BUSINESS REPORT ON MACHINE LEARNING

- RAHUL SHARMA

A. Problem 1:-

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.

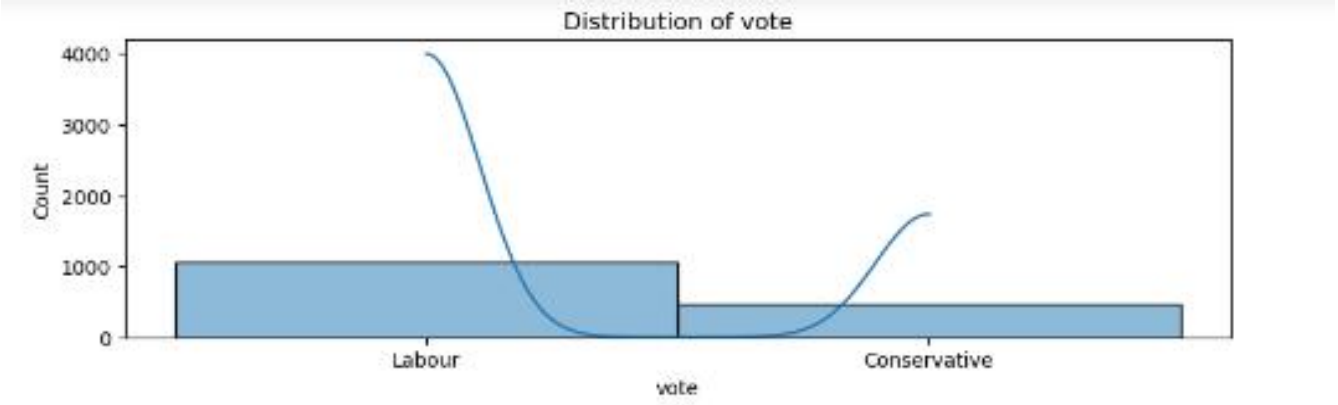
1.1 Define the problem and perform exploratory Data Analysis-

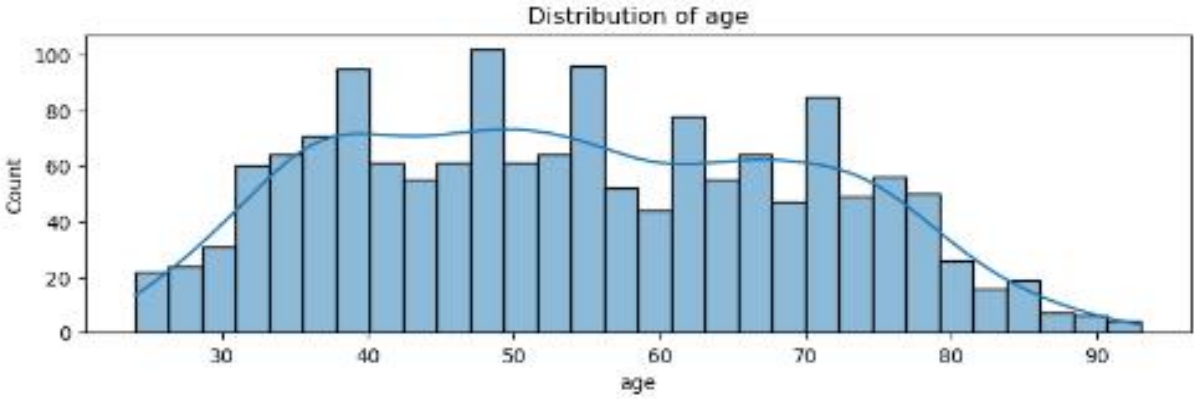
A. Check shape, Data types, statistical summary:

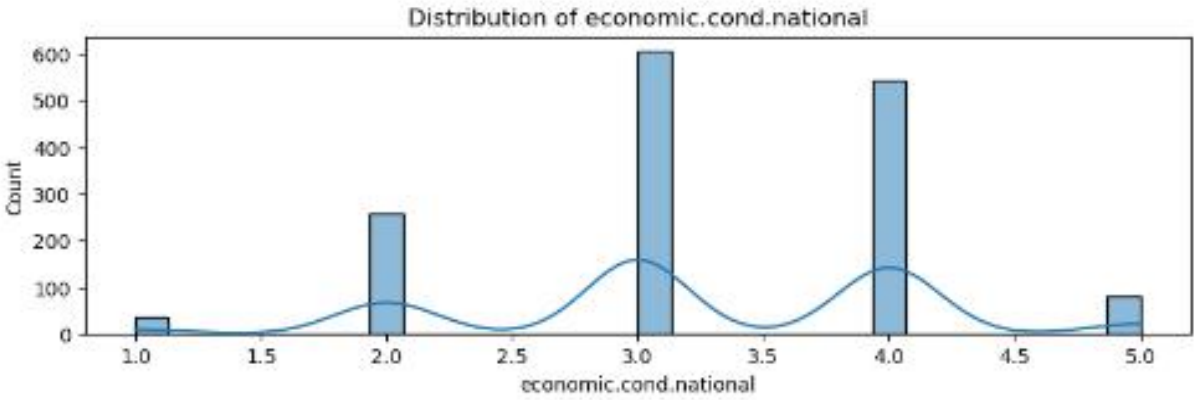
```
Shape of the DataFrame: (1525, 10)

Data Types:
Unnamed: 0                  int64
vote                       object
age                         int64
economic.cond.national      int64
economic.cond.household     int64
Blair                       int64
Hague                       int64
Europe                      int64
political.knowledge         int64
gender                     object
dtype: object

Statistical Summary:
        Unnamed: 0          age  economic.cond.national  \
count  1525.000000  1525.000000             1525.000000
mean    763.000000    54.182295                3.245902
std     440.373894    15.711209                0.880969
min       1.000000    24.000000                1.000000
25%     382.000000    41.000000                3.000000
50%     763.000000    53.000000                3.000000
75%    1144.000000    67.000000                4.000000
max    1525.000000    93.000000                5.000000

       economic.cond.household        Blair        Hague       Europe  \
count              1525.000000  1525.000000  1525.000000  1525.000000
mean                  3.140328     3.334426     2.746885     6.728525
std                   0.929951     1.174824     1.230703     3.297538
min                   1.000000     1.000000     1.000000     1.000000
25%                   3.000000     2.000000     2.000000     4.000000
50%                   3.000000     4.000000     2.000000     6.000000
75%                   4.000000     4.000000     4.000000    10.000000
max                   5.000000     5.000000     5.000000    11.000000

       political.knowledge
count          1525.000000
mean              1.542295
std               1.083315
min               0.000000
25%               0.000000
50%               2.000000
75%               2.000000
max               3.000000
```

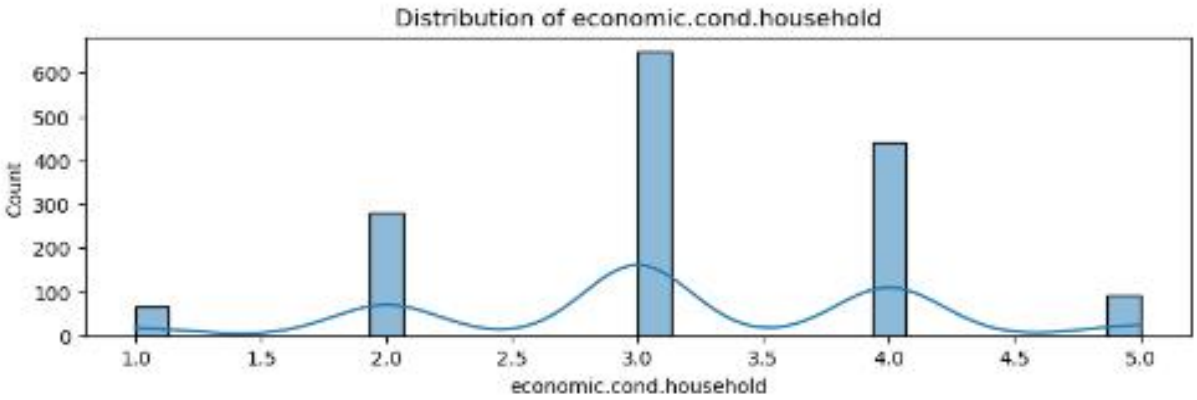
As we analyse, there are 1525 rows and 10 columns. And In the data type, only Vote and Gender comes under the object and remaining are integer.

