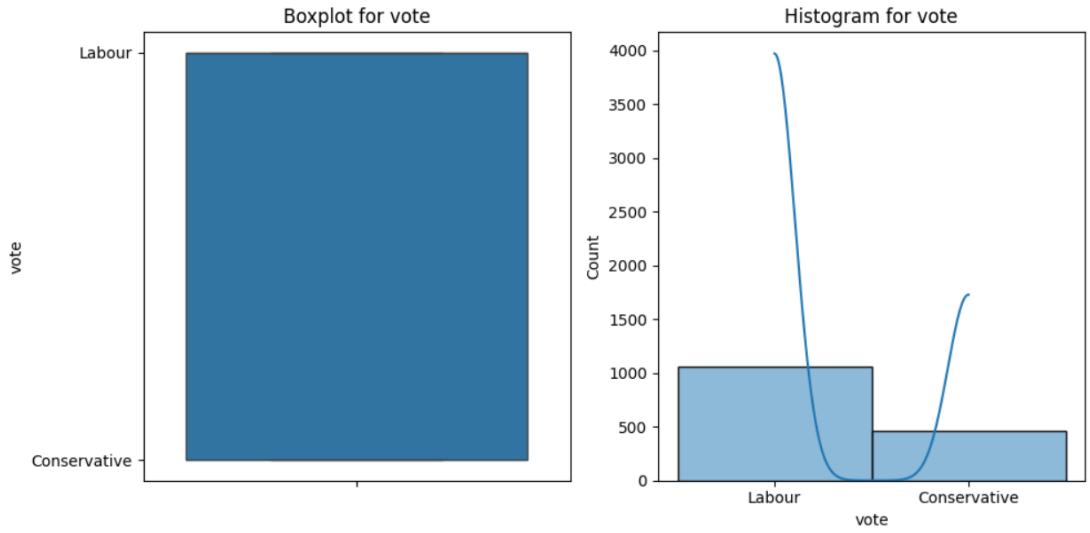
Key Meaningful Observations from Multivariate analysis:

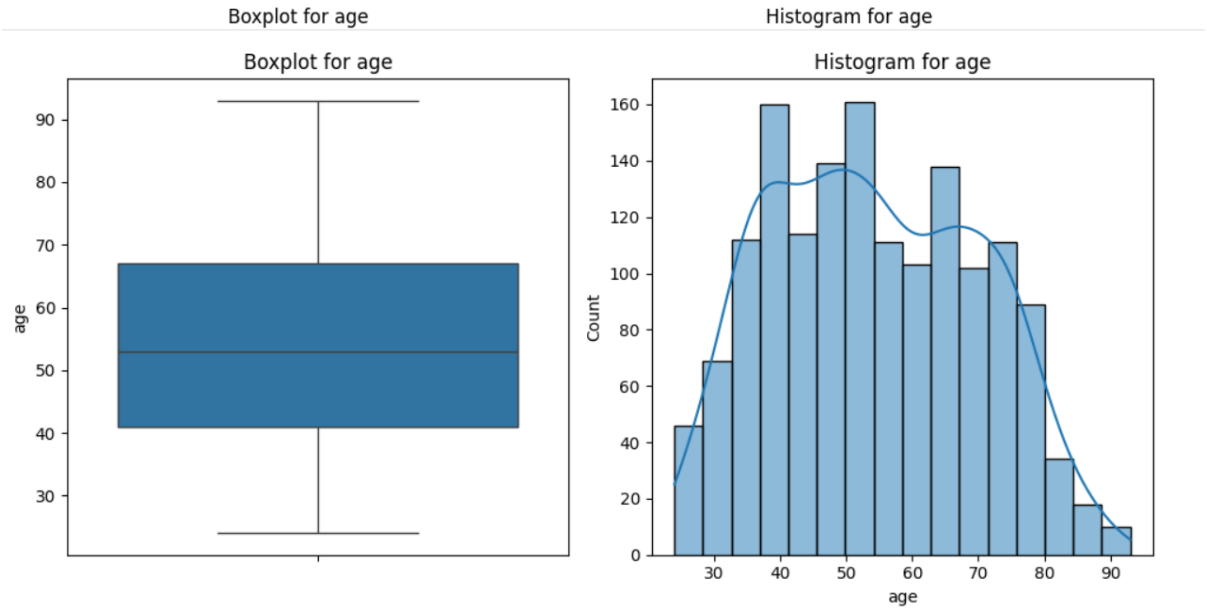
- Labour is the highest preferred party, accounting for approximately 70% of the votes.
- Conservative party support is around 30%.
- While this distribution doesn't necessarily indicate under sampling, a larger sample size could yield more robust results.
- The age variable follows an almost normal distribution with three minor peaks.
- No outliers or skewness are present, which is positive for analysis.
- The Labour leader receives the highest assessment at point 4 (just below the maximum of 5).
- The Conservative leader's highest assessment is at point 2 (above the minimum of 1), but they also receive significant assessments at point 4.
- Most voters express Eurosceptic sentiment, with scale 11 having the maximum count.
- The majority of voters lean toward Euroscepticism (scales above 5).
- Only 1.3% of voters have full knowledge of party positions on European integration.
- Approximately 30% lack knowledge in this area.
- Median age for Labour Party voters is around 50.
- Median age for Conservative Party voters is around 60.
- Labour skews slightly younger, while Conservative skews slightly older.

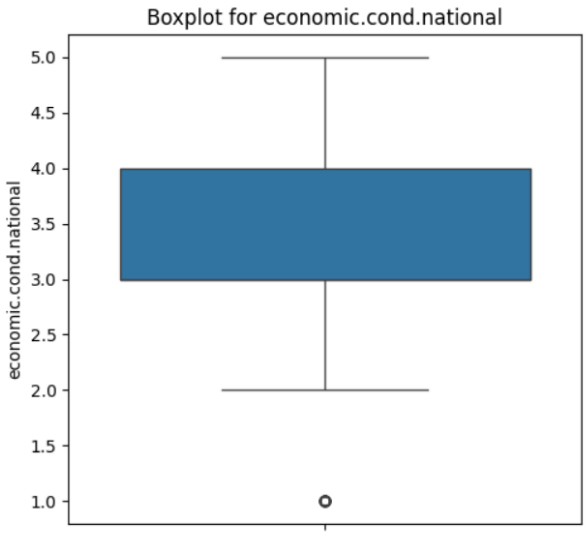
1.1.3 Data-preprocessing:

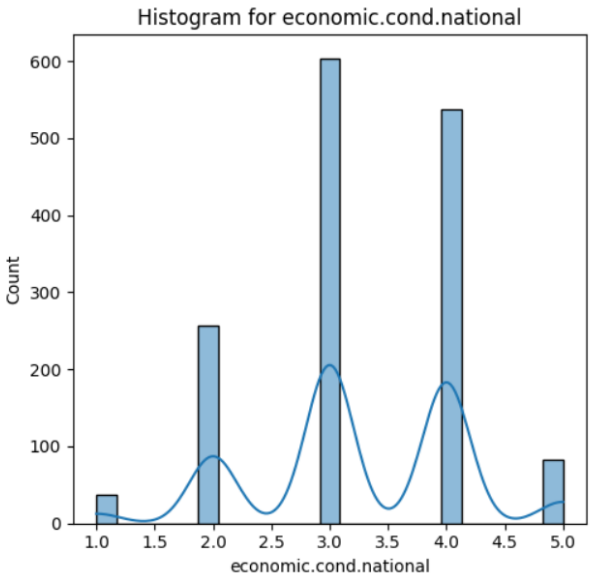
In this part we will perform outlier treatment, Data Encoding, Splitting the data into train test sets, Data Scaling and explain why we scaled the dataset.

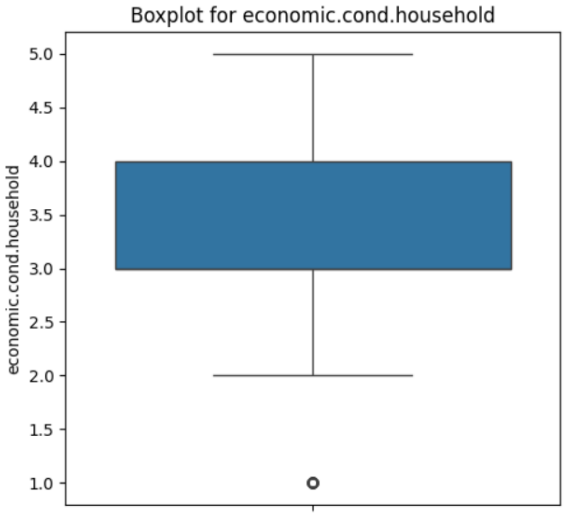
Now, as shown below figure we can see the that we have performed Univariate analysis on the numerical dataset that has presence of outliers.

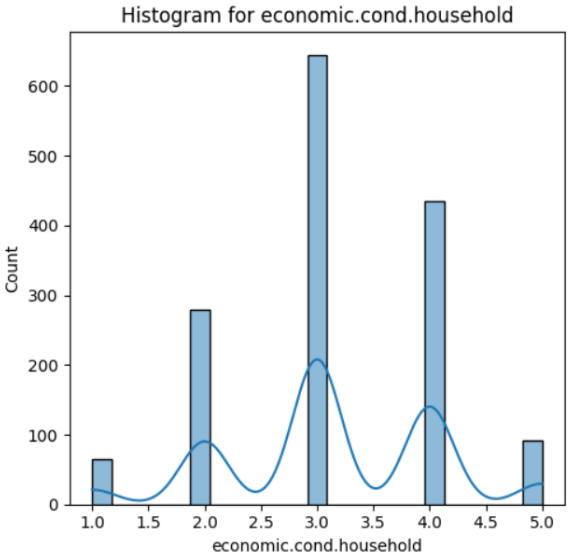


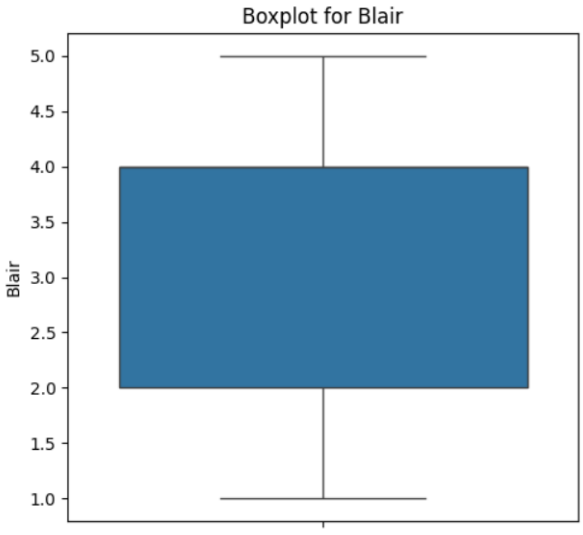


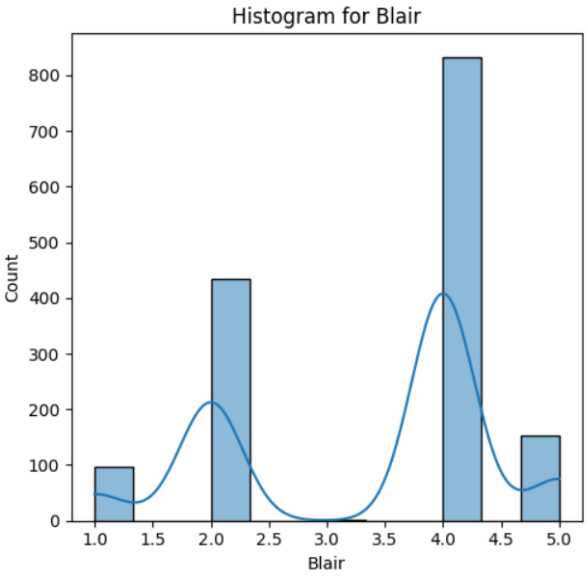

Boxplot for economic.cond.national
economic.cond.national
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0

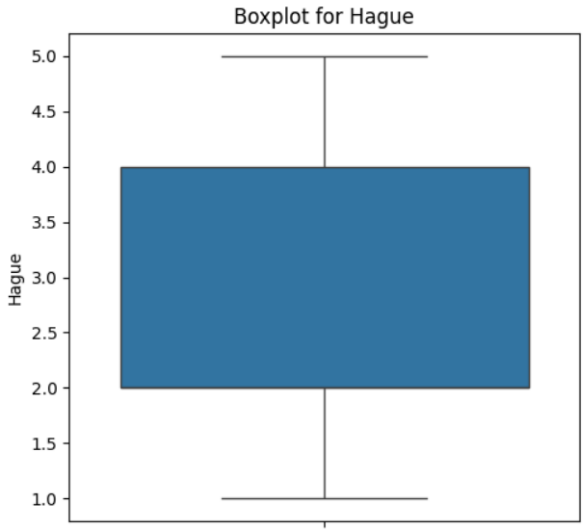

Histogram for economic.cond.national
Count
600
500
400
300
200
100
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
economic.cond.national

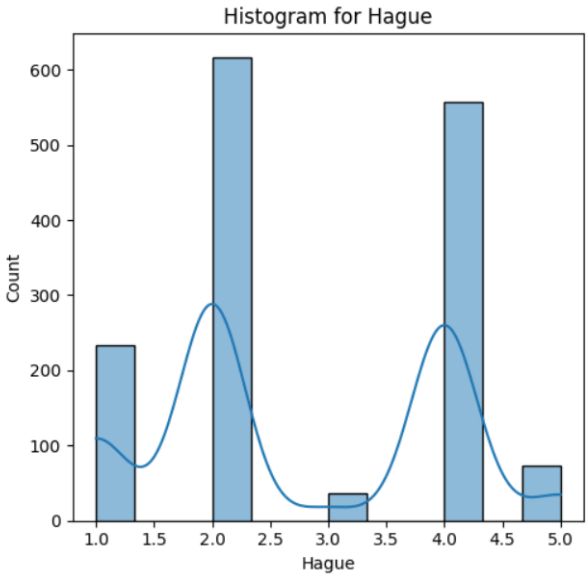

Boxplot for economic.cond.household
economic.cond.household
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0

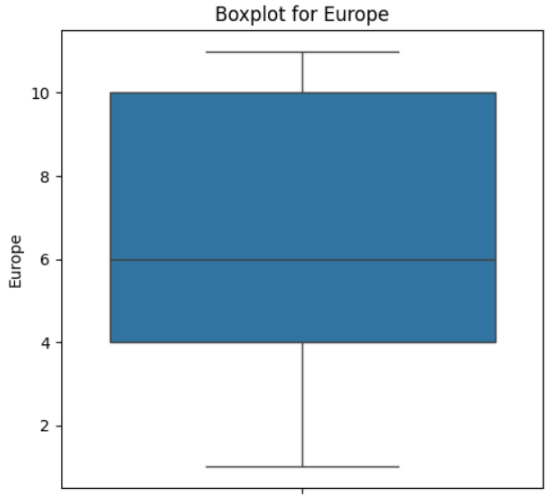

Histogram for economic.cond.household
Count
600
500
400
300
200
100
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
economic.cond.household

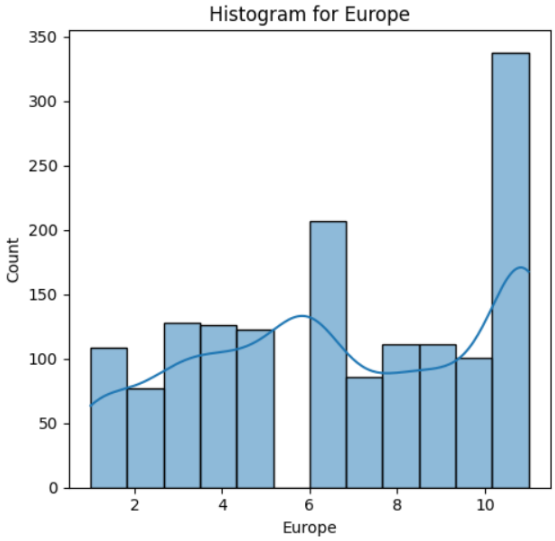

Boxplot for Blair
Blair
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0


Histogram for Blair
Count
800
700
600
500
400
300
200
100
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Blair


Boxplot for Hague
Hague
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0


Histogram for Hague
Count
600
500
400
300
200
100
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Hague


Boxplot for Europe
Europe
10
8
6
4
2


Histogram for Europe
Count
350
300
250
200
150
100
50
0
2
4
6
8
10
Europe

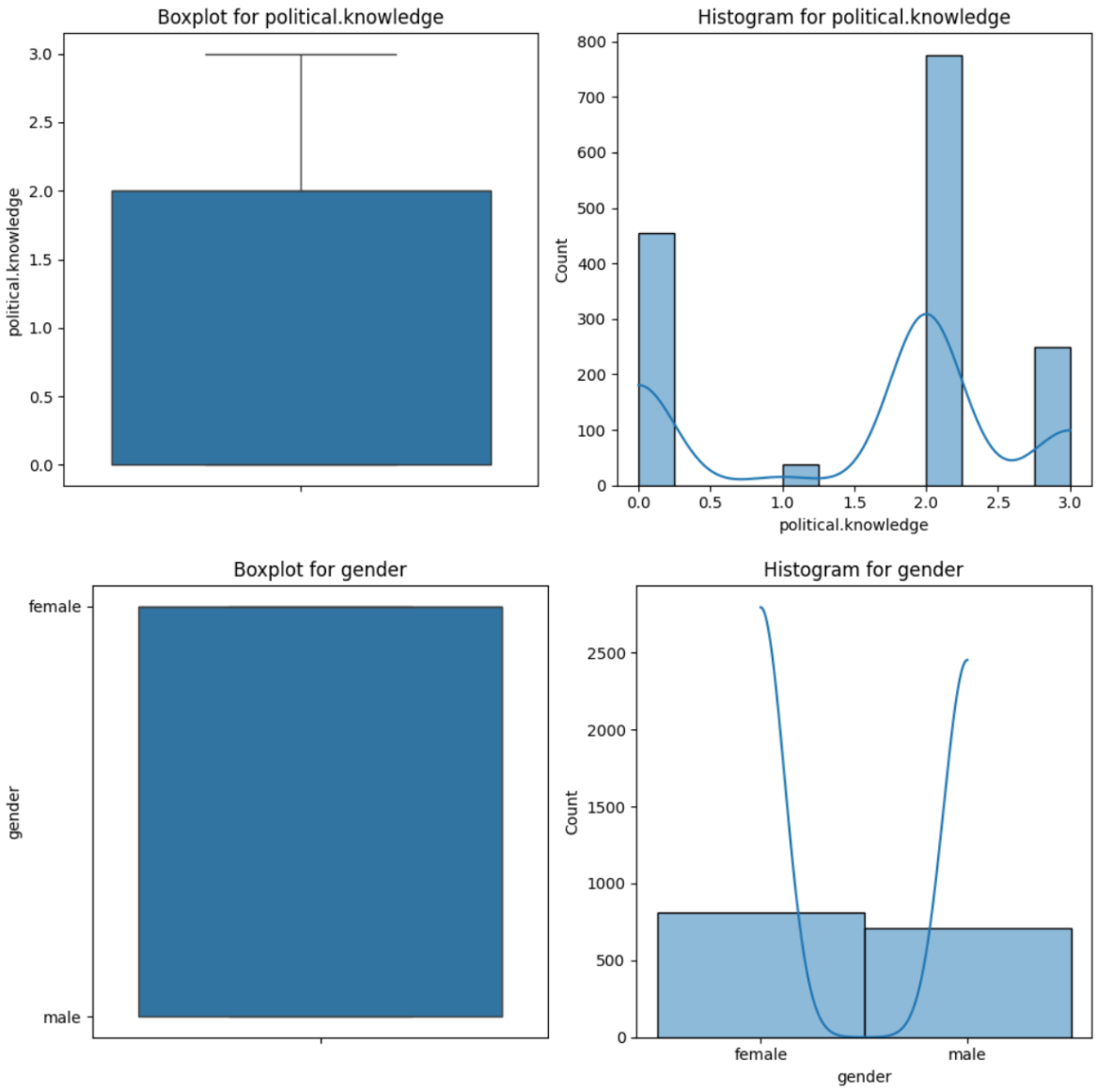
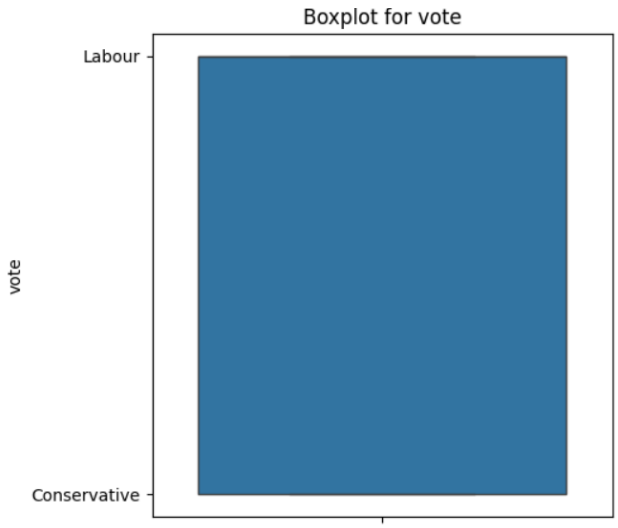
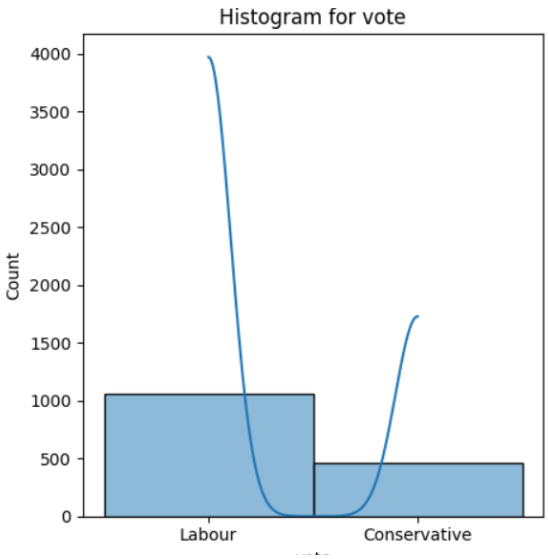
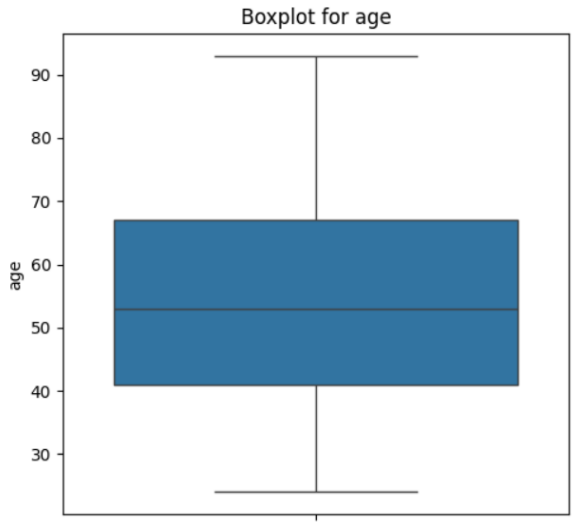


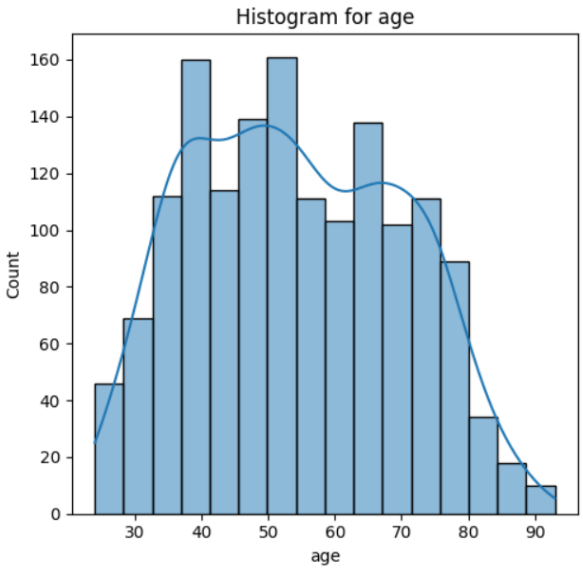
Fig-10 Boxplot Before Outlier Treatment

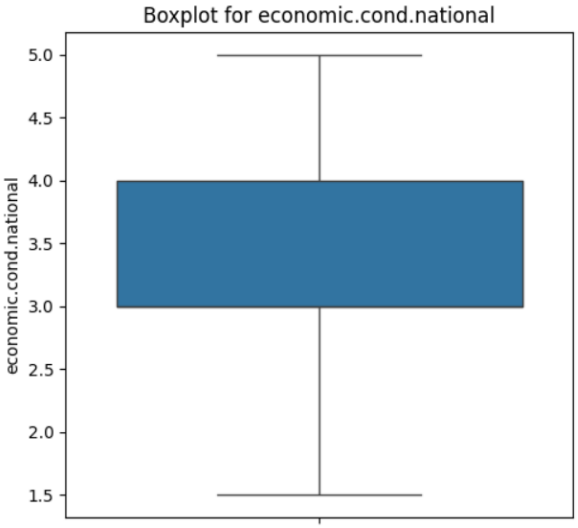
Now, we will perform outlier treatment using q1 and q3 to find IQR and using that to find out the upper and lower limit whiskers and finally bring all those outlier's points to these whiskers and obtain the below boxplots indicated that treatment has been done properly.

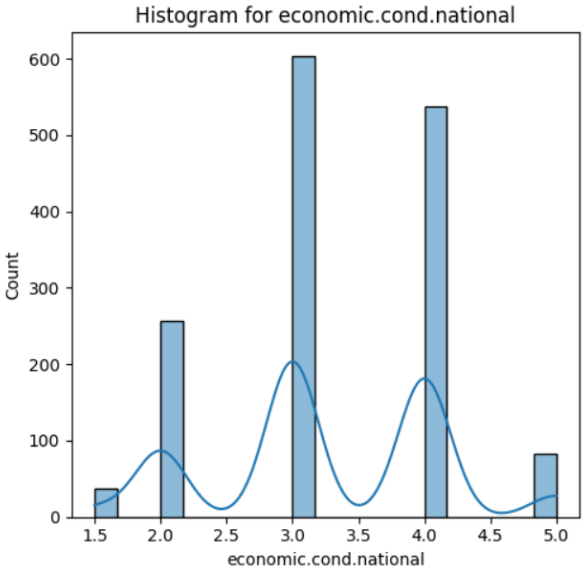

Boxplot for vote
Labour
Conservative
vote

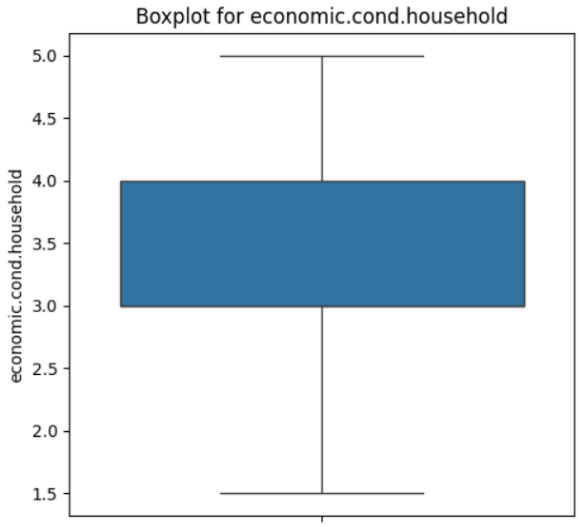

Histogram for vote
Count
Labour
Conservative

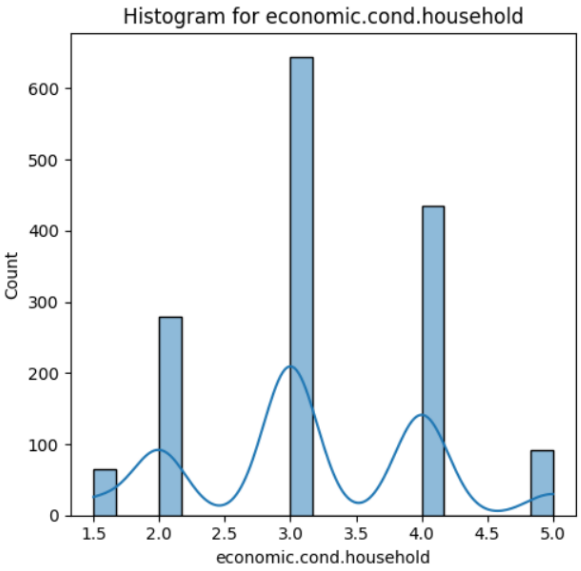

Boxplot for age
age

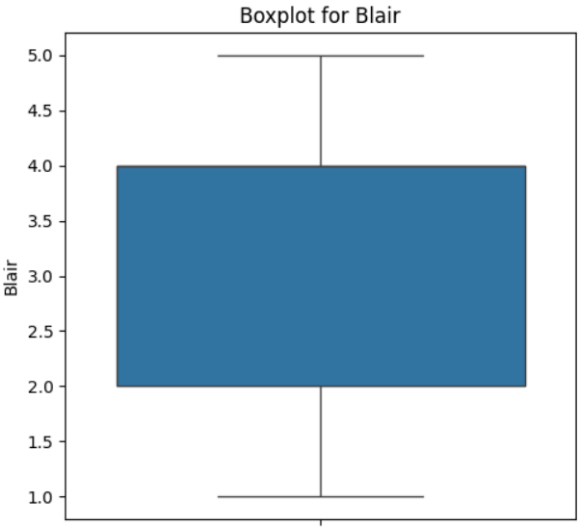

Histogram for age
Count
age

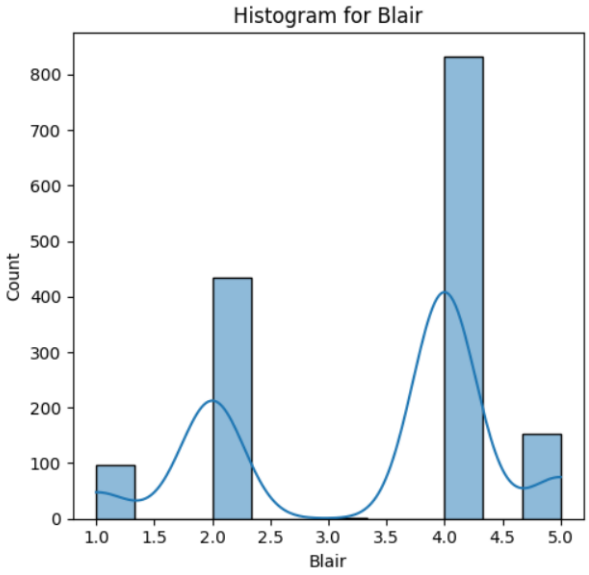

Boxplot for economic.cond.national
economic.cond.national

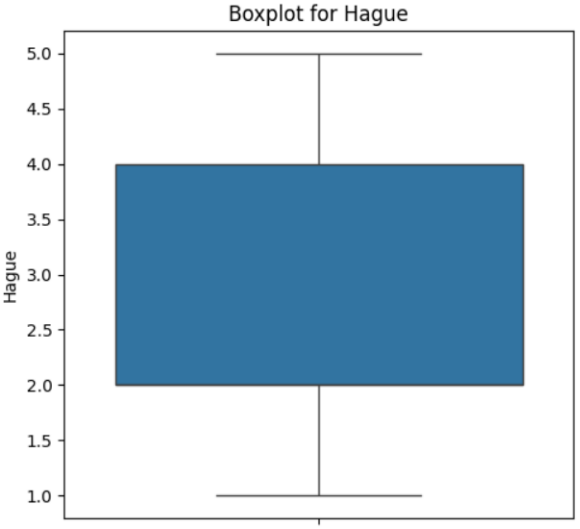

Histogram for economic.cond.national
Count
economic.cond.national

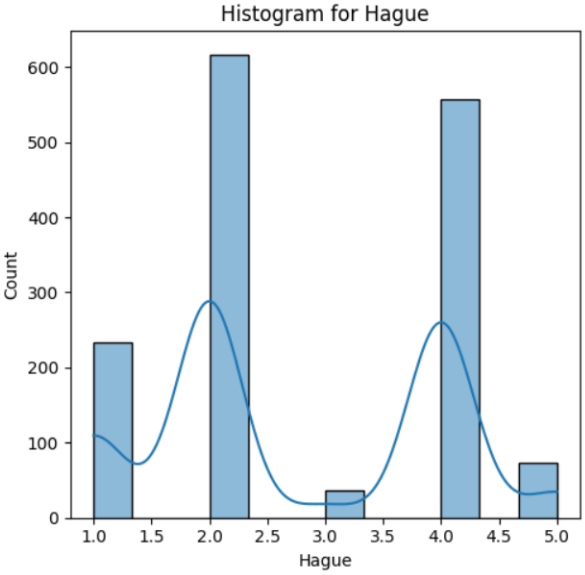

Boxplot for economic.cond.household
economic.cond.household
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5