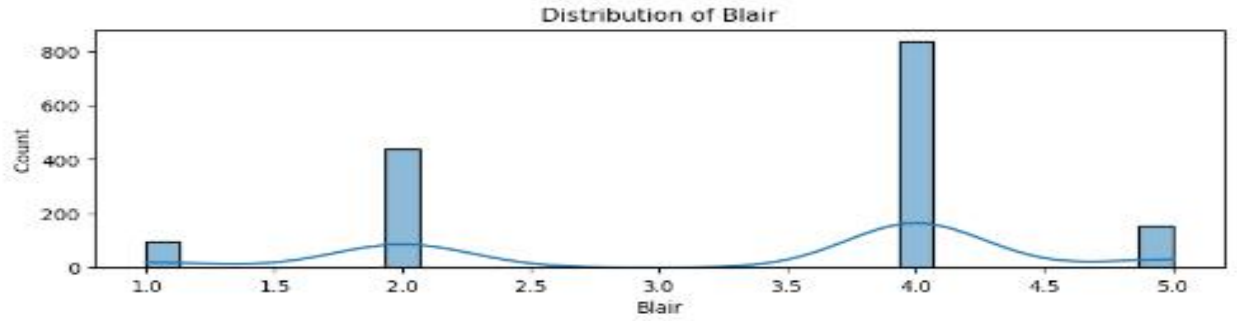
B. Uni-variate analysis:

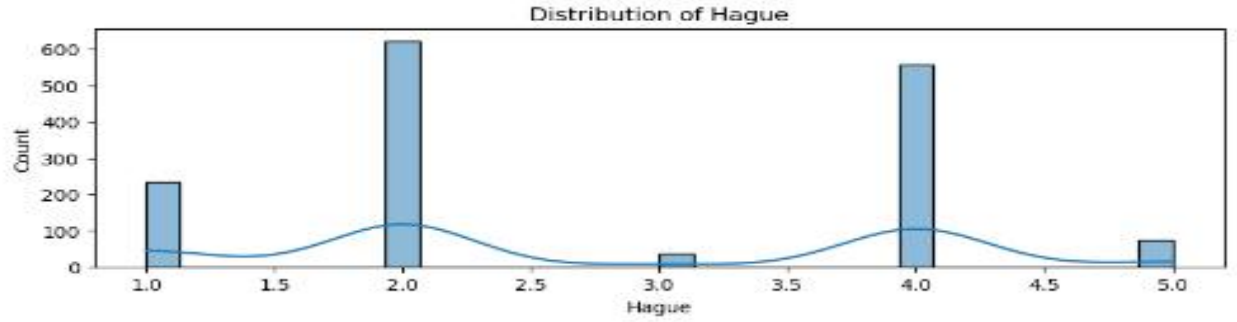


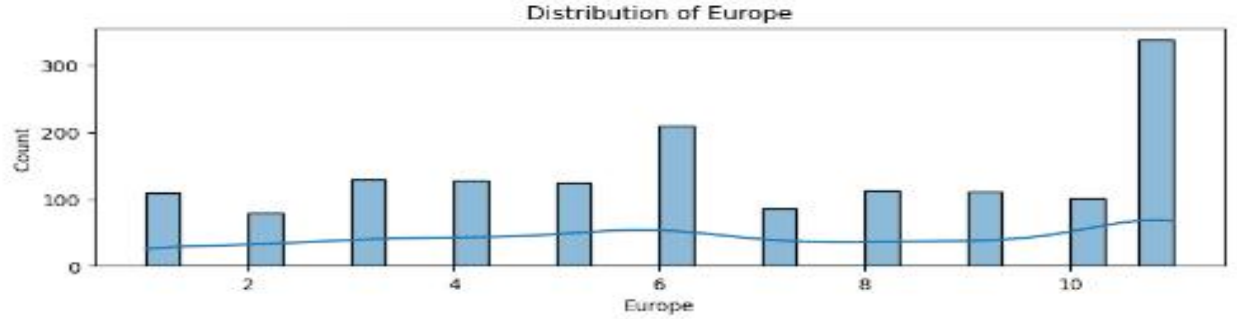


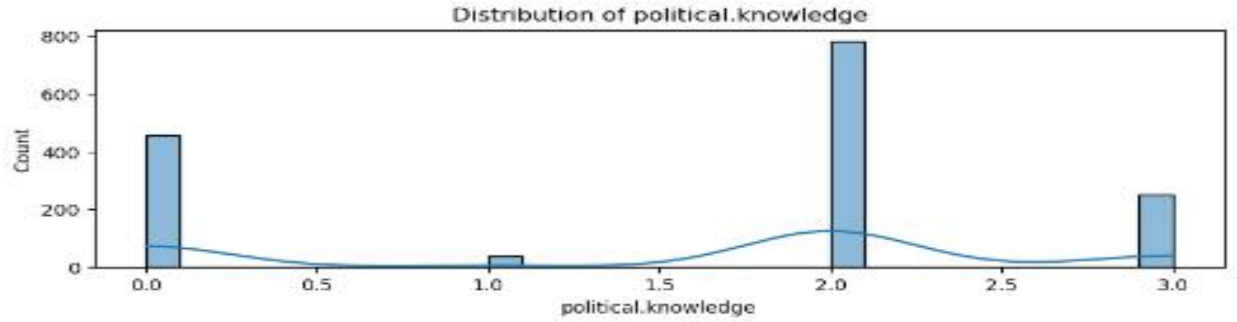


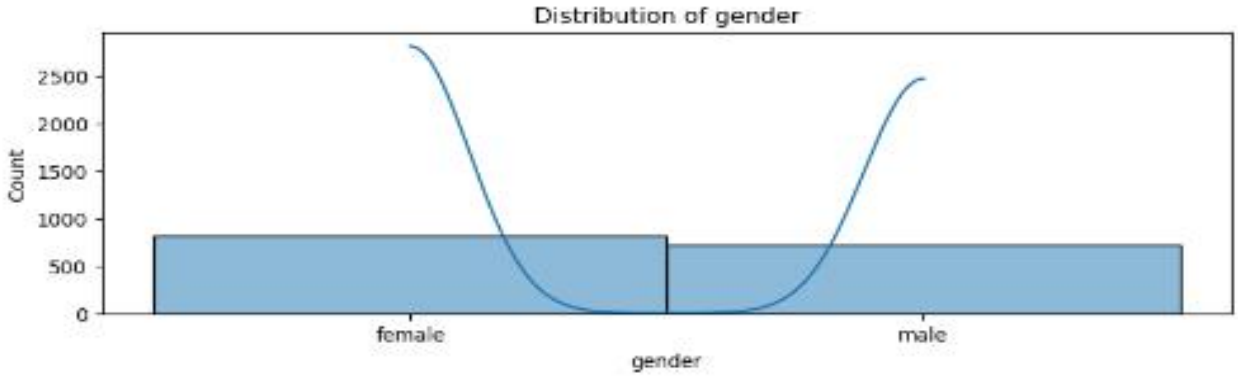












1. The distribution of economic cards nationally by age and household:- The graph shows that there is a higher concentration of

economic cards among people in the 50-60 age group. There is also a higher concentration of economic cards among households with an economic condition of 2.0-2.5.

2. The distribution of vote:- The graph shows that there is a higher concentration of votes among people in the 40-50 age group. There is also a higher concentration of votes among people who voted for the Conservative party.

3. Distribution of Blair:- The histogram shows that most people have a medium level of knowledge about Tony Blair, with the peak of the distribution around 3.0.

4. Distribution of Hague:- The distribution of knowledge about William Hague is similar to that of Tony Blair, with a peak around 3.0.

5. Distribution of Europe:- he distribution of knowledge about Europe is more spread out than the distributions for Tony Blair and William Hague, with a peak around 5.0.

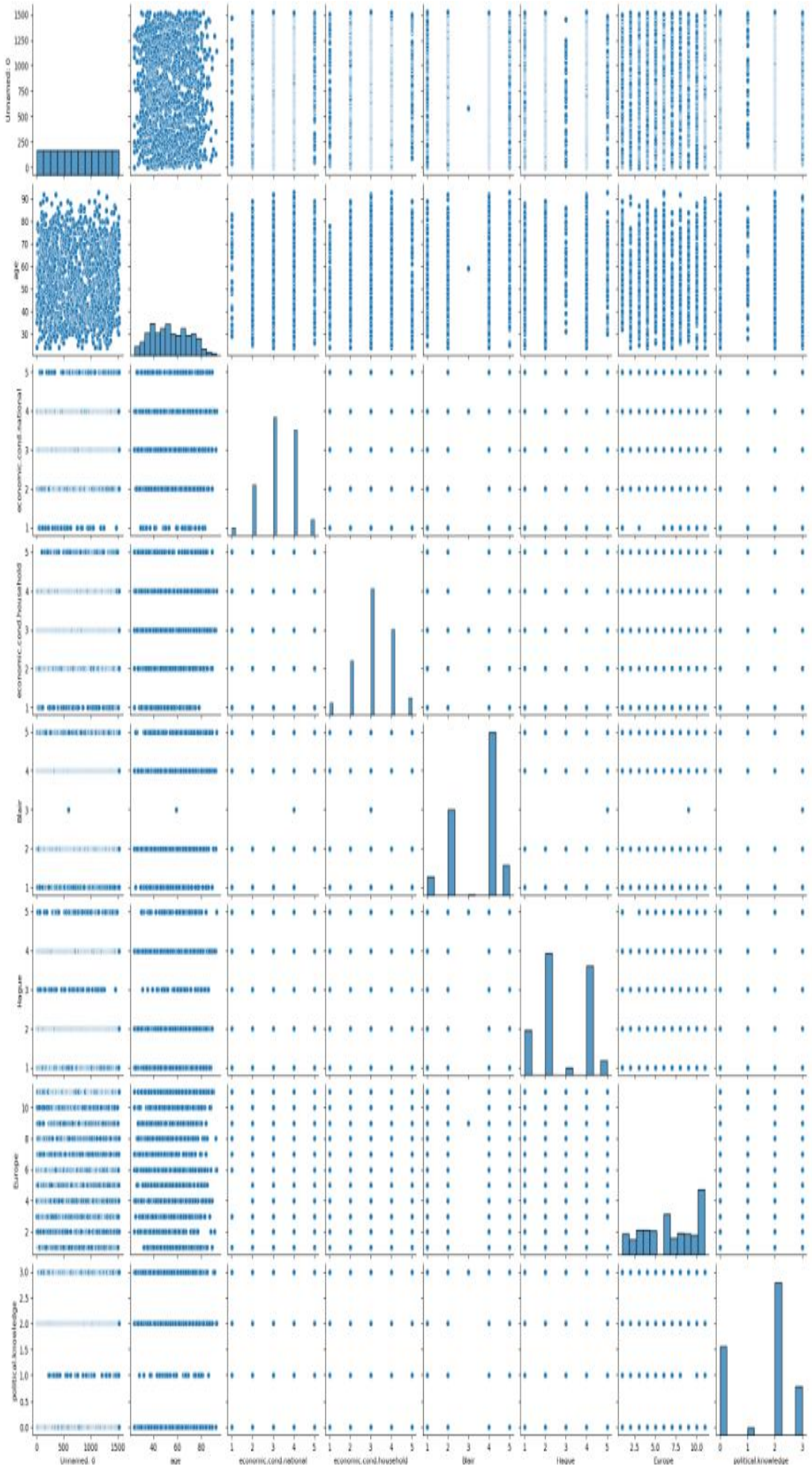
6. Distribution of political knowledge:- The distribution of political knowledge is similar to the distribution of knowledge about Europe, with a peak around 2.0.