Histogram for economic.cond.household
Count
600
500
400
300
200
100
0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
economic.cond.household


Boxplot for Blair
Blair
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0


Histogram for Blair
Count
800
700
600
500
400
300
200
100
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Blair


Boxplot for Hague
Hague
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0


Histogram for Hague
Count
600
500
400
300
200
100
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Hague

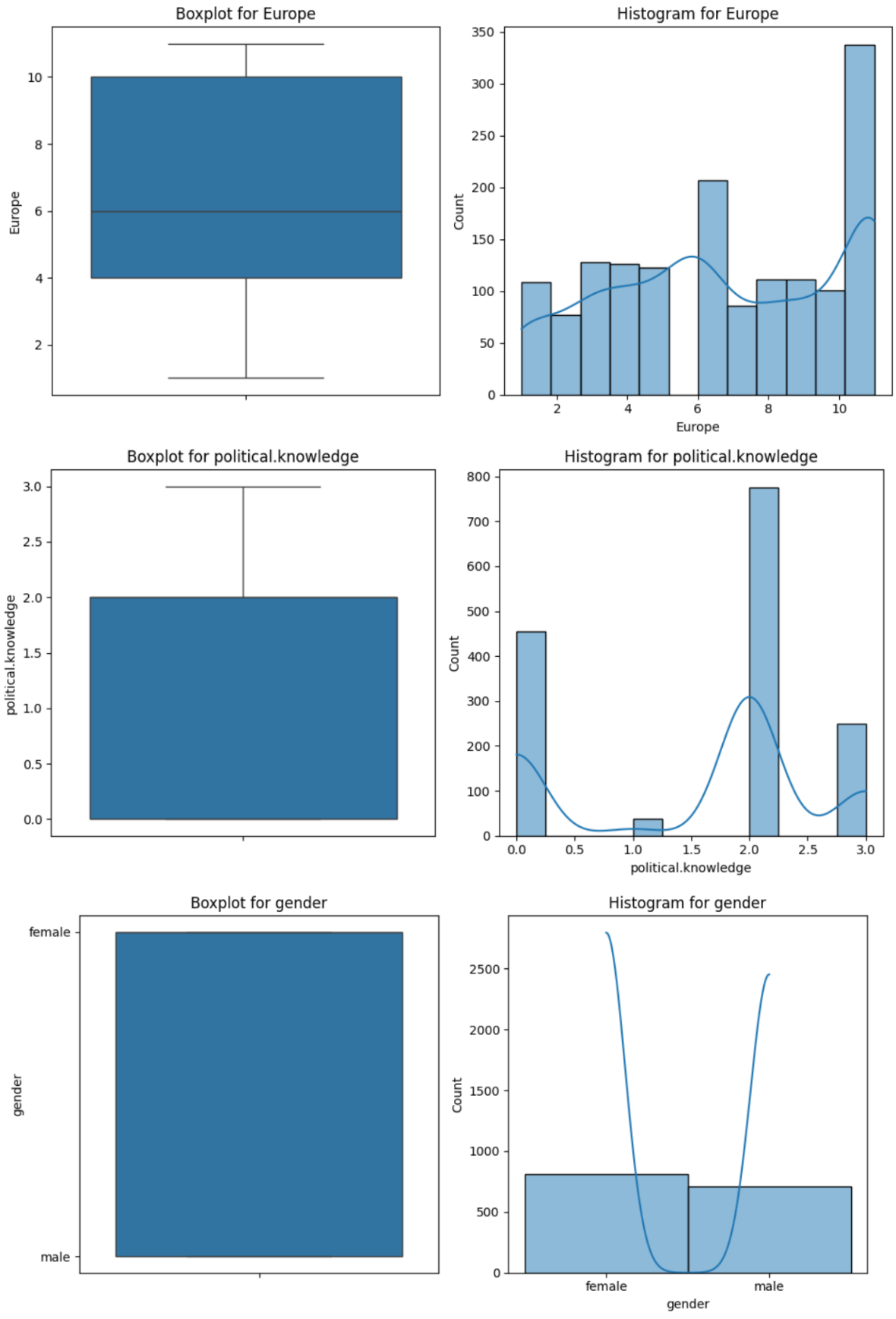


Fig-11 Boxplot After Outlier Treatment

Now, moving on next part we will do label encoding on vote and gender variables and we obtain the Table-7 as seen below.

	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	0	43.0	3.0	3.0	4.0	1.0	2.0	2.0	1
1	0	36.0	4.0	4.0	4.0	4.0	5.0	2.0	0
2	0	35.0	4.0	4.0	5.0	2.0	3.0	2.0	0
3	0	24.0	4.0	2.0	2.0	1.0	4.0	0.0	1
4	0	41.0	2.0	2.0	1.0	1.0	6.0	2.0	0

Table-7 Encoded dataset

Now, we will divide our final dataset into independent and dependent variables and save it in "X" for independent variables and "y" for dependent variable. Here our dependent variable is "vote" as, we are predicting the exit poll for election. After that we will split our this two dataset into training and testing dataset using train_test_split function where, 70% of data will be in training dataset and 30% will be in testing dataset with random state as 1.

Is Scaling Necessary:

- Scaling is a necessity when using Distance-based models such as KNN. Scaling can be done on continuous and ordinal variables.
- Scaling is performed when we are dealing with Gradient Descent Based algorithms (Linear and Logistic Regression, Neural Network) and Distance-based algorithms (KNN, K-means, SVM) as these are very sensitive to the range of the data points.
- Here we need scaled data only for KNN model and it is not required to scale the data for all other model but there is no harm in performing all the model on scaled data, hence proceeding with scaled data.
- After scaling the data is transformed having mean as 0 and standard deviation as 1.
- we will try to build the model using both scaled and unscaled data to see what difference it makes.
- Scaling is to be done After Train Test split.

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	1.085923	-0.313103	-0.196335	-1.161925	-0.593283	0.411542	0.452231	0.936950
1	-0.717151	-0.313103	-1.316288	0.550300	-0.593283	0.411542	1.382110	-1.067292
2	2.245042	2.098531	2.043572	1.406413	-0.593283	-1.736401	0.452231	-1.067292
3	-0.459569	-1.518921	-0.196335	-1.161925	1.029070	0.411542	0.452231	0.936950
4	-1.361106	2.098531	0.923619	0.550300	1.029070	0.411542	-1.407526	-1.067292

Table-8 first five rows of scaled training dataset.

1.2 Model Building, Model tuning and Performance Evaluation:

We will build model using KNN, Naive baye, CARTs Method, Bagging, Boosting. Then we will do model tuning to improve their performance and then check their performance using confusion

matrix, Classification matrix and AUC-ROC curves. our main metrics of choice would be F1, Precision, Recall and Confusion Matrix. Finally, compare all the models a table will be created and select the best suitable one from all the models and give insights.

1.2.1 K Nearest Neighbors Method:

For this method we will use Euclidean distance method and k value from 3 to 9 for unscaled train and test sets and scaled train and test sets, we obtain the accuracy scores for both as shown below and can say that we for KNN we will be using only scaled dataset for further improvement as shown in below Table-9 and Table-10.

```
For K = 3 ------->
Accuracy Score for Training Data is:  0.8699340245051838
Accuracy Score for Test Data is:  0.7850877192982456

For K = 5 ------->
Accuracy Score for Training Data is:  0.8605089538171536
Accuracy Score for Test Data is:  0.8223684210526315

For K = 7 ------->
Accuracy Score for Training Data is:  0.8463713477851084
Accuracy Score for Test Data is:  0.8223684210526315

For K = 9 ------->
Accuracy Score for Training Data is:  0.8482563619227145
Accuracy Score for Test Data is:  0.8135964912280702
```

Table-9 Unscaled Dataset KNN

```
For K = 3 ------->
Accuracy Score for Training Data is:  0.8718190386427899
Accuracy Score for Test Data is:  0.8048245614035088

For K = 5 ------->
Accuracy Score for Training Data is:  0.8548539114043355
Accuracy Score for Test Data is:  0.8245614035087719

For K = 7 ------->
Accuracy Score for Training Data is:  0.8539114043355325
Accuracy Score for Test Data is:  0.8179824561403509

For K = 9 ------->
Accuracy Score for Training Data is:  0.8444863336475024
Accuracy Score for Test Data is:  0.8157894736842105
```

Table-10 Scaled Dataset KNN

After that we will grid search for KNN using Grid Search CV we obtain the best fit will be if we use Minkowski method for distance and K=7 and weights as uniform. Using a function called cv_results_ we obtained the mean test score in an array for as shown below:

```
array([0.75872862, 0.75872862, 0.78514371, 0.75872862, 0.80020279,
       0.79361665, 0.79455123, 0.78889085, 0.81057133, 0.80774114,
       0.80679774, 0.80584553, 0.81622289, 0.81150591, 0.80679774,
       0.8124493 , 0.80773232, 0.80772351, 0.75872862, 0.75872862,
       0.78514371, 0.75872862, 0.80020279, 0.79361665, 0.79455123,
       0.78889085, 0.81057133, 0.80774114, 0.80679774, 0.80584553,
       0.81622289, 0.81150591, 0.80679774, 0.8124493 , 0.80773232,
       0.80772351])
```

Before we create a model for best fit values, we will create a graph for misclassification error vs K value and we obtained the Fig-12 as shown below:

Fig-11 K-value vs Misclassification error

Hence, will see the accuracy score for K value 5,16, best fit. And they are as follows:

1> KNN with K=5, basic model:

Accuracy of Training Data: 0.8548539114043355

Accuracy of Test Data: 0.8245614035087719

	precision	recall	f1-score	support
0	0.88	0.92	0.90	754
1	0.77	0.70	0.74	307
accuracy			0.85	1061
macro avg	0.83	0.81	0.82	1061
weighted avg	0.85	0.85	0.85	1061

Table-11 Classification report for K=5 of train 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	0.81	0.80	0.80	456
weighted avg	0.82	0.82	0.82	456

Table-12 Classification report for K=5 of test set

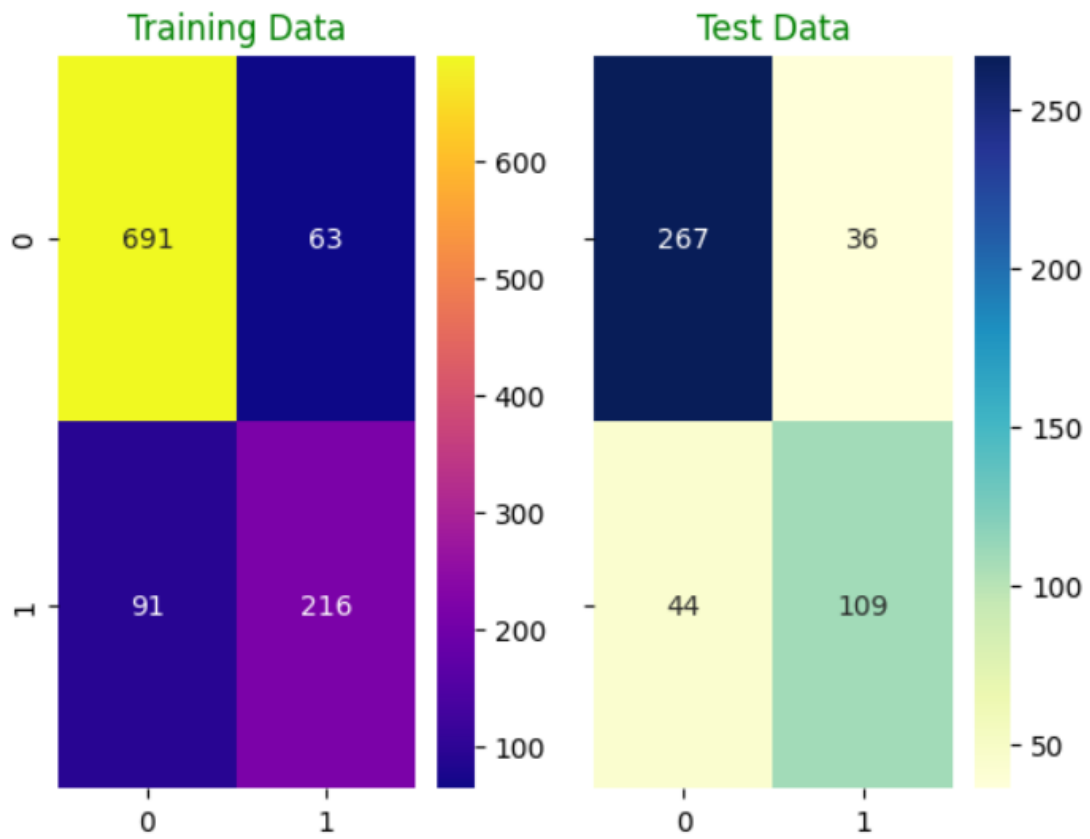
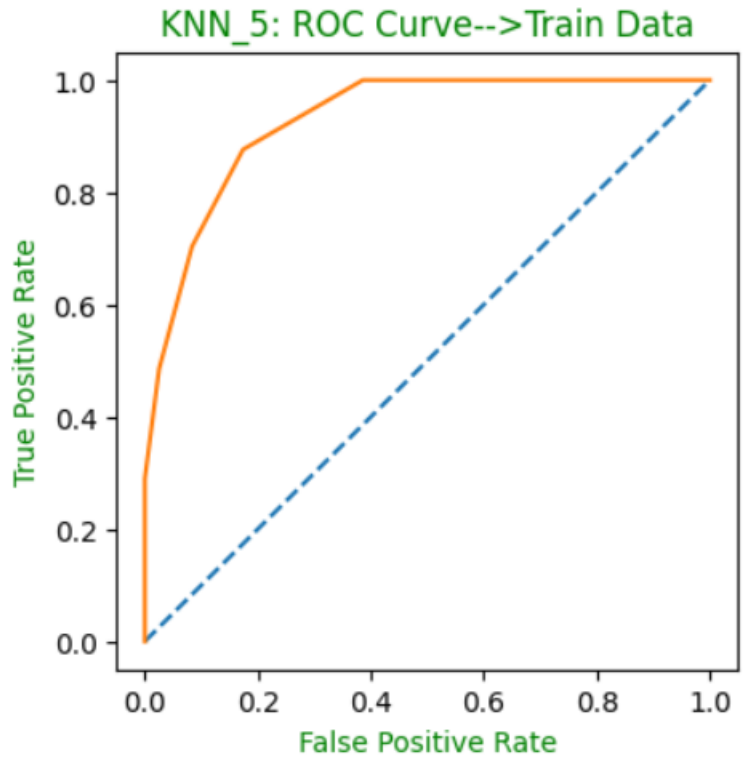


Fig-12 Confusion Matrix for K=5

```
AUC: 0.929
```




```
AUC: 0.866
```

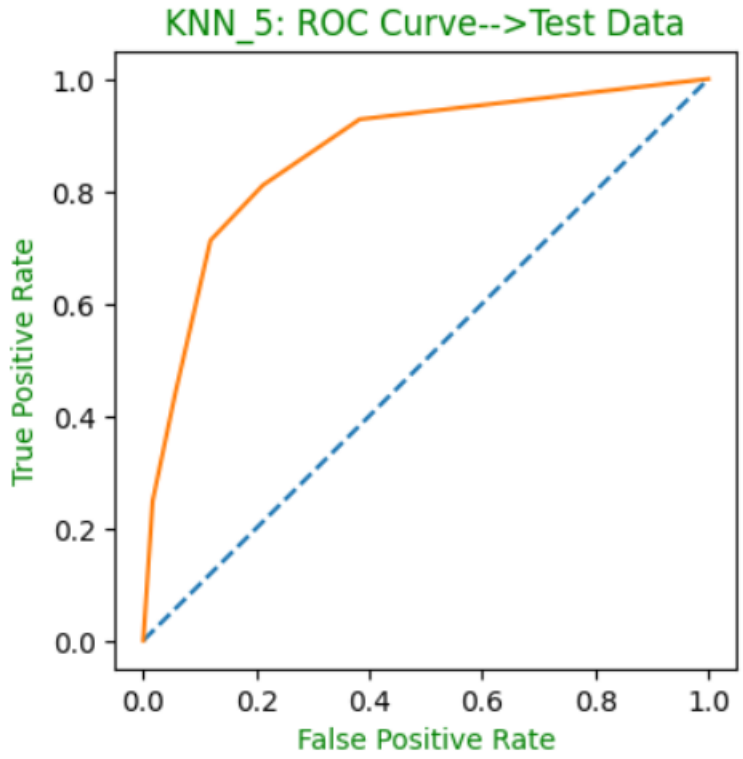
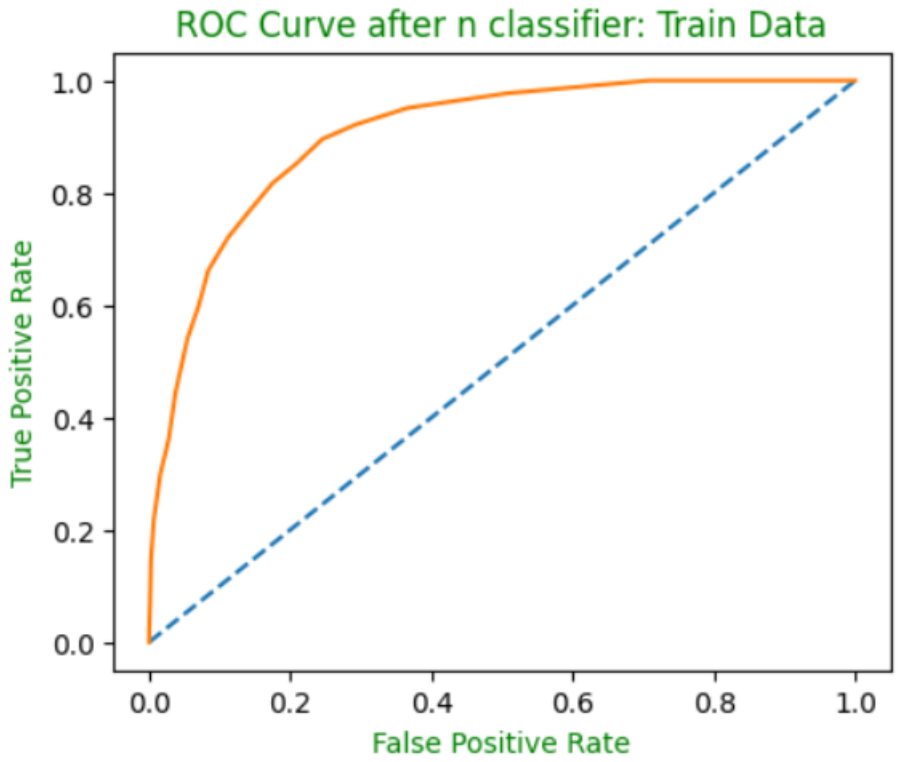


Fig-13 AUC-ROC graph for K=5

```
the auc curve 0.904
```




```
the auc curve 0.888
```

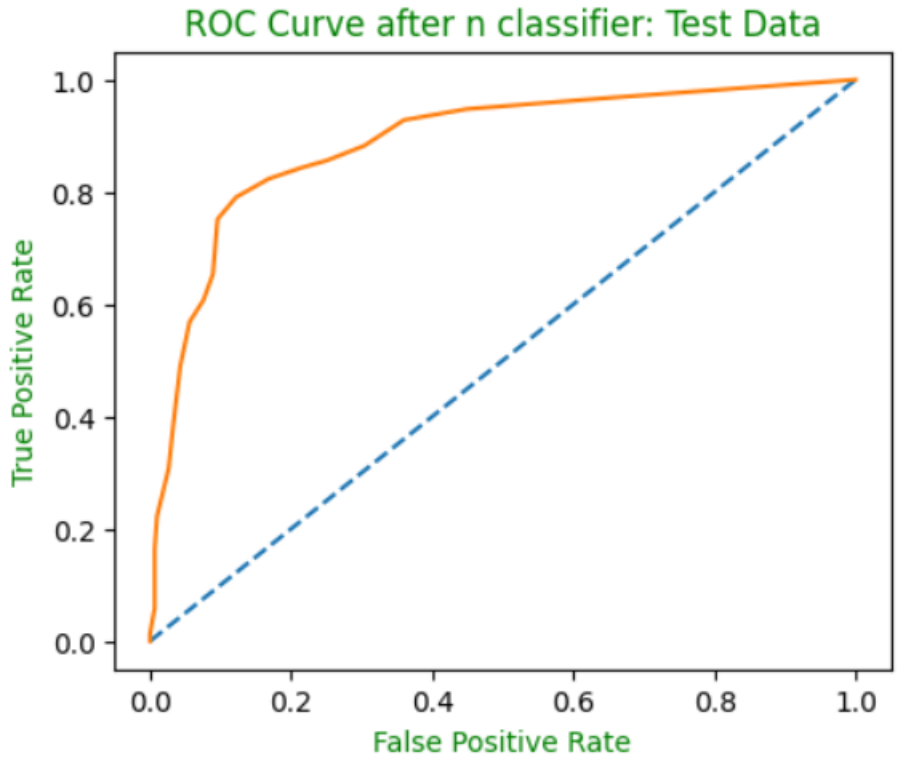


Fig-14 AUC-ROC graph after n-classifier

2> KNN for k = 16 , as it has lowest misclassification error :

	precision	recall	f1-score	support
0	0.86	0.92	0.89	754
1	0.76	0.63	0.69	307
accuracy			0.84	1061
macro avg	0.81	0.78	0.79	1061
weighted avg	0.83	0.84	0.83	1061

Table-13 Classification report for K=16 of train set

	precision	recall	f1-score	support
0	0.83	0.92	0.87	303
1	0.80	0.63	0.71	153
accuracy			0.82	456
macro avg	0.81	0.78	0.79	456
weighted avg	0.82	0.82	0.82	456

Table-14 Classification report for K=16 of test set

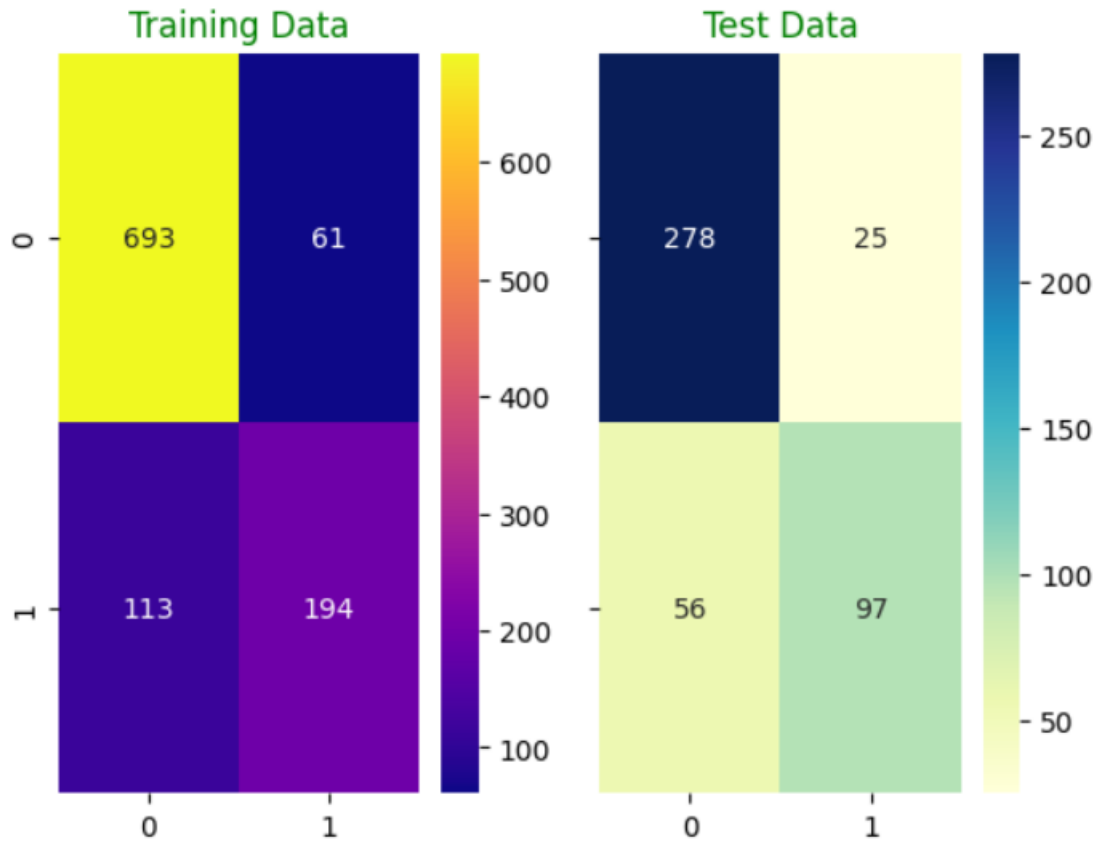
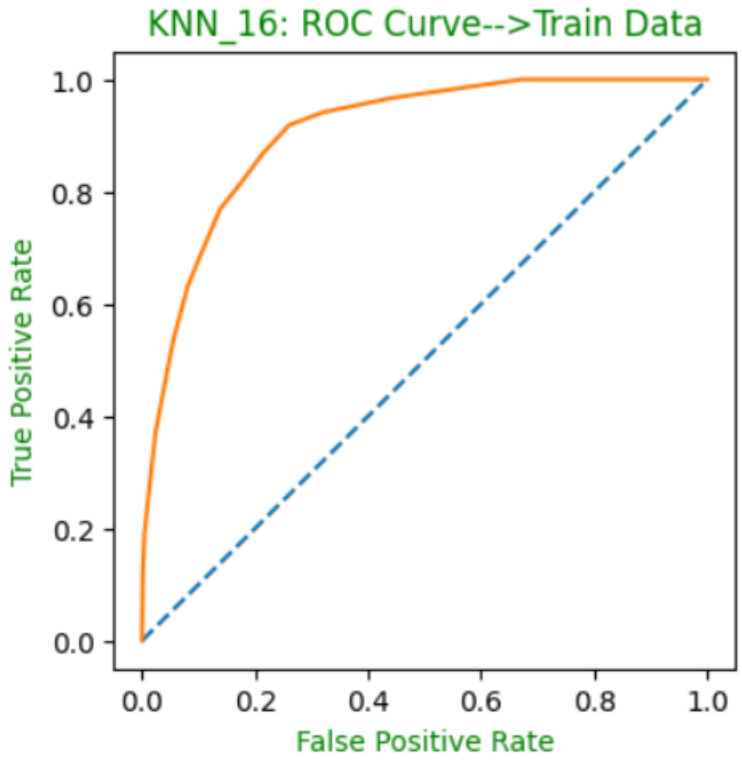


Fig-15 Confusion Matrix for K=16

```
AUC: 0.905
```




```
AUC: 0.887
```

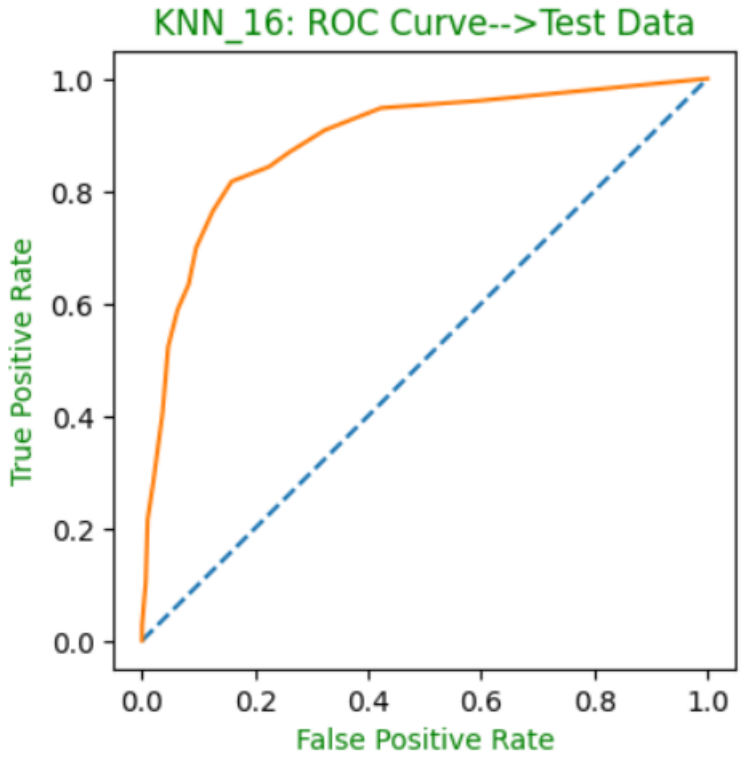


Fig-16 AUC-ROC graph for K=16

3> KNN after Model Tuning:

	precision	recall	f1-score	support
0	0.89	0.91	0.90	754
1	0.77	0.71	0.74	307
accuracy			0.85	1061
macro avg	0.83	0.81	0.82	1061
weighted avg	0.85	0.85	0.85	1061

Table-14 Classification report for tunned KNN of train set

	precision	recall	f1-score	support
0	0.86	0.87	0.86	303
1	0.73	0.72	0.73	153
accuracy			0.82	456
macro avg	0.80	0.79	0.79	456
weighted avg	0.82	0.82	0.82	456