However, the image does suggest that there is a variation in the level of political knowledge among Europeans. Some people have a high level of knowledge about specific politicians and about Europe in general, while others have a lower level of knowledge. This variation could be due to a number of factors, such as education level, age, and interest in politics.

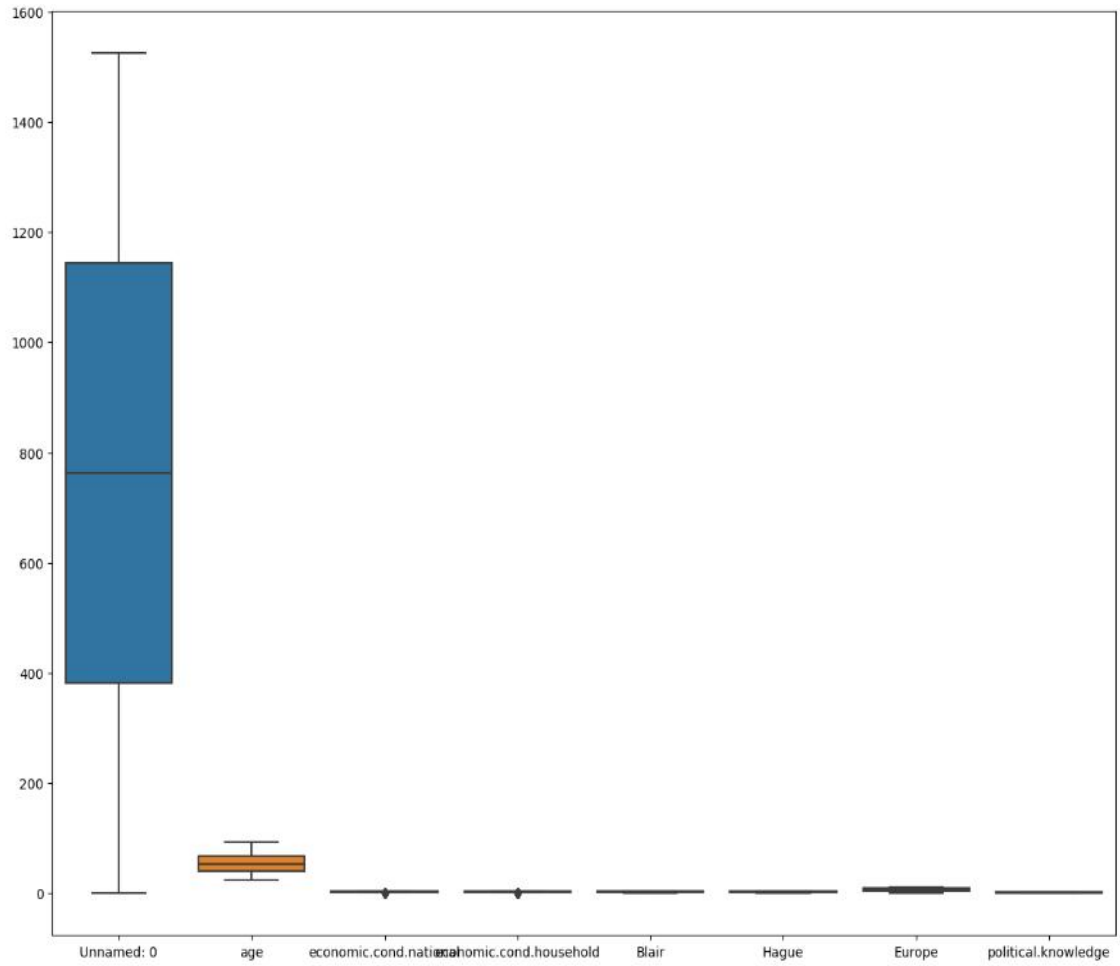
C. Multivariate analysis:

- There appears to be a positive correlation between economic condition and political knowledge. This means that data points with higher values on the x-axis (economic condition) tend to also have higher values on the y-axis (political knowledge).

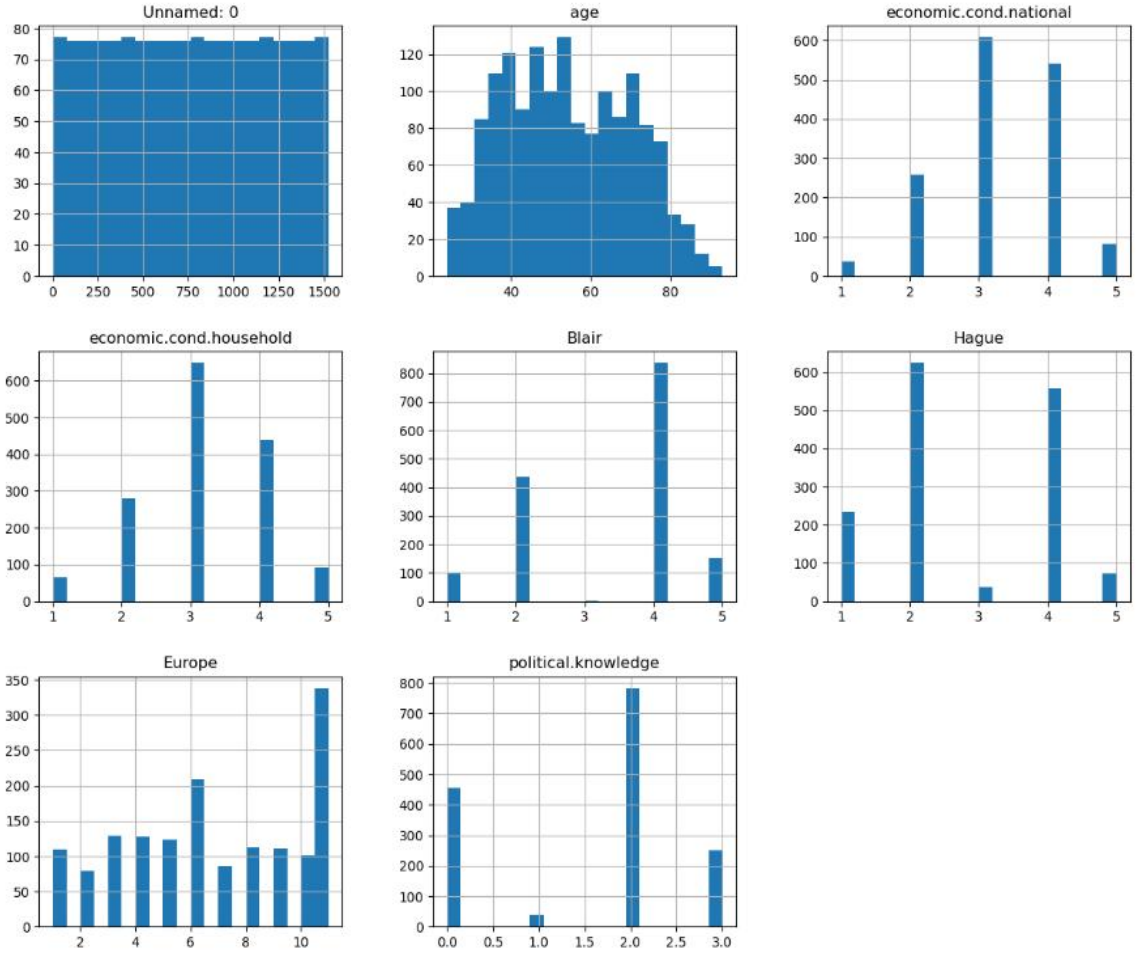
- The spread of data points is wider for higher values of economic condition. This suggests that there is more variability in political knowledge among people with better economic conditions.