Table-15 Classification report for tunned KNN test set

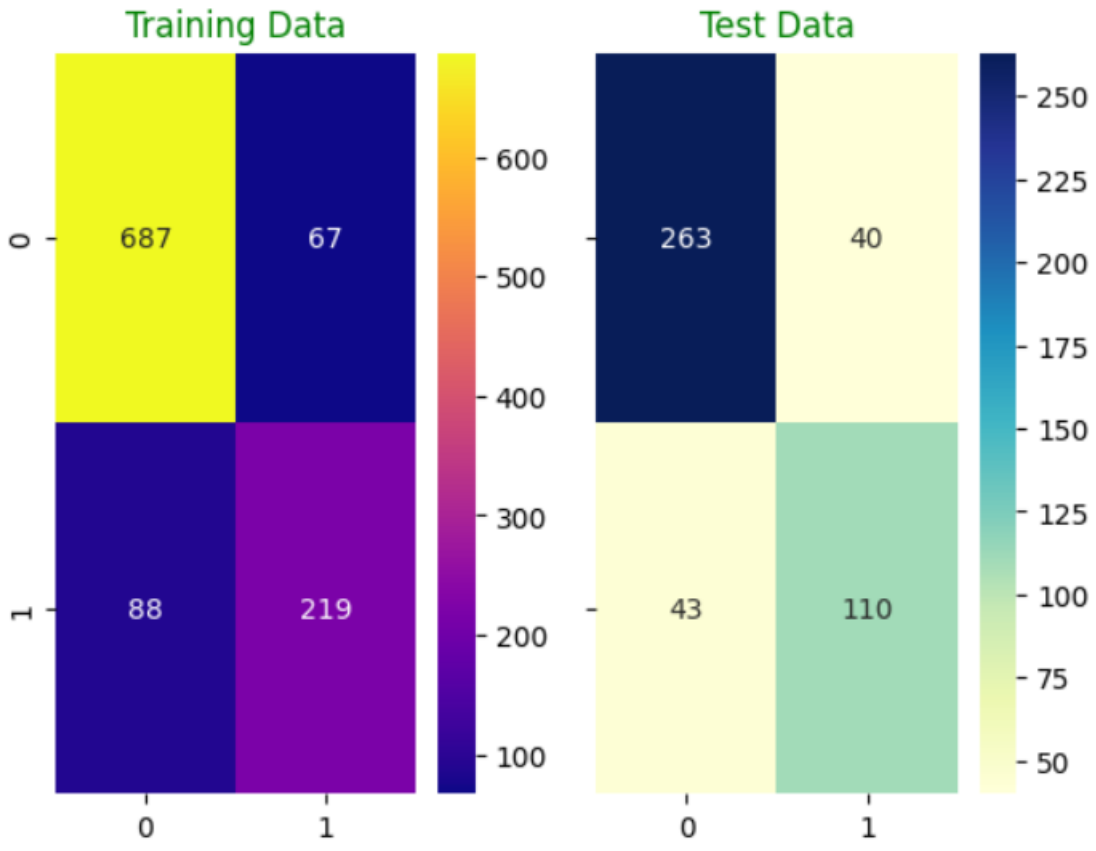
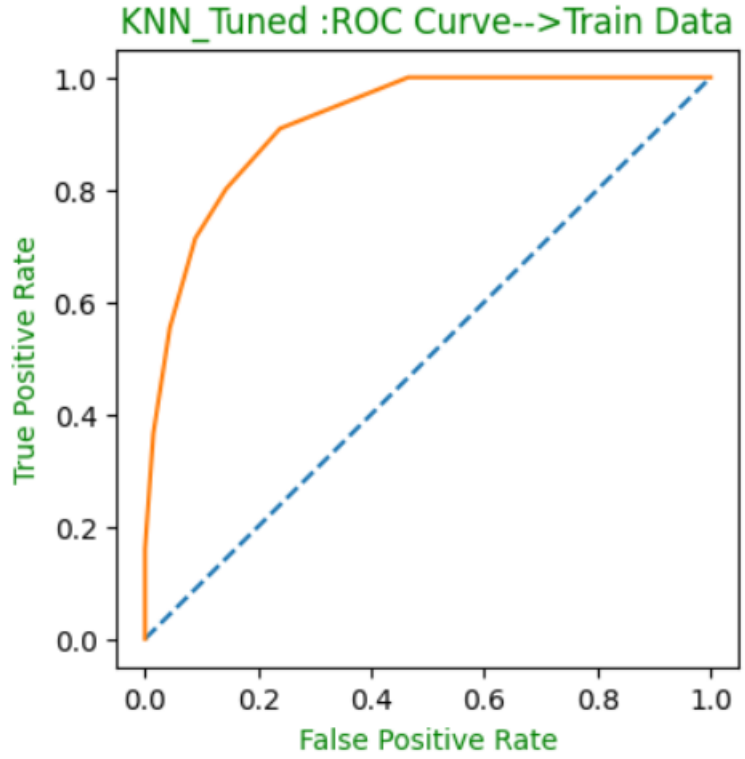


Fig-17 Confusion Matrix for tunned KNN

AUC: 0.920



AUC: 0.881

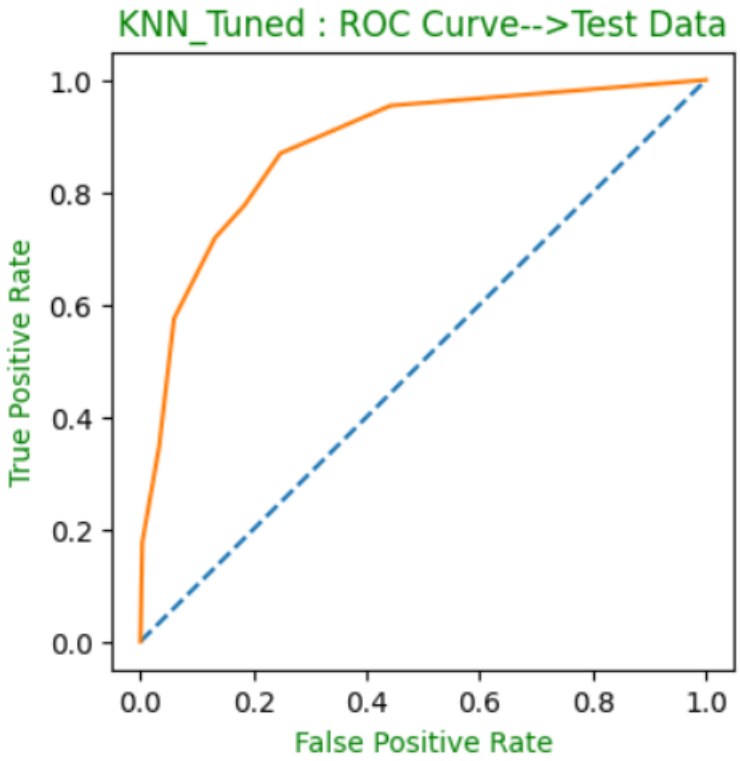


Fig-18 AUC-ROC graph for tunned KNN

1.2.2 Gaussian Naive Bayes Method:

We will fit the training set into GaussianNB algorithm and we obtained the Accuracy scores:

Accuracy Score for Training Data is: 0.8341187558906692
Accuracy Score for Test Data is: 0.8223684210526315

	precision	recall	f1-score	support
0	0.88	0.89	0.88	754
1	0.72	0.69	0.71	307
accuracy			0.83	1061
macro avg	0.80	0.79	0.80	1061
weighted avg	0.83	0.83	0.83	1061

Table-16 Classification report for Navie Bayes of train set

	precision	recall	f1-score	support
0	0.87	0.87	0.87	303
1	0.74	0.73	0.73	153
accuracy			0.82	456
macro avg	0.80	0.80	0.80	456
weighted avg	0.82	0.82	0.82	456

Table-17 Classification report for Navie Bayes of test set

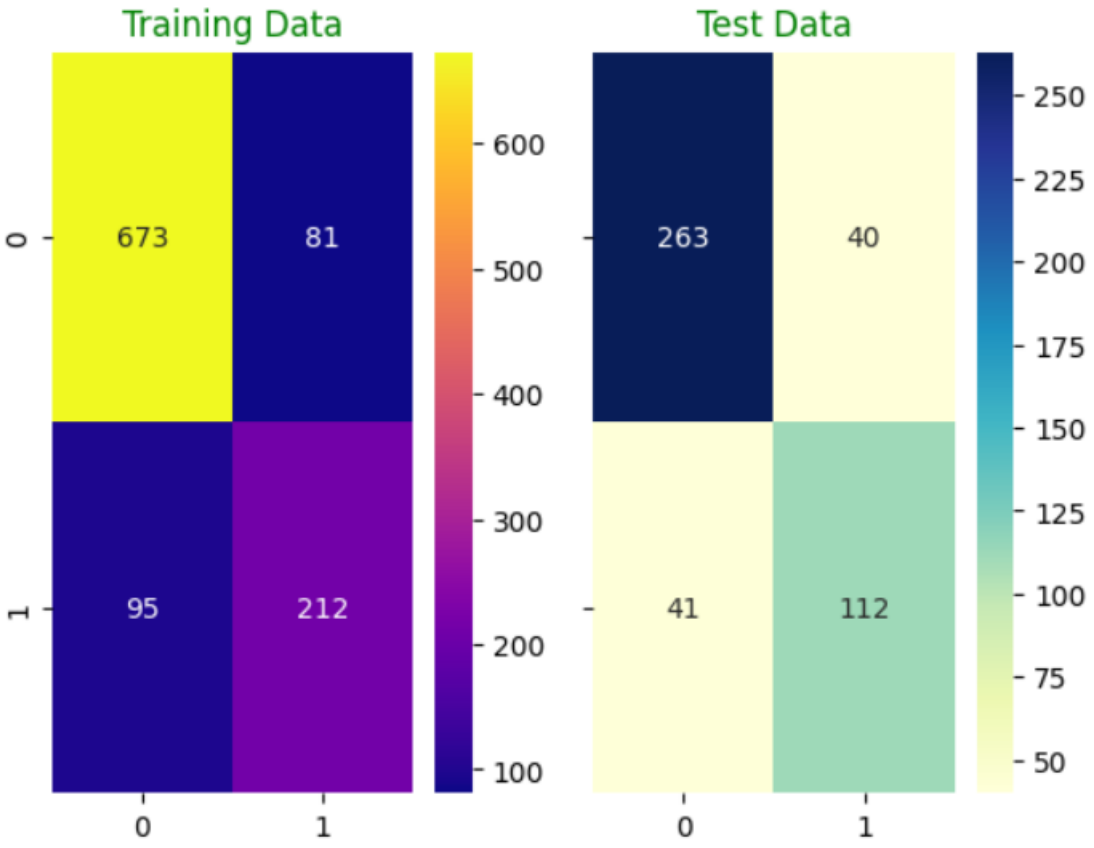


Fig-19 Confusion Matrix for Navie Bayes

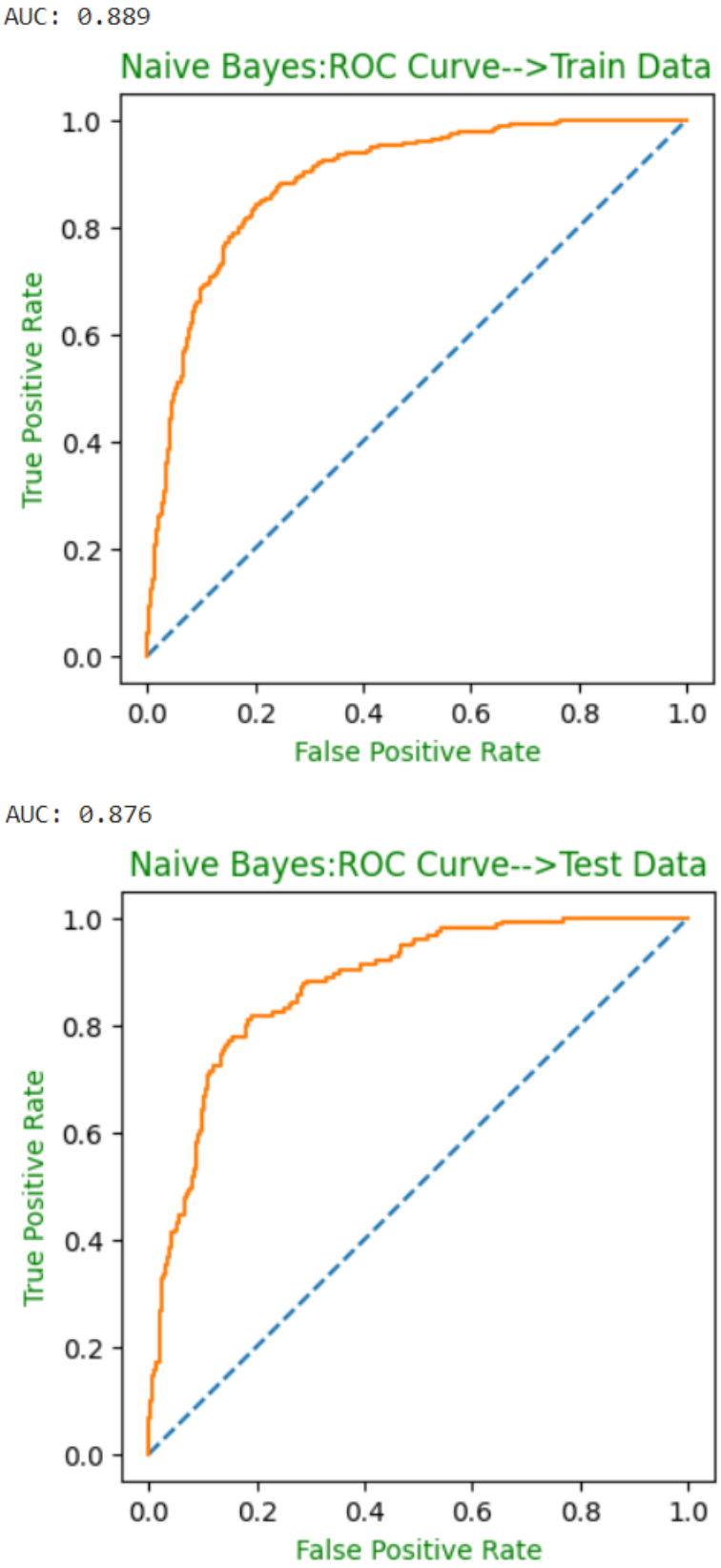


Fig-20 AUC-ROC graph for Navie Bayes

The below table shows the comparison of KNN and Naive Bayes method Train and Test scores:

	KNN	Naive_Bayes
Train_Accuracy	0.854854	0.834119
Test_Accuracy	0.824561	0.822368

Table-18 Comparision between KNN and Navie Bayes

1.2.3 Bagging Method:

For, this method we have created a Decision Tree that is an overfit and will be used as a base estimator for the Bagging Method.

The accuracy scores obtained from them are:

Accuracy on Train data: 0.9509896324222432

Accuracy on Test Data: 0.8289473684210527

	precision	recall	f1-score	support
0	0.95	0.99	0.97	754
1	0.96	0.87	0.91	307
accuracy			0.95	1061
macro avg	0.95	0.93	0.94	1061
weighted avg	0.95	0.95	0.95	1061

Table-19 Classification report for Bagging of train set

	precision	recall	f1-score	support
0	0.85	0.90	0.88	303
1	0.78	0.69	0.73	153
accuracy			0.83	456
macro avg	0.81	0.79	0.80	456
weighted avg	0.83	0.83	0.83	456

Table-20 Classification report for Bagging of test set

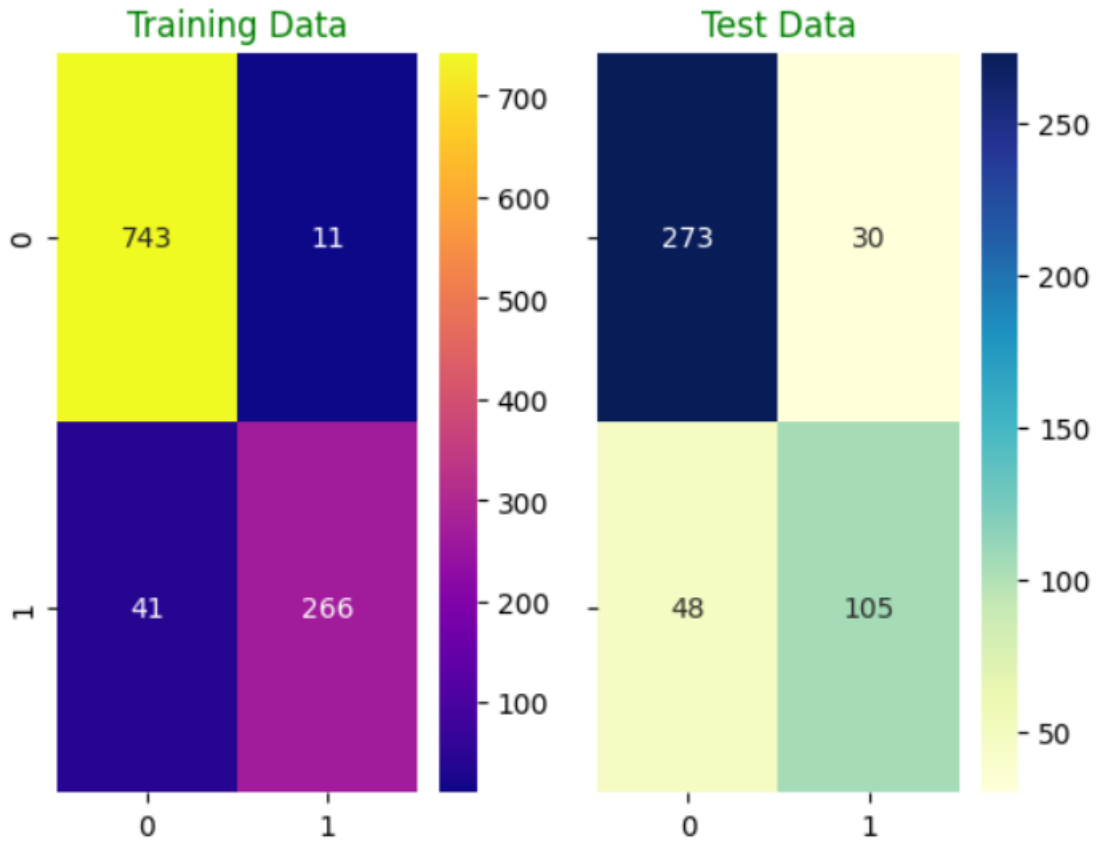
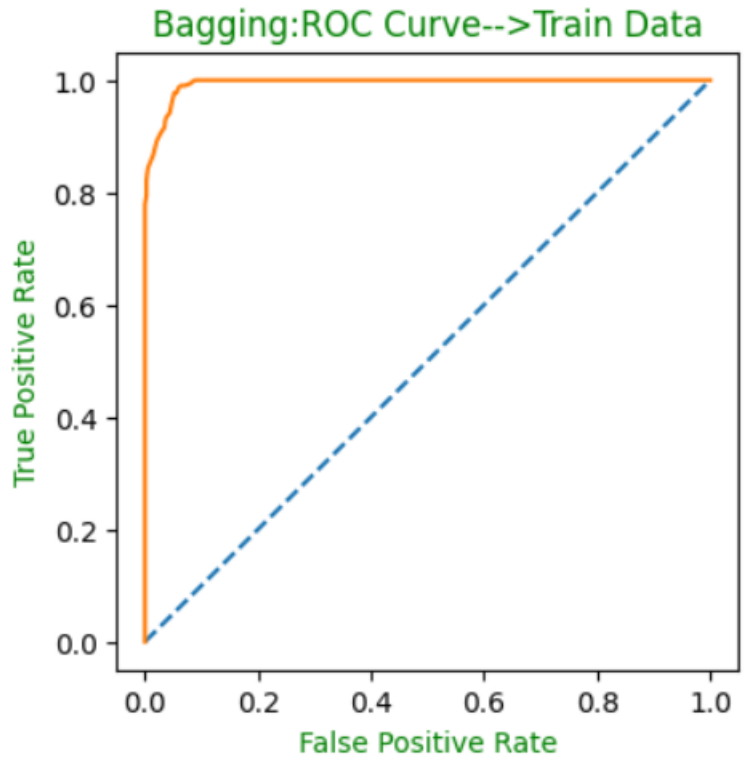


Fig-21 Confusion Matrix for Bagging

AUC: 0.994



AUC: 0.890

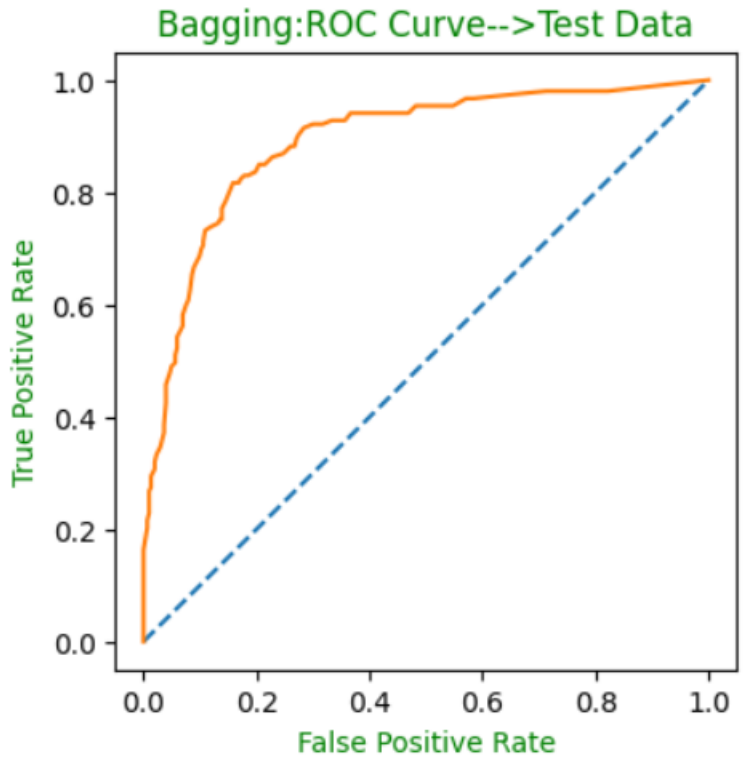


Fig-22 AUC-ROC graph for Bagging

1.2.4 Ada-Boosting Method:

Now, we will see the values for accuracy score before tuning and after tuning.

Before grid search:

Accuracy on Train Data: 0.8501413760603205

Accuracy on Test Data: 0.8135964912280702

	precision	recall	f1-score	support
0	0.88	0.91	0.90	754
1	0.76	0.70	0.73	307
accuracy			0.85	1061
macro avg	0.82	0.80	0.81	1061
weighted avg	0.85	0.85	0.85	1061

Table-21 Classification report for Ada-Boosting of train set

	precision	recall	f1-score	support
0	0.84	0.88	0.86	303
1	0.75	0.67	0.71	153
accuracy			0.81	456
macro avg	0.79	0.78	0.79	456
weighted avg	0.81	0.81	0.81	456

Table-22 Classification report for Ada-Boosting of test set

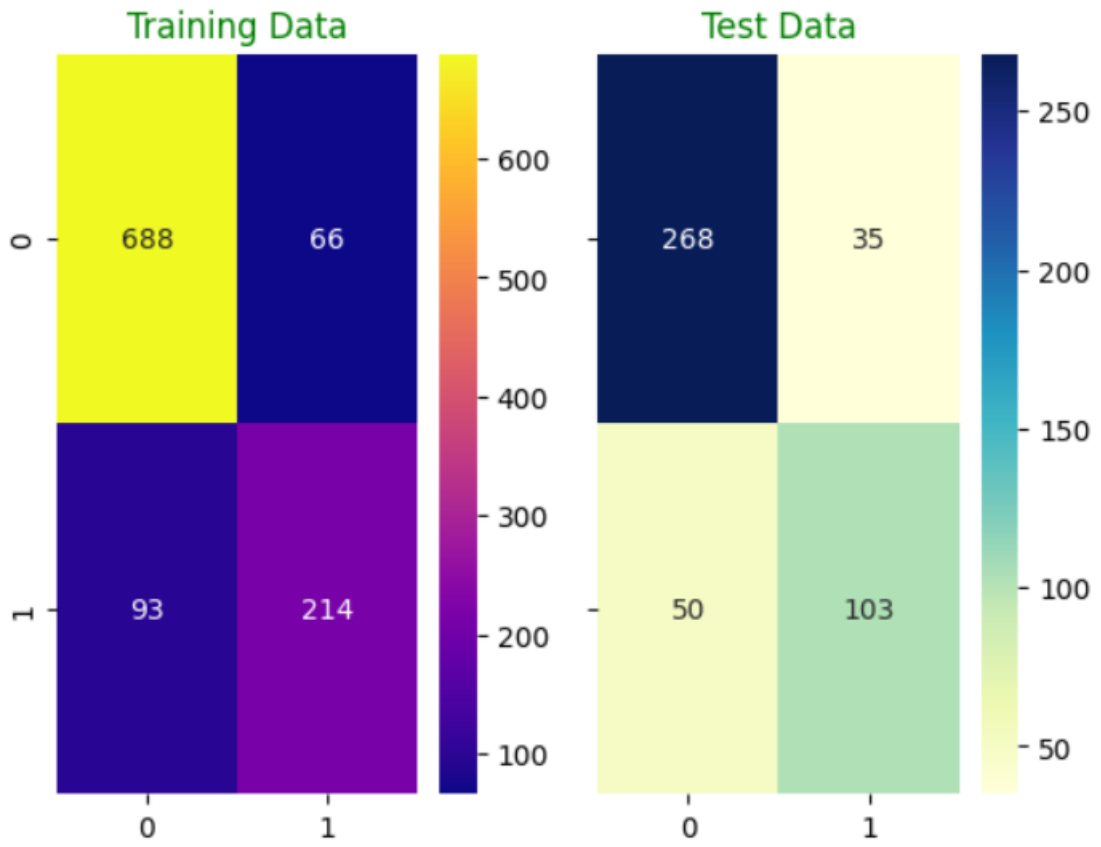
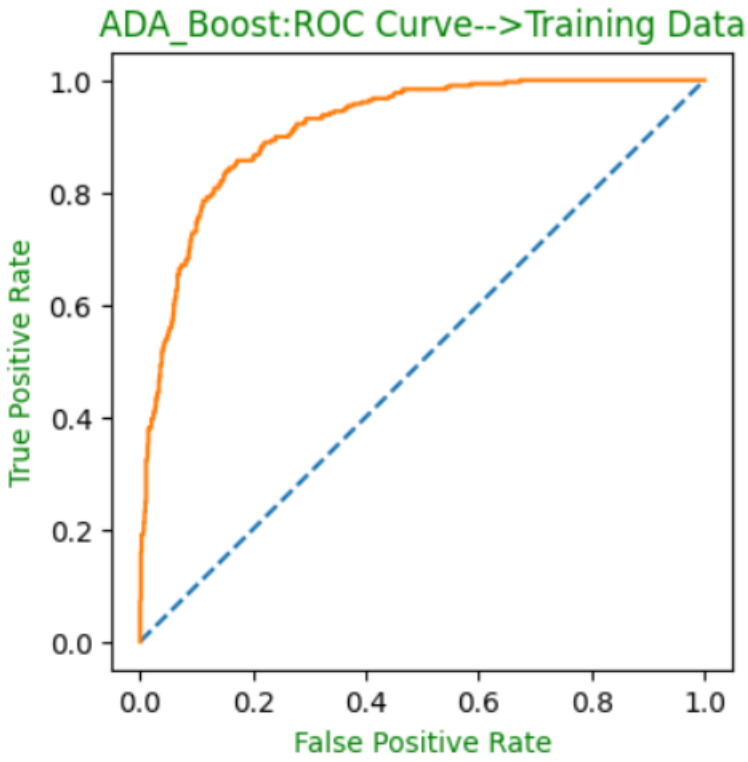


Fig-23 Confusion Matrix for Ada-Boosting

AUC: 0.915



AUC: 0.877

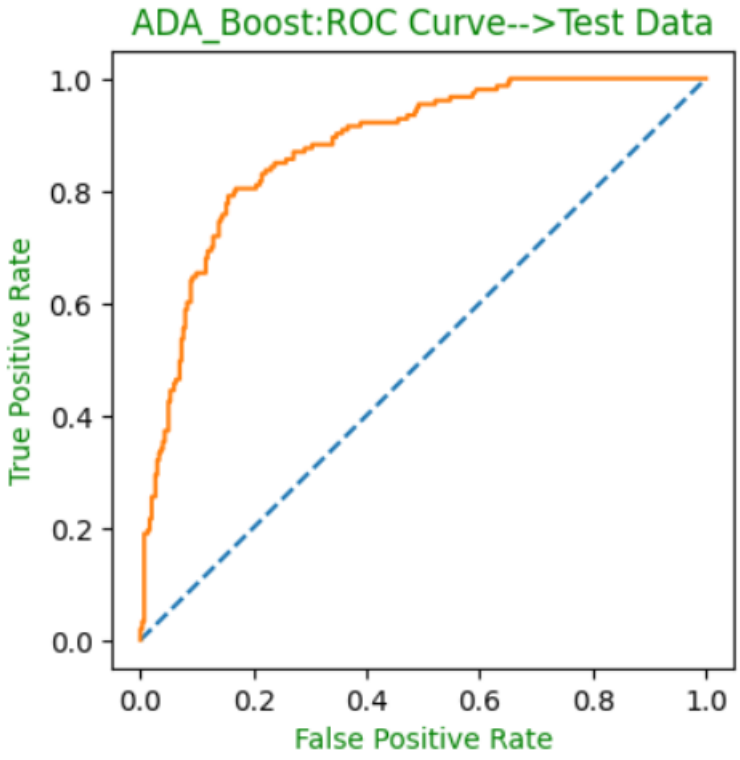


Fig-24 AUC-ROC graph for Ada-Boosting

After Grid search and Tuning:

Accuracy on Train Data: 0.8378887841658812

Accuracy on Test Data: 0.8135964912280702

	precision	recall	f1-score	support
0	0.86	0.93	0.89	754
1	0.77	0.62	0.69	307
accuracy			0.84	1061
macro avg	0.82	0.77	0.79	1061
weighted avg	0.83	0.84	0.83	1061

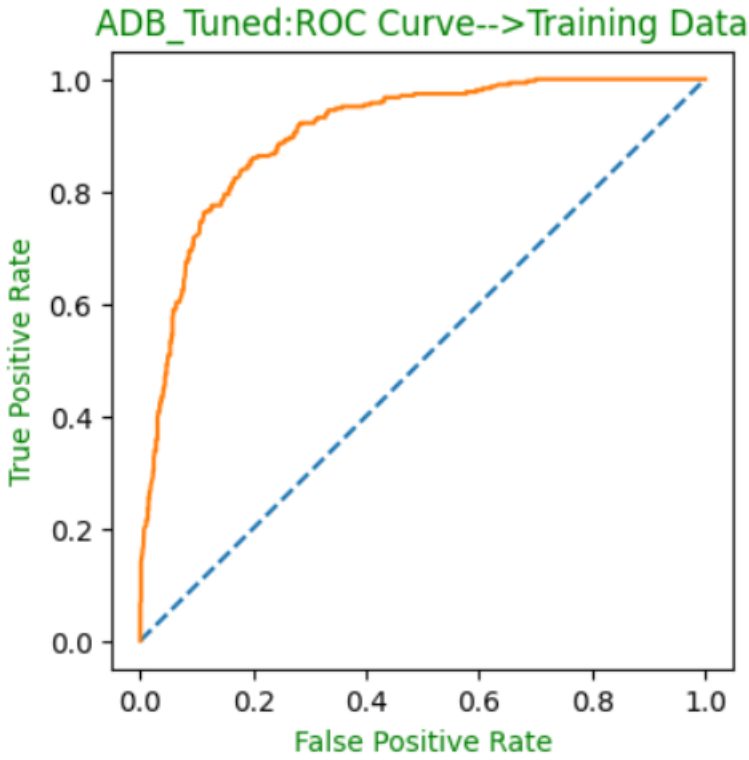
Table-23 Classification report for Tuned Ada-Boosting of train set

	precision	recall	f1-score	support
0	0.84	0.88	0.86	303
1	0.75	0.67	0.71	153
accuracy			0.81	456
macro avg	0.79	0.78	0.79	456
weighted avg	0.81	0.81	0.81	456

Table-24 Classification report for Tuned Ada-Boosting of test set

Fig-25 Confusion Matrix for Tuned Ada-Boosting

AUC: 0.904



AUC: 0.887

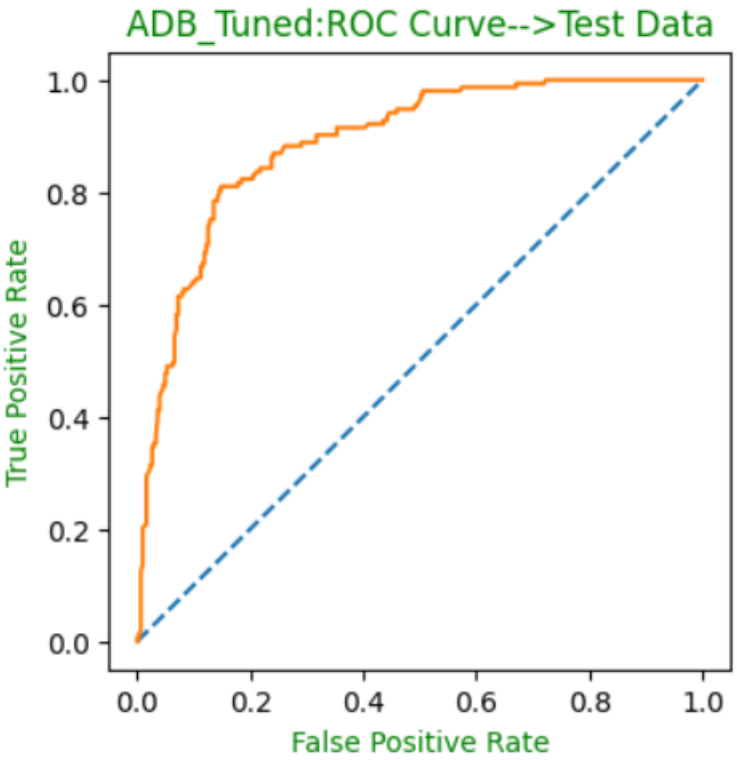


Fig-26 AUC-ROC graph for Tuned Ada-Boosting

1.2.5 Gradient Boosting Method:

Now, we will see the values for accuracy score before tuning and after tuning.