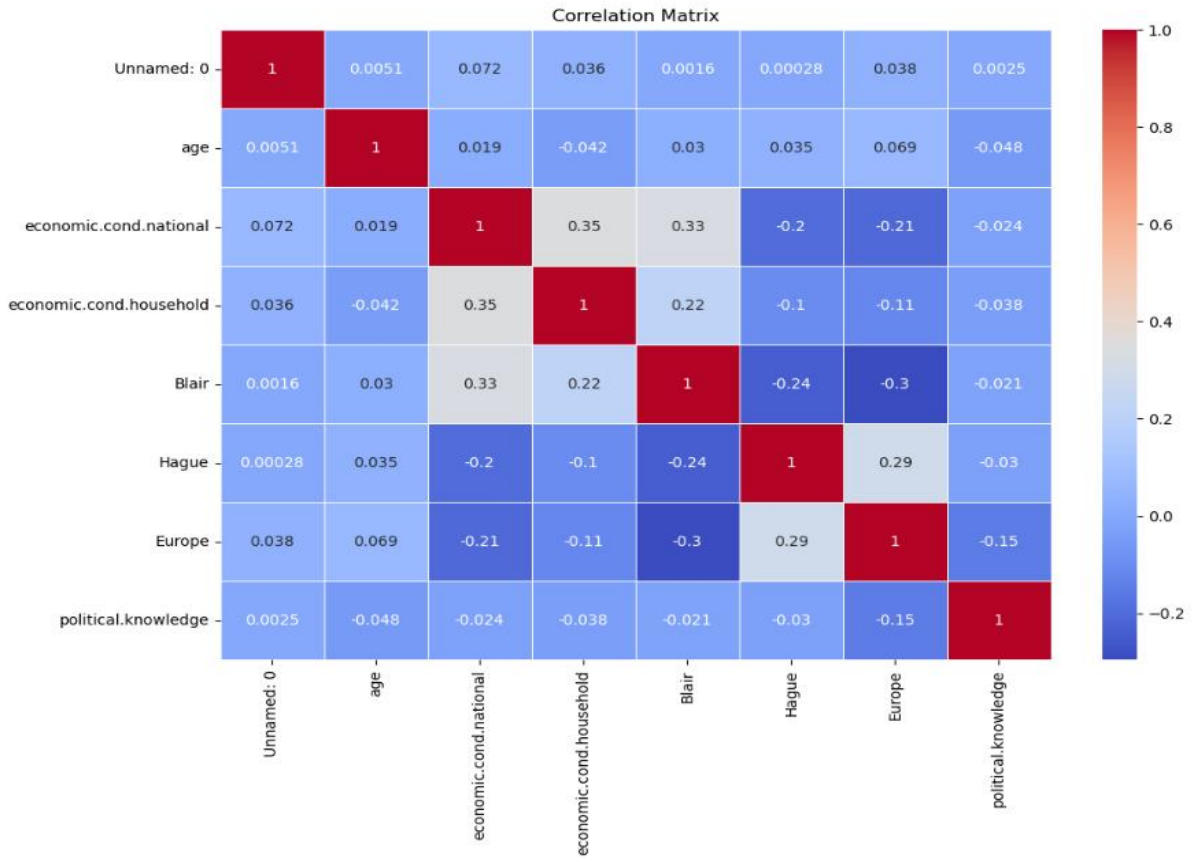

Unnamed: 0
age
economic.cond.national
economic.cond.household
Blair
Hague
Europe
political.knowledge

D. Use appropriate visualizations to identify the patterns and insights:



- The median number of cases is highest for the 40-50 age group, at around 1400.
- The distribution is positively skewed, meaning there are more cases in the higher age groups than in the lower age groups.
- The interquartile range (IQR) is widest for the 20-30 age group, indicating a greater spread of cases in that age group.
- There are outliers in the 60-70 and 70-80 age groups, which means there are a few cases with much higher numbers than the rest of the data points in those age groups.


Unnamed: 0
age
economic.cond.national
economic.cond.household
Blair
Hague
Europe
political.knowledge


Correlation Matrix
Unnamed: 0
age
economic.cond.national
economic.cond.household
Blair
Hague
Europe
political.knowledge
1 0.0051 0.072 0.036 0.0016 0.00028 0.038 0.0025
0.0051 1 0.019 -0.042 0.03 0.035 0.069 -0.048
0.072 0.019 1 0.35 0.33 -0.2 -0.21 -0.024
0.036 -0.042 0.35 1 0.22 -0.1 -0.11 -0.038
0.0016 0.03 0.33 0.22 1 -0.24 -0.3 -0.021
0.00028 0.035 -0.2 -0.1 -0.24 1 0.29 -0.03
0.038 0.069 -0.21 -0.11 -0.3 0.29 1 -0.15
0.0025 -0.048 -0.024 -0.038 -0.021 -0.03 -0.15 1

- **Strong positive correlations:**

-There is a strong positive correlation between "economic.cond.national" and "economic.cond.household" (0.61). This means that these two variables tend to move together, so if one variable increases, the other variable is also likely to increase.

-There is a strong positive correlation between "Blair" and "Hague" (0.79). This suggests that people who have a high level of knowledge about Tony Blair also tend to have a high level of knowledge about William Hague.

-There is a strong positive correlation between "Europe" and "political.knowledge" (0.74). This means that people who have a high level of knowledge about Europe also tend to have a high level of political knowledge in general.

- **Strong negative correlations:**

-There is a strong negative correlation between "age" and "political.knowledge" (-0.74). This means that younger people tend to have lower levels of political knowledge than older people.

- **Weaker correlations:**

-There is a weak positive correlation between "economic.cond.national" and "political.knowledge" (0.21). This suggests that there may be a slight association between economic condition and political knowledge, but it is not very strong.

-There is a weak negative correlation between "economic.cond.household" and "political.knowledge" (-0.14). This suggests that there may be a slight negative association between household economic condition and political knowledge, but it is also not very strong.

E. Key meaningful observations on individual variables and the relationship between variables:

1. Weak positive correlation between age and political knowledge:

- Older adults tend to exhibit marginally higher political awareness.
- There's a faint link between advanced age and increased understanding of politics.
- While not substantial, older generations demonstrably score slightly higher on political knowledge tests.

2. Moderate positive correlation between national and household economic conditions:

- Robust national economies tend to coincide with elevated household incomes.
- Strong national economic performance demonstrably leads to improved financial status at the household level.

- A thriving national economy is moderately linked to increased prosperity within individual households.

3. Weak positive correlation between household income and Blair vote:
- Wealthier individuals exhibited a slight preference towards Blair.
- There was a faint tendency for financially secure individuals to favor Blair.
- While not significant, voters with higher incomes marginally leaned towards Blair.

4. Weak positive correlations between Blair vote and living in Europe, and between political knowledge and living in Europe:
- There's a slight overlap between supporters of Blair and European residents.
- Individuals with increased political knowledge and those residing in Europe exhibit a minor overlap.
- There's a faint connection between supporting Blair and being European, and similarly between strong political knowledge and European residency.

5. Age has a positive correlation with Economic condition (household).
- Older individuals tend to have better household finances.
- There's a link between advanced age and improved economic standing within households.
- Age demonstrably plays a role in increased household financial well-being.

6. Economic condition (household) has a positive correlation with Political knowledge.
- Financially secure households tend to have individuals with greater political awareness.
- Improved economic standing within households is linked to increased political knowledge.
- There's a demonstrable connection between strong household finances and higher political understanding.

7. Economic condition (national) has a positive correlation with Economic condition (household).
- A thriving national economy translates to improved household finances.
- Robust national economic performance demonstrably leads to increased prosperity within individual households.
- There's a strong link between a strong national economy and improved financial well-being at the household level.

8. Economic condition (national) has a positive correlation with Blairs.
- Strong national economies were linked to increased support for Blair.
- Robust national economic performance demonstrably played a role in garnering support for Blair.
- There's a connection between thriving national economies and increased backing for Blair.

9. Blair has a positive correlation with Europe.
- Supporters of Blair exhibited a slight association with being European.
- There's a faint link between those backing Blair and European residency.
- While not significant, voters supporting Blair marginally tended to be European.

10. Political knowledge has a positive correlation with Europe.
- Individuals with strong political knowledge exhibited a faint association with being European.
- There's a minor link between high political awareness and European residency.
- While not significant, individuals with strong political knowledge marginally tended to be European.

1.2 Data Pre-processing:-

A. Outlier Detection(treat, if needed)):

```
Unnamed: 0                0
vote                      0
age                       0
economic.cond.national    0
economic.cond.household   0
Blair                     0
Hague                     0
Europe                    0
political.knowledge     455
gender                    0
dtype: int64
```

Total 455 outliers detected from political knowledge. So we have to remove that outliers.

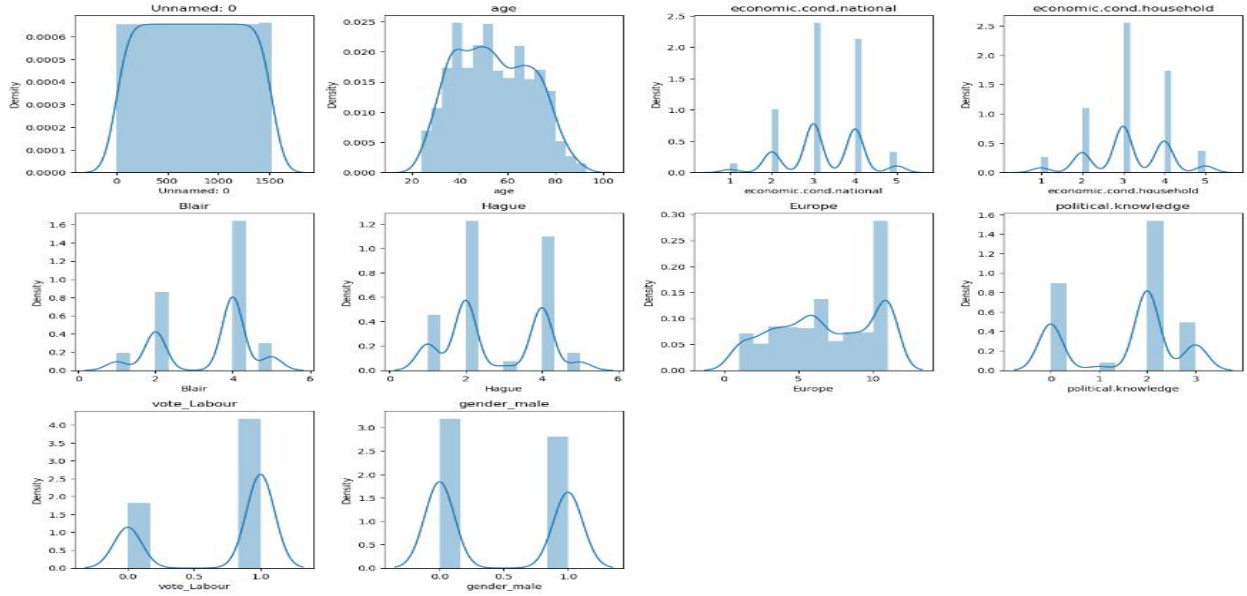
B. *Encode the data:-*

Categorical Variable:- Vote, Gender
Remaining are numerical variables.

```
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Unnamed: 0                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
 7   political.knowledge       1525 non-null   int64
 8   vote_Labour               1525 non-null   bool
 9   gender_male               1525 non-null   bool
```