Before grid search:

Accuracy on Train Data: 0.8925541941564562

Accuracy on Test Data: 0.8355263157894737

	precision	recall	f1-score	support
0	0.91	0.94	0.93	754
1	0.84	0.78	0.81	307
accuracy			0.89	1061
macro avg	0.88	0.86	0.87	1061
weighted avg	0.89	0.89	0.89	1061

Table-25 Classification report for Gradient Boosting of train set

	precision	recall	f1-score	support
0	0.85	0.91	0.88	303
1	0.80	0.69	0.74	153
accuracy			0.84	456
macro avg	0.82	0.80	0.81	456
weighted avg	0.83	0.84	0.83	456

Table-26 Classification report for Gradient Boosting of test set

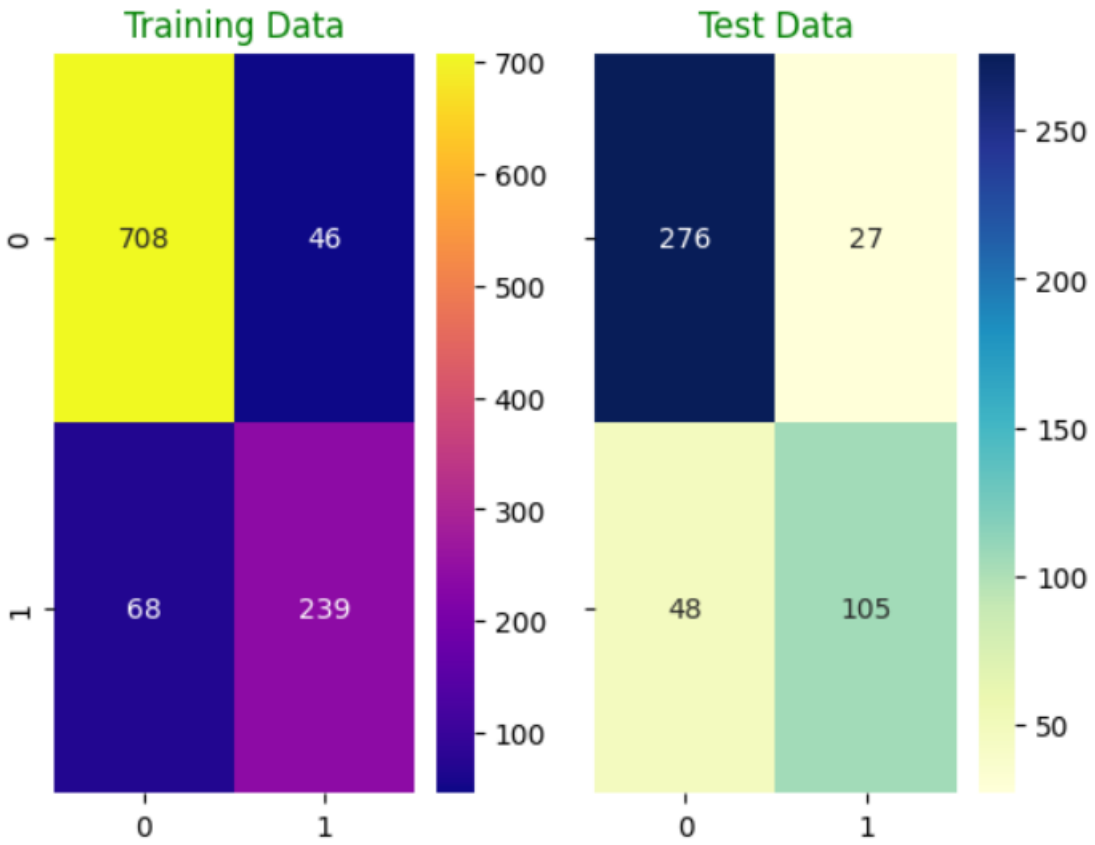
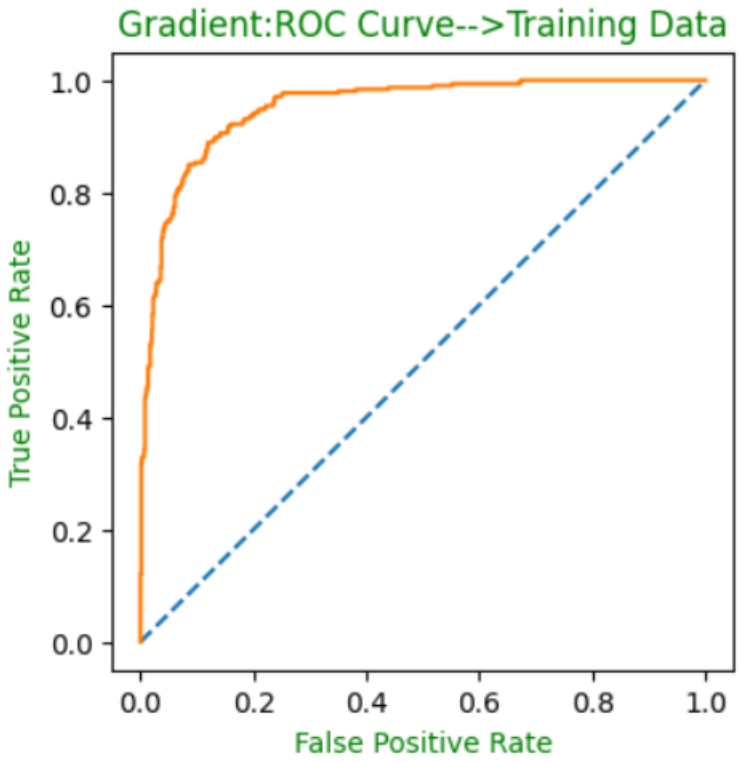


Fig-27 Confusion Matrix for Gradient Boosting

AUC: 0.951



AUC: 0.899

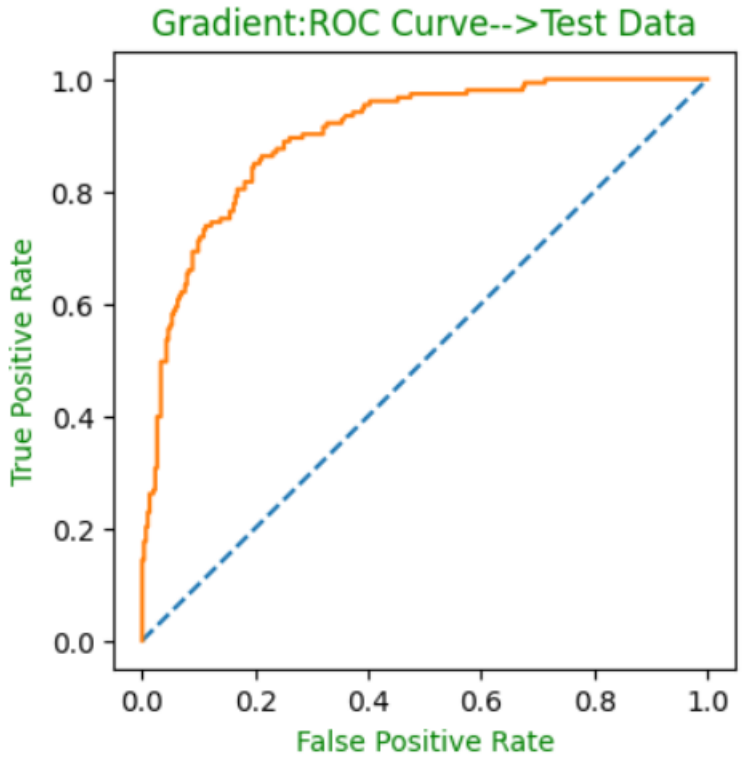


Fig-28 AUC-ROC graph for Gradient Boosting

After Grid search and Tuning:

Accuracy on Train Data: 0.8803016022620169

Accuracy on Test Data: 0.8289473684210527

	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-27 Classification report for Tuned Gradient Boosting of train set

	precision	recall	f1-score	support
0	0.85	0.90	0.87	303
1	0.77	0.69	0.73	153
accuracy			0.83	456
macro avg	0.81	0.80	0.80	456
weighted avg	0.83	0.83	0.83	456

Table-28 Classification report for Tuned Gradient Boosting of test set

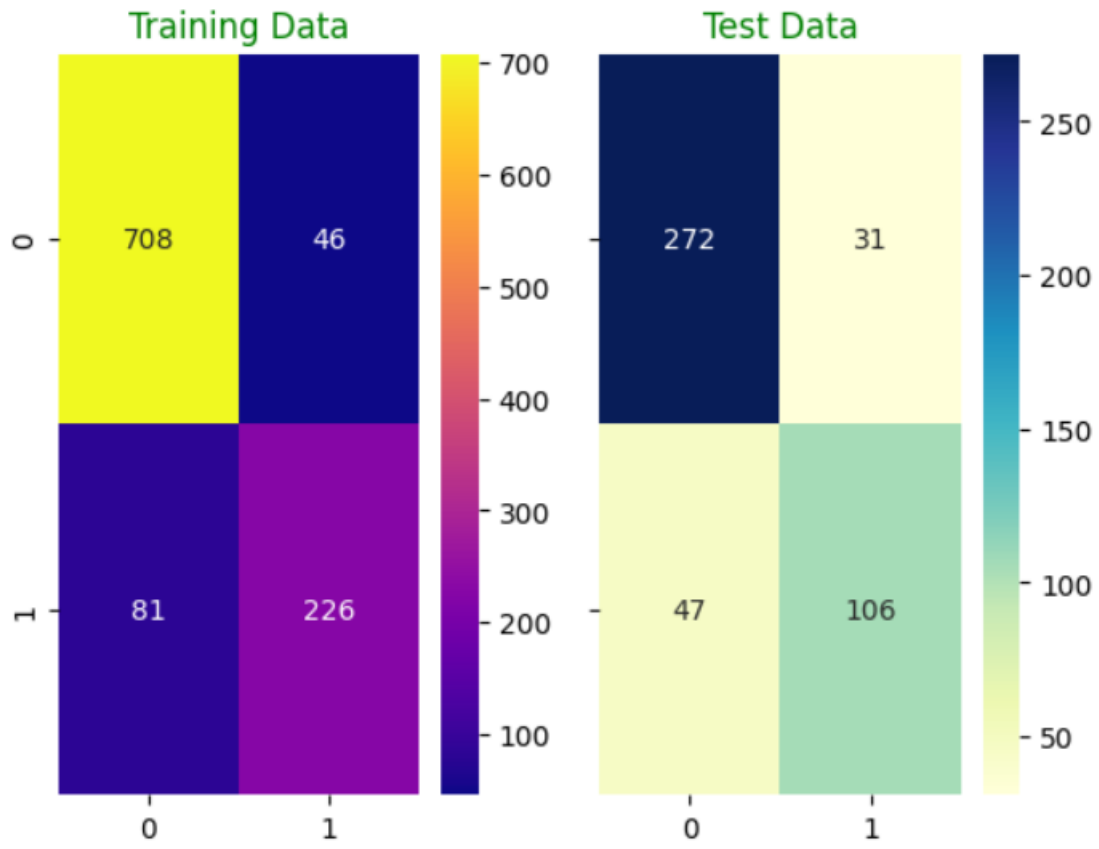
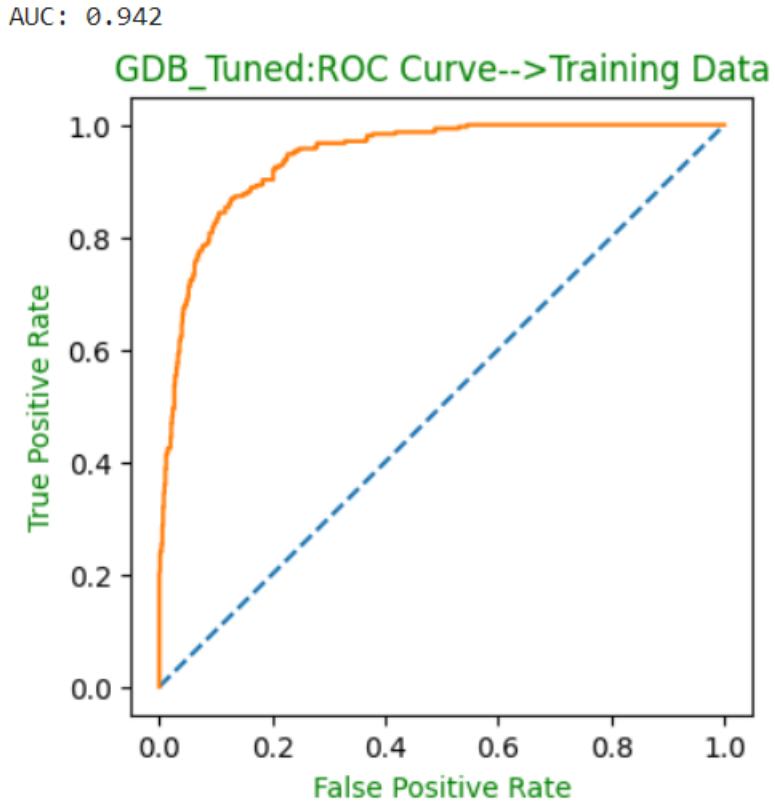


Fig-29 Confusion Matrix for Tuned Gradient Boosting



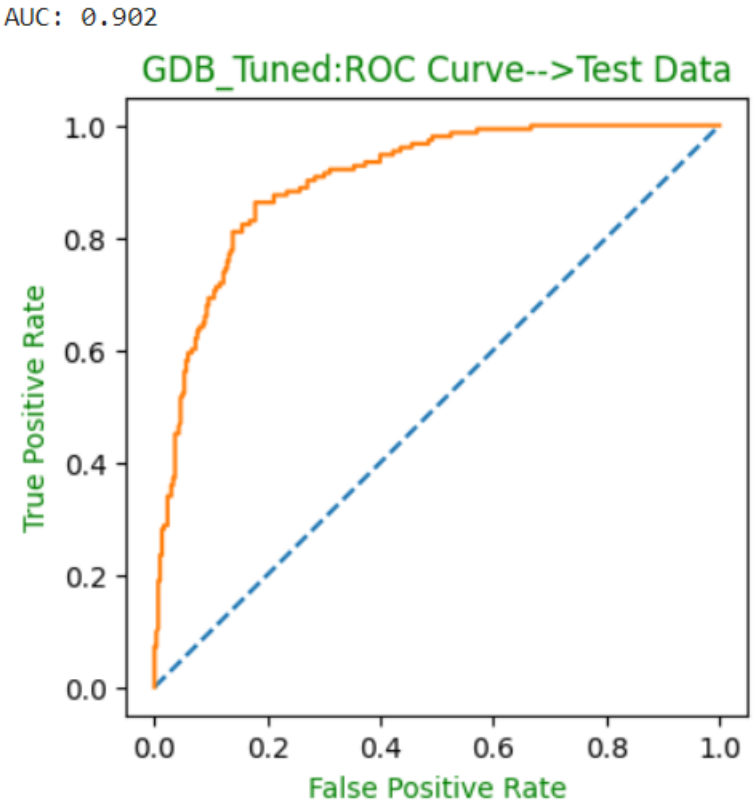


Fig-30 AUC-ROC graph for Tuned Gradient Boosting

1.3 Model Comparison and Conclusion:

Of all the models, Tuned Gradient Boosting has performed better in terms of accuracy, precision, recall and ROC-AUC curve. Although algorithms such as KNN, Naïve bayes, pruned Decision trees and Ada-Boost have performed exceptionally well it was Tuned Gradient boosting which outperformed rest all other good performing algorithms. However, models such as Decision trees without pruning and bagging without tuning the parameters resulted in Overfitting and to overcome that pruning and hyper tuning parameters were employed and the results were outstanding.

Model	Class	Accuracy		Precision		Recall		AUC	
		Train	Test	Train	Test	Train	Test	Train	Test
KNN_5	Labour(1)	0.8444	0.8201	0.88	0.85	0.9	0.88	0.91	0.88
	Conservative(0)			0.74	0.75	0.7	0.69		
KNN_16	Labour(1)	0.836	0.83	0.87	0.85	0.9	0.92	0.905	0.88
	Conservative(0)			0.74	0.81	0.67	0.67		
KNN_Tuned	Labour(1)	0.8473	0.8157	0.88	0.85	0.91	0.87	0.918	0.879
	Conservative(0)			0.75	0.74	0.7	0.7		
Naïve Bayes	Labour(1)	0.835	0.8222	0.88	0.87	0.9	0.87	0.888	0.876
	Conservative(0)			0.73	0.74	0.69	0.73		
Bagging	Labour(1)	0.91	0.83	0.92	0.84	0.96	0.91	0.98	0.89
	Conservative(0)			0.89	0.8	0.79	0.67		
ADA Boost	Labour(1)	0.85	0.81	0.88	0.84	0.91	0.88	0.91	0.87
	Conservative(0)			0.76	0.75	0.7	0.67		
ADA Boost Tuned	Labour(1)	0.83	0.81	0.86	0.84	0.93	0.88	0.9	0.88
	Conservative(0)			0.77	0.75	0.62	0.67		
Gradient Boost	Labour(1)	0.89	0.83	0.91	0.85	0.94	0.91	0.95	0.89
	Conservative(0)			0.84	0.8	0.78	0.69		
Gradient Boost Tuned	Labour(1)	0.87	0.84	0.9	0.87	0.94	0.91	0.94	0.9
	Conservative(0)			0.83	0.81	0.73	0.72		

Table-29 All Model comparison Table

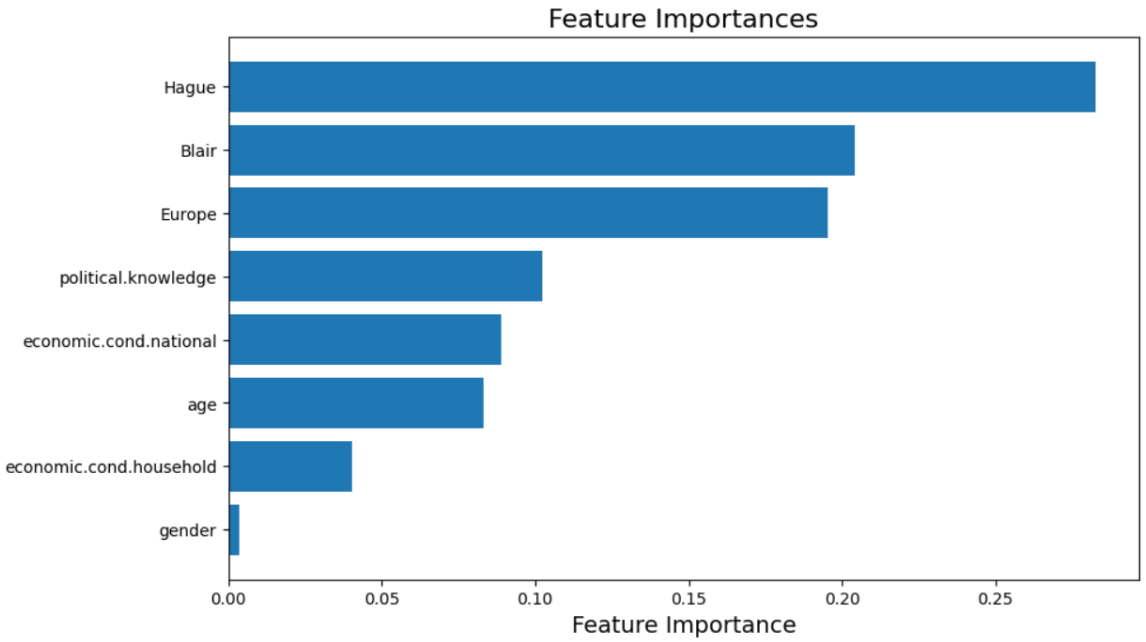


Fig-31 Important Model feature of Tuned Gradient Boosting

It is observed that the most import feature surprisingly is Hague, the leader of Conservative party, which is followed by the leader if Labour party Blair and third important feature is Europe i.e. for people to have European sentiments. And the least important features are gender, Household economic condition.

Any further decisions with respect to dropping unimportant features can be taken post consulting domain and further improve the model prediction and so.

Business Recommendation's:

The chance of Labour party to win the Election is more as compared to the Conservative party as most of the votes are casted for the Labour party. Female candidate casts more vote as compared to male candidate.

This is a helpful insight while considering votes that might be casted in future to win an election. Older age group people caste more votes as compared to younger. candidates. Gathering more data will help in the Training the model more accurately. and thus, improving the predictive powers. News channel should gather large sample of the dataset for correct. predictions. Parties should try to attract more Male candidates to increase the vote. banks.

As it evident from the Bivariate analysis that male voters are less in not as compared to female voters. For Male voters, more employment opportunities should be provided to attract voters as a part of their election campaign. For Female voters, Women Empowerment laws should be a part of election campaign.

Problem-2: Text Analysis of Presidents

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 Number of Characters, words and Sentences in all 3 Speeches:

First of all, we will load and download the necessary library files and packages as per requirement. We will download inaugurat, punkt ans stopwords packages form the nltk files.

From all the speeches loaded we will select and save the text file in roosevelt_speech, Kennedy_speech and nixon_speech as per requirement.

Now, we will use count stats function on the three variables and obtain the characters, words and sentences counts and save the in the respect new variables. And finally print those variables and we obtain the given below table.

Table-32 Count Stats for all 3 speeches.

2.2 Text Cleaning:

In this section we perform remove stop words using corpus and stemming method; in doing so we have reduced the unwanted information down to its base root.

The most common first three words being repeated in individual speeches are:

The 3 most common words in Roosevelt speech are: ('nation' = 17), ('know' = 10), ('peopl' = 9)

The 3 most common words in Kennedy speech are :('let' = 16), ('us' = 12), ('power' = 9)

The 3 most common words in Nixon speech are :('us' = 26), ('let' = 22), ('america' = 21)

And three most common words in all three speeches after preprocessing:

us: 46 times

nation: 40 times

let: 39 times

2.3 Plotting Word Cloud for all three speeches:

By using word cloud library, we have plotted the word clouds all three president's speeches are as follows:

For Roosevelt Speech:

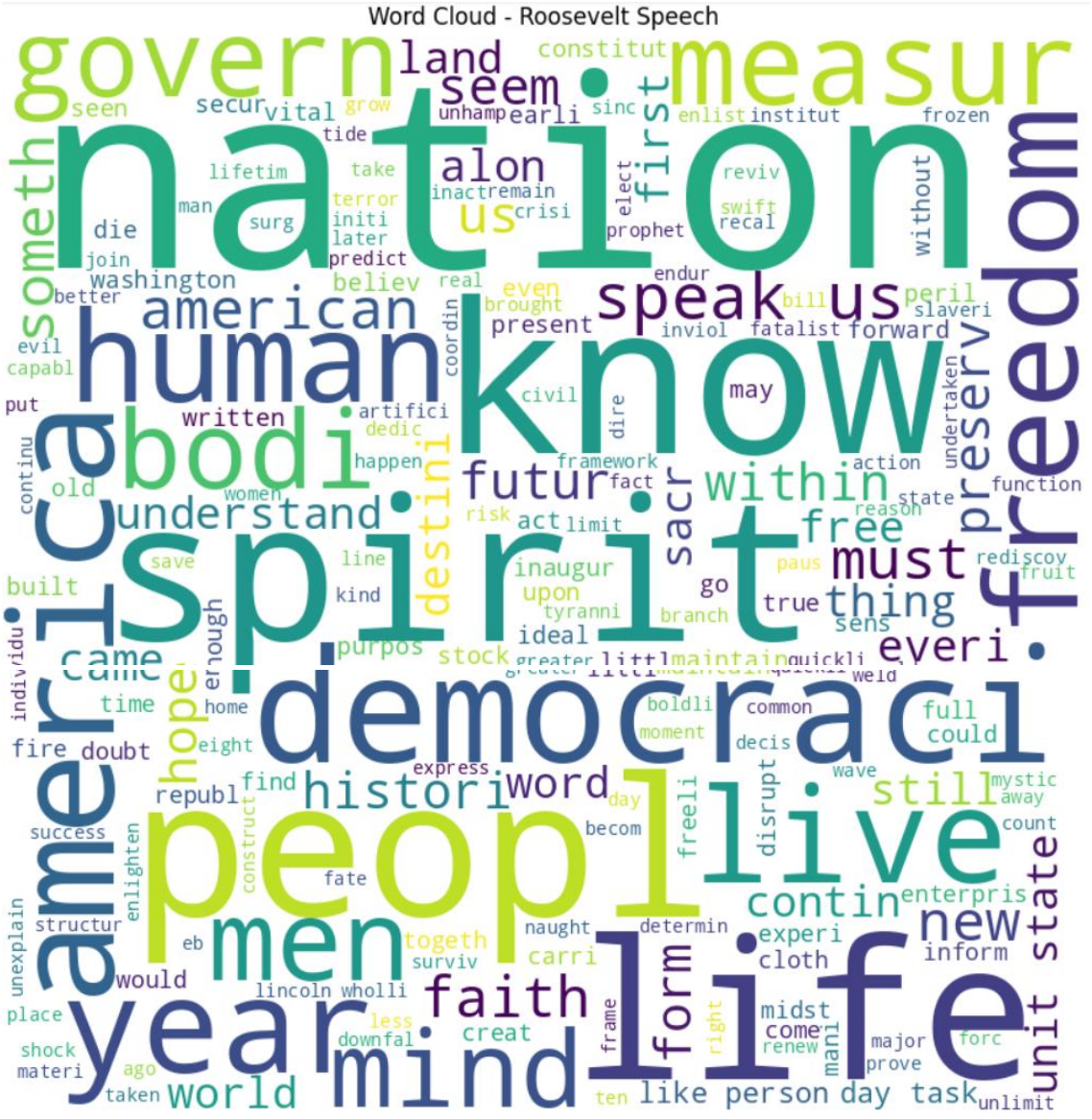


Fig-32 Word Could of Roosevelt speech

For Kennedy Speech:

Fig-33 World Could of Kennedy speech

For Nixon Speech:

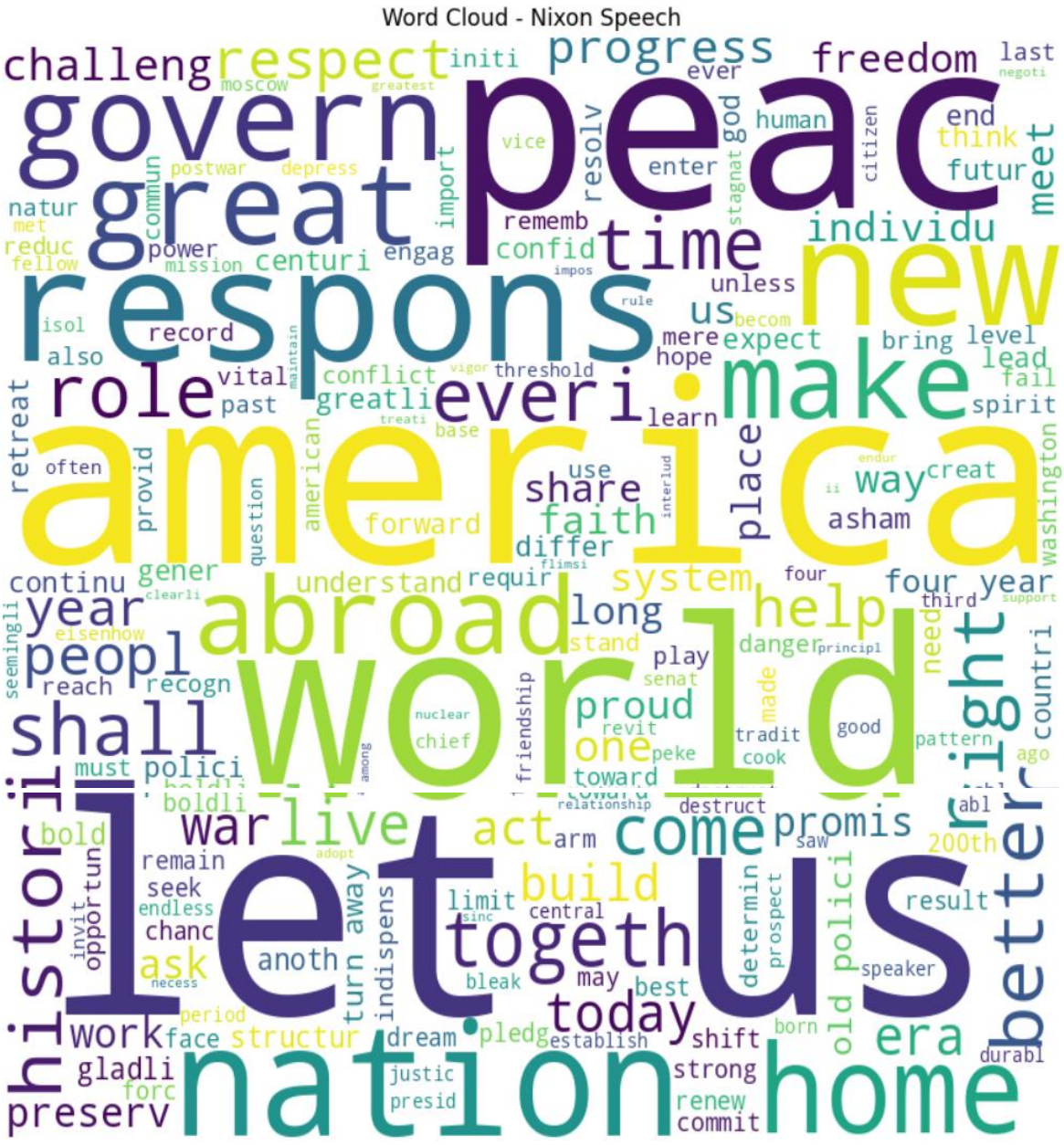


Fig-34 World Could of Nixon speech