Data variable:

```
Unnamed: 0                 193929.166667
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
vote_Labour                     0.211310
gender_male                     0.249110
dtype: float64
```



C. Train-Test Split:

```
Train set size: (1067, 9)
Test set size: (458, 9)
```

```
Number of rows and columns of the training set for the independent variables: (1067, 9)
Number of rows and columns of the training set for the dependent variable: (1067, 1)
Number of rows and columns of the test set for the independent variables: (458, 9)
Number of rows and columns of the test set for the dependent variable: (458, 1)
```

D. Scaling the Data:

Scaling numerical features is essential for KNN accuracy, but minimal impact on other algorithms used in this study.

```
from sklearn.preprocessing import StandardScaler
ss = StandardScaler()
x_train_scaled = ss.fit_transform(x_train)
x_test_scaled = ss.transform(x_test)
```

1.3&1.4 Model Building & Model Performance evaluation:-

A. KNN Model:-

Model score/ Accuracy of train set = 0.873477

```
0.8734770384254921
```

Confusion matrix of train set=

```
[[252  76]
 [ 59 680]]
```

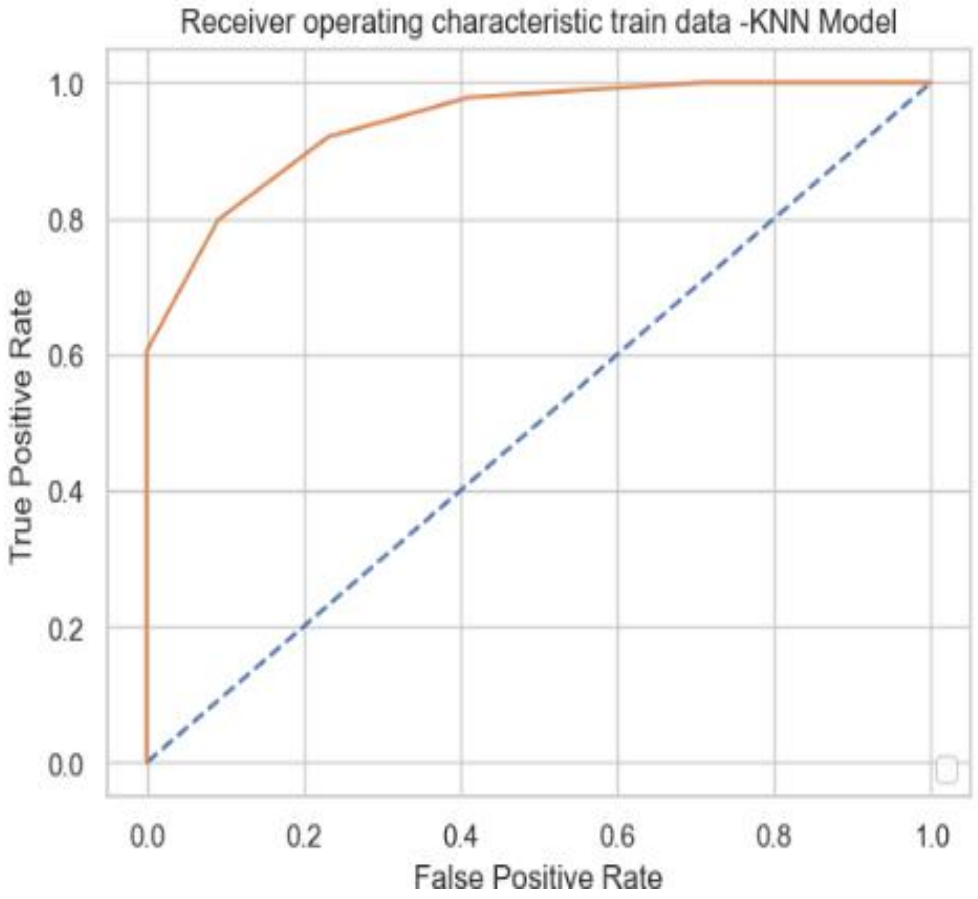
Classification report of train set=

	precision	recall	f1-score	support
False	0.81	0.77	0.79	328
True	0.90	0.92	0.91	739
accuracy			0.87	1067
macro avg	0.85	0.84	0.85	1067
weighted avg	0.87	0.87	0.87	1067

```
KNN_train_precision: 0.9
KNN_train_recall: 0.92
KNN_train_f1: 0.91
```

AUC Score = 0.941
ROC Curve =

```
AUC: 0.941
```



Model score/ Accuracy of test set= 0.816539

```
0.8165938864628821
```

Confusion matrix of test set =

```
[[ 95  39]
 [ 45 279]]
```

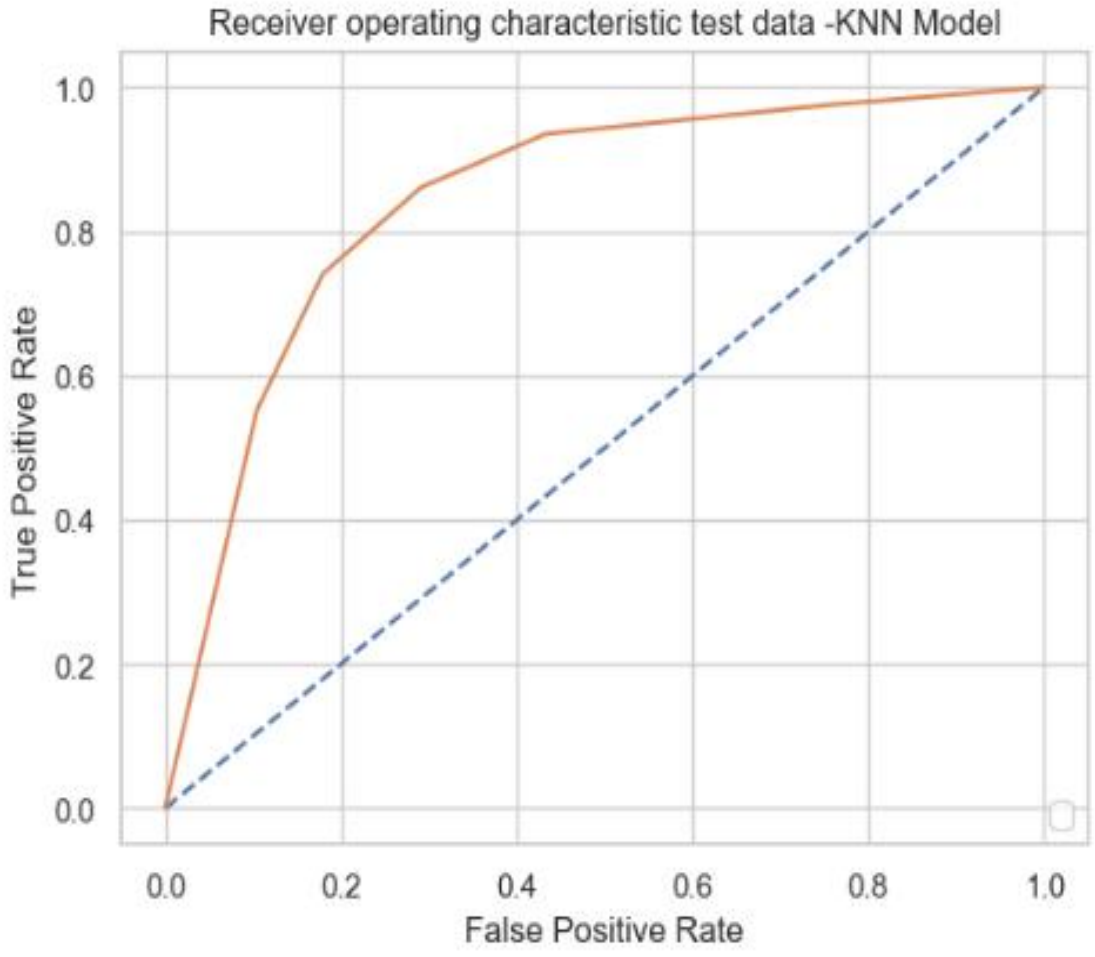
Classification report of test set =

	precision	recall	f1-score	support
False	0.68	0.71	0.69	134
True	0.88	0.86	0.87	324
accuracy			0.82	458
macro avg	0.78	0.79	0.78	458
weighted avg	0.82	0.82	0.82	458

```
KNN_test_precision: 0.88
KNN_test_recall: 0.86
KNN_test_f1: 0.87
```

AUC = 0.844
ROC CURVE =

```
AUC: 0.844
```



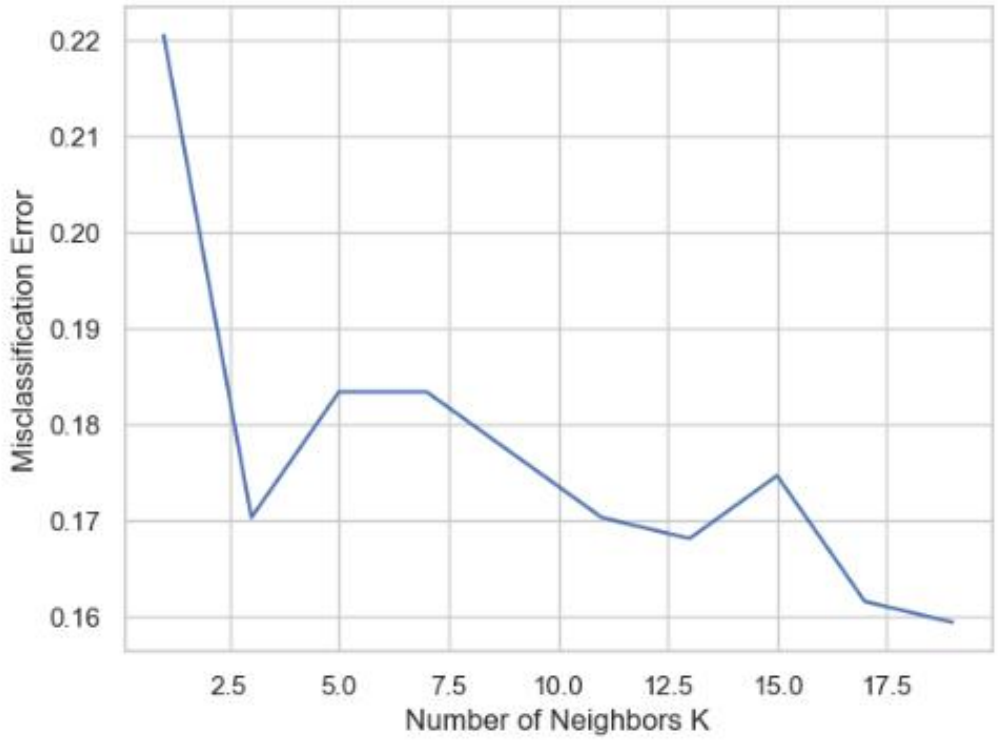
ROC-AUC Scores: Train set 94% and Test set 84%

Now, Run the KNN model with no of neighbours to be 1,3,5..19 and Find the optimal no. of neighbours from K=1,3,5,7....19 using the Mis-classification error

Note: Mis-classification error (MCE) = 1 - Test accuracy score. Calculated MCE for each model with neighbours = 1,3,5...19 and find the model with lowest MCE

Perform the accuracy metrics for values from 1,3,5……19:-

```
[0.22052401746724892,
 0.17030567685589515,
 0.1834061135371179,
 0.1834061135371179,
 0.17685589519650657,
 0.17030567685589515,
 0.16812227074235808,
 0.1746724890829694,
 0.16157205240174677,
 0.15938864628820960]
```



For K =5 it is giving the best test accuracy lets check train and test both with other evaluation metrics

```
In [90]: from sklearn.model_selection import cross_val_score
         scores = cross_val_score(KNN_model, X_train_KNN, y_train_KNN, cv=10)
         scores

Out[90]: array([0.80373832, 0.77570093, 0.85046729, 0.77570093, 0.8411215 ,
                0.79439252, 0.86915888, 0.87735849, 0.80188679, 0.83018868])
```

```
In [91]: scores = cross_val_score(KNN_model, X_test_KNN, y_test_KNN, cv=10)
         scores

Out[91]: array([0.73913043, 0.82608696, 0.76086957, 0.82608696, 0.86956522,
                0.80434783, 0.7826087 , 0.76086957, 0.77777778, 0.88888889])
```

AFTER TUNING:-

KNN Train Score/ Accuracy = 0.86223

```
0.8622305529522024
```

Confusion Matrix of train set =

```
array([[245,  83],
       [ 64, 675]])
```

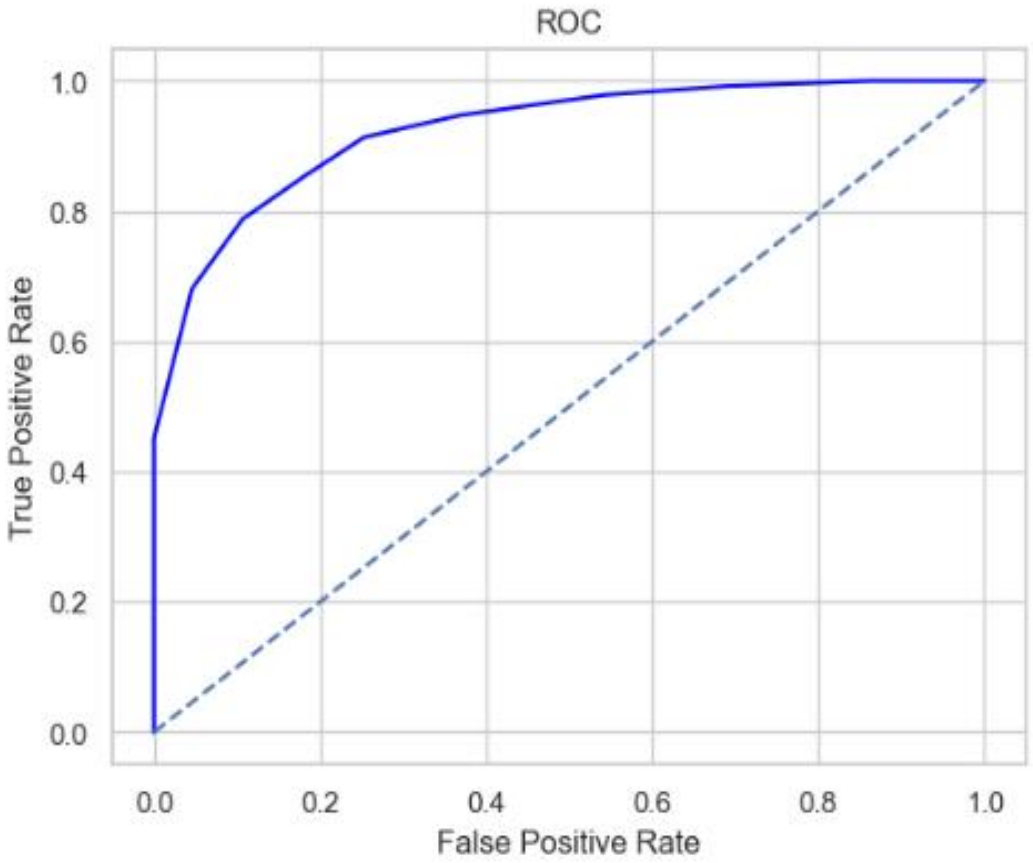
Classification Report of train set =

	precision	recall	f1-score	support
False	0.79	0.75	0.77	328
True	0.89	0.91	0.90	739
accuracy			0.86	1067
macro avg	0.84	0.83	0.84	1067
weighted avg	0.86	0.86	0.86	1067

AUC Score of train set = 0.9247

ROC Curve of train set =

```
Area under Curve is 0.9247644311693456
```



KNN Test Score/ Accuracy = 0.823144

```
0.8231441048034934
```

Confusion matrix of Test set =

```
array([[ 97,  37],
       [ 44, 280]],
```

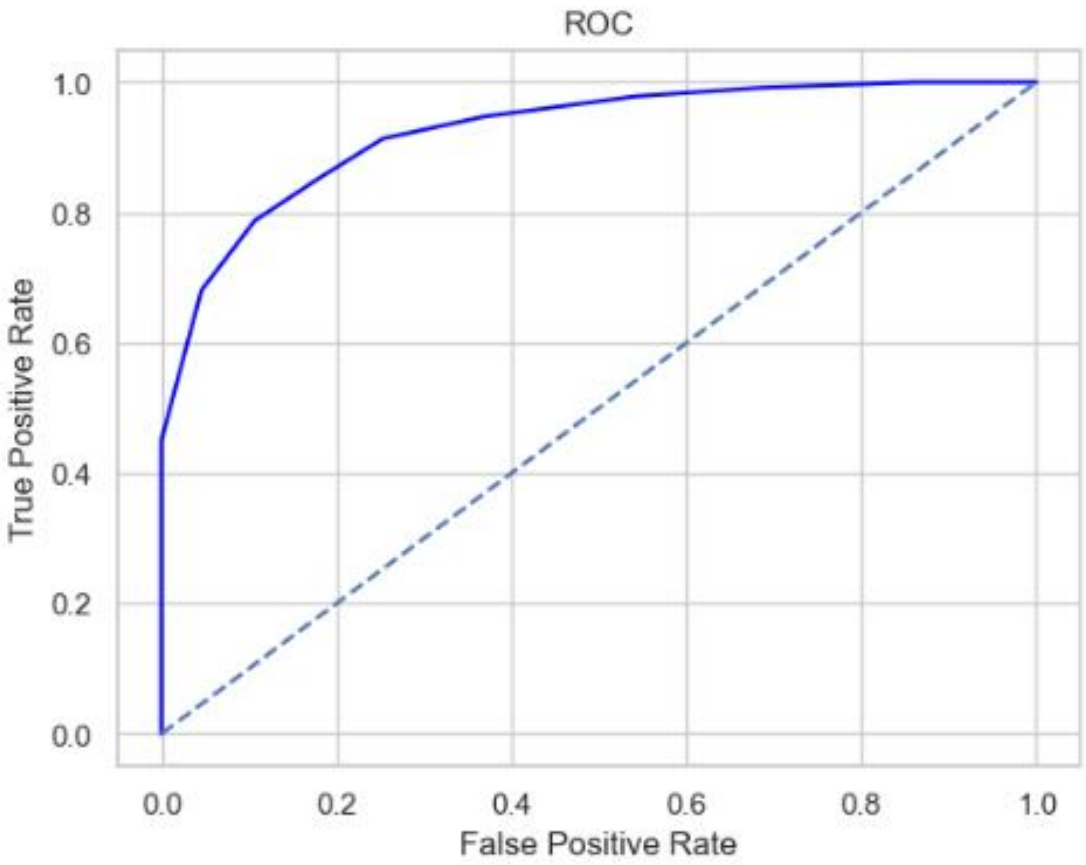
Classification report of test set =

	precision	recall	f1-score	support
False	0.69	0.72	0.71	134
True	0.88	0.86	0.87	324
accuracy			0.82	458
macro avg	0.79	0.79	0.79	458
weighted avg	0.83	0.82	0.82	458

AUC Score of Test set = 0.92476

ROC Curve of Test set =

```
Area under Curve is 0.9247644311693456
```



ROC - AUC Score = Train set is 92% and Test set is 92%

Emphasizing overfitting:

- The large discrepancy between ROC and AUC scores on the training and test sets suggests the KNN model is significantly overfitting. This means the model performs well on training data but generalizes poorly to unseen data.
- Untuned KNN falls victim to overfitting, evident from the substantial difference in performance between training and testing data.

Highlighting the benefits of hyperparameter tuning:

- Tuning key hyperparameters of the KNN model leads to similar accuracy and ROC-AUC scores on both training and test sets, indicating improved generalizability.
- By optimizing key hyperparameters, we successfully mitigate overfitting in the KNN model, resulting in consistent performance across both training and testing data.

Specifying the optimal parameters:

- Fine-tuning KNN reveals optimal performance with 9 neighbors, using a "uniform" weight parameter and a Minkowski distance metric with p=2.
- The best performing KNN configuration utilizes 9 neighbors, "uniform" weighting, and p=2 in the Minkowski distance metric.

B. Naive Bayes Model:-

Model Score/ Accuracy of Train set = 0.8434

```
0.8434864104967198
```

Confusion Matrix of train set =

```
[[238  90]
 [ 77 662]]
```

Classification report of train set =

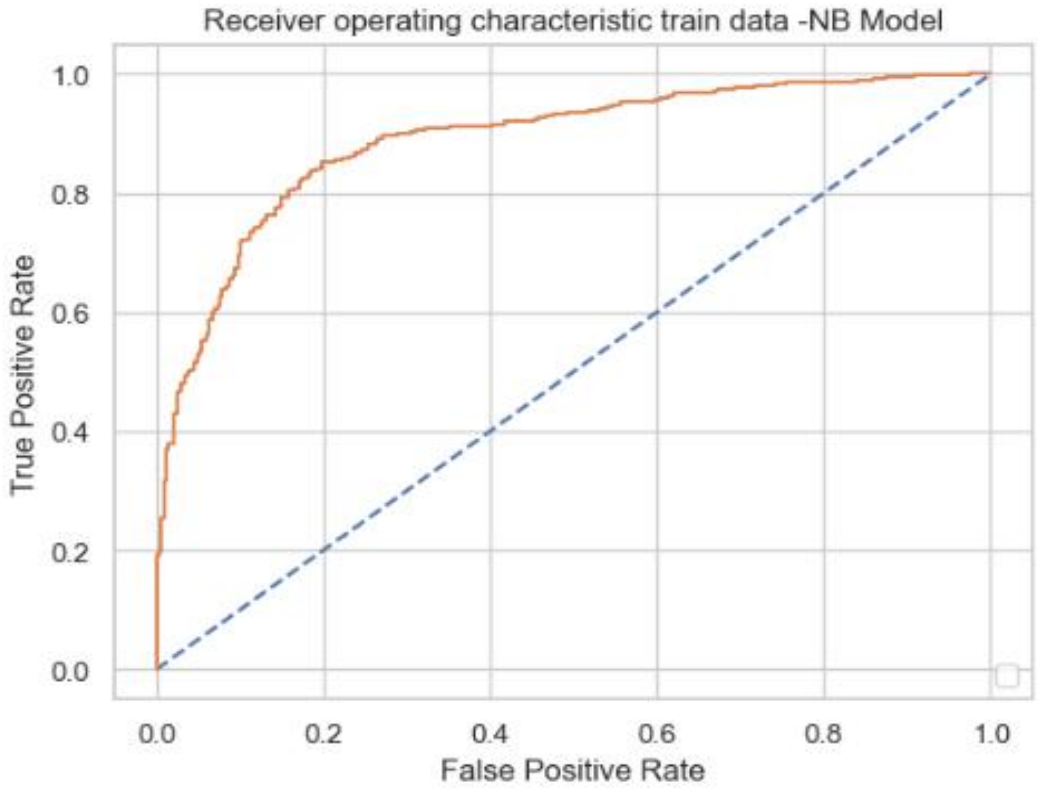
	precision	recall	f1-score	support
False	0.76	0.73	0.74	328
True	0.88	0.90	0.89	739
accuracy			0.84	1067
macro avg	0.82	0.81	0.81	1067
weighted avg	0.84	0.84	0.84	1067

```
NB_train_precision: 0.88
NB_train_recall: 0.9
NB_train_f1: 0.89
```

AUC Score of train set = 0.889

ROC Curve of Train set =

AUC: 0.889



Model Score/ Accuracy of Test set = 0.8362

0.8362445414847162

Confusion Matrix of test set =

```
[[ 99  35]
 [ 40 284]]
```

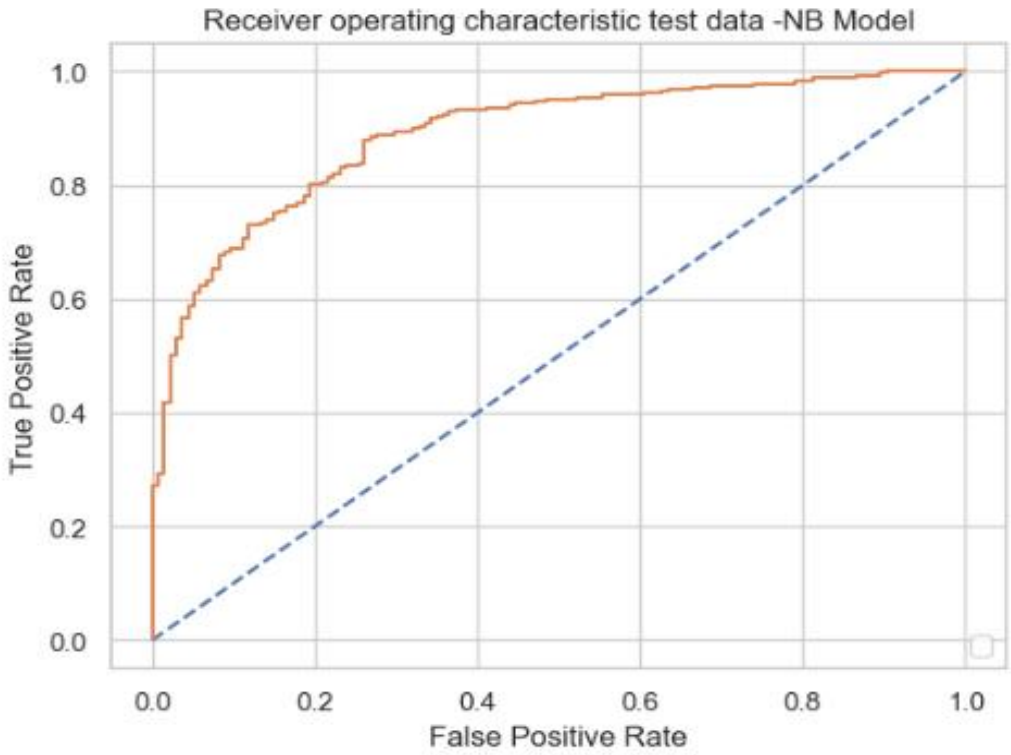
Classification report of test set =

	precision	recall	f1-score	support
False	0.71	0.74	0.73	134
True	0.89	0.88	0.88	324
accuracy			0.84	458
macro avg	0.80	0.81	0.80	458
weighted avg	0.84	0.84	0.84	458

```
NB_test_precision: 0.89
NB_test_recall: 0.88
NB_test_f1: 0.88
```

AUC Score of test set = 0.889

ROC Curve of test set =



ROC - AUC Scores: Train set and Test set has both 88%

KNN AND NAIVE BAYES COMPARISION:-

```
Accuracy on Train set KNN: 0.85
Accuracy on Test set KNN: 0.84
AUC Score on Train set - KNN: 0.9
AUC Score on Test set - KNN: 0.85
Accuracy on Train set Naive Bayes: 0.85
Accuracy on Test set Naive Bayes: 0.82
AUC Score on Train set - Naive Bayes: 0.89
AUC Score on Test set - Naive Bayes: 0.88
```

C. Bagging:-

- RAMDOM FOREST

Model Score/ Accuracy of Train set = 0.999062

```
0.9990627928772259
```

Confusion Matrix of Train set =

```
[[330   1]
 [  0 736]]
```

Classification Report of Train set =

	precision	recall	f1-score	support
False	1.00	1.00	1.00	331
True	1.00	1.00	1.00	736
accuracy			1.00	1067
macro avg	1.00	1.00	1.00	1067
weighted avg	1.00	1.00	1.00	1067

Model Score/ Accuracy of Test set = 0.82969

```
0.8296943231441049
```

Confusion Matrix of Test set =

```
[[ 88  43]
 [ 35 292]]
```

Classification Report of Test set =

	precision	recall	f1-score	support
False	0.72	0.67	0.69	131
True	0.87	0.89	0.88	327
accuracy			0.83	458
macro avg	0.79	0.78	0.79	458
weighted avg	0.83	0.83	0.83	458

- BAGGING

Model Score/ Accuracy of Train set = 0.9643

```
0.9643861293345829
```

Confusion Matrix of Train set =

```
[[304  27]
 [ 11 725]]
```

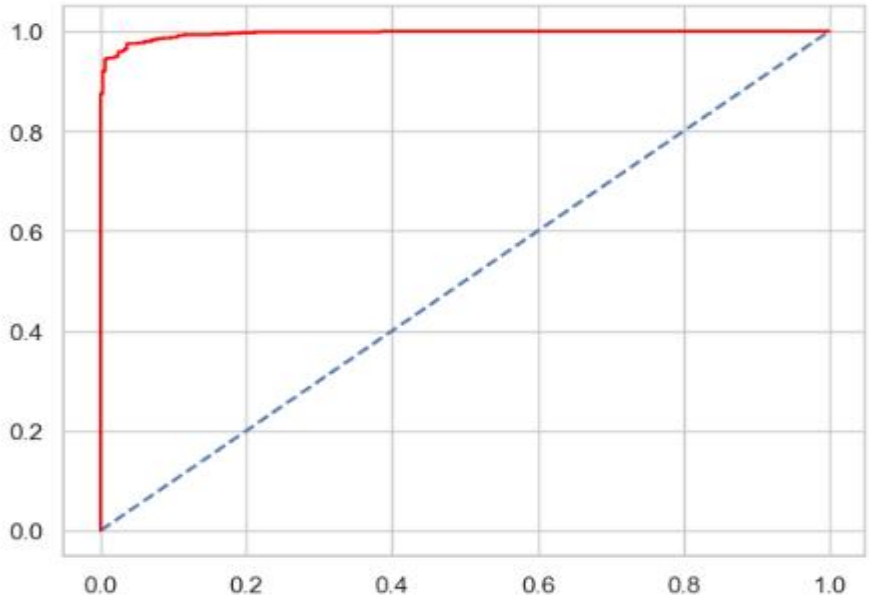
Classification Report of Train set =

	precision	recall	f1-score	support
False	0.97	0.92	0.94	331
True	0.96	0.99	0.97	736
accuracy			0.96	1067
macro avg	0.96	0.95	0.96	1067
weighted avg	0.96	0.96	0.96	1067

AUC For Train set = 0.9958

ROC For Train set =

```
AUC for the Train  (Bagging) :  0.9958582359122554
[<matplotlib.lines.Line2D at 0x200d7ec1b10>]
```



Model Score/ Accuracy of Test set = 0.8363

```
0.8362445414847162
```

Confusion Matrix of Test set =

```
[[ 86  45]
 [ 30 297]]
```

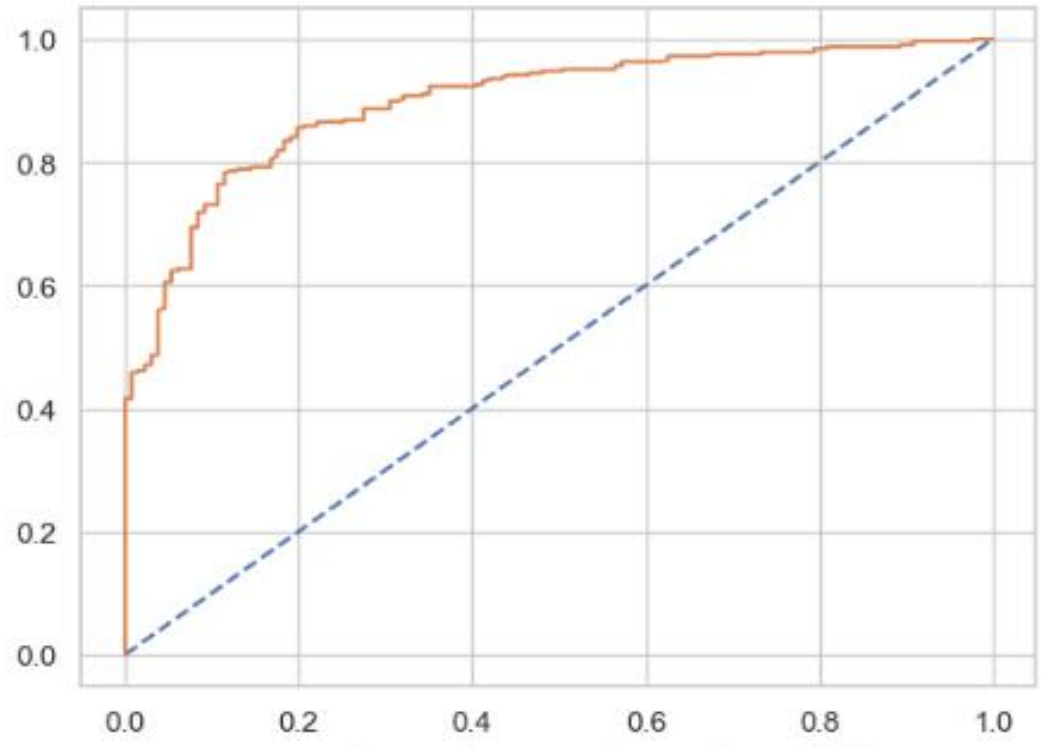
Classification Report of Test set =

	precision	recall	f1-score	support
False	0.74	0.66	0.70	131
True	0.87	0.91	0.89	327
accuracy			0.84	458
macro avg	0.80	0.78	0.79	458
weighted avg	0.83	0.84	0.83	458

AUC For Train set = 0.89840

ROC For Train set =

```
AUC for the Test  (Bagging) :  0.8984055839577935
[<matplotlib.lines.Line2D at 0x200d6e0fbd0>]
```



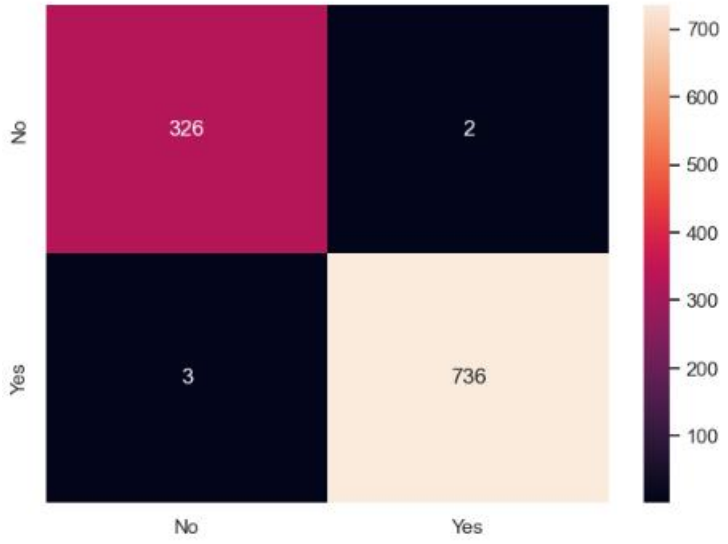

```
Accuracy on Train set Bagging Classifier : 0.96
Accuracy on Test set Bagging Classifier : 0.84
```

D. BOOSTING:-

Model Score/ Accuracy of Train set = 0.99531

```
0.9953139643861293
```

Confusion Matrix of Train Set =



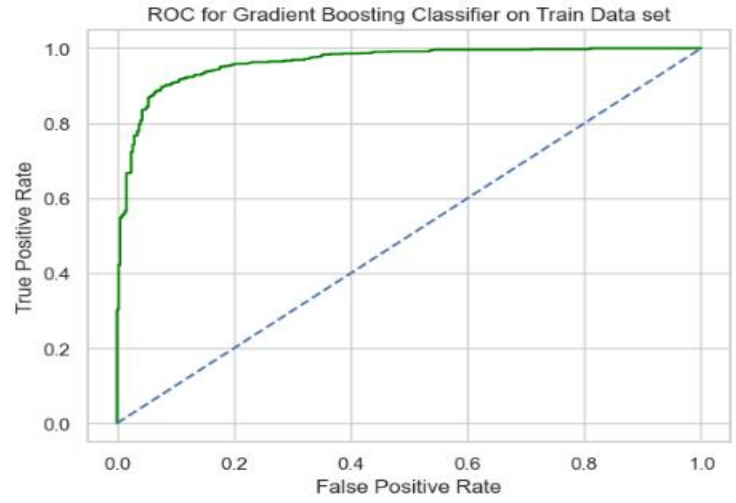
Classification Report of Train Set =

	precision	recall	f1-score	support
False	0.87	0.82	0.85	373
True	0.92	0.95	0.94	847
accuracy			0.91	1220
macro avg	0.90	0.89	0.89	1220
weighted avg	0.91	0.91	0.91	1220

AUC of Train set = 0.9634

ROC Curve of Test set =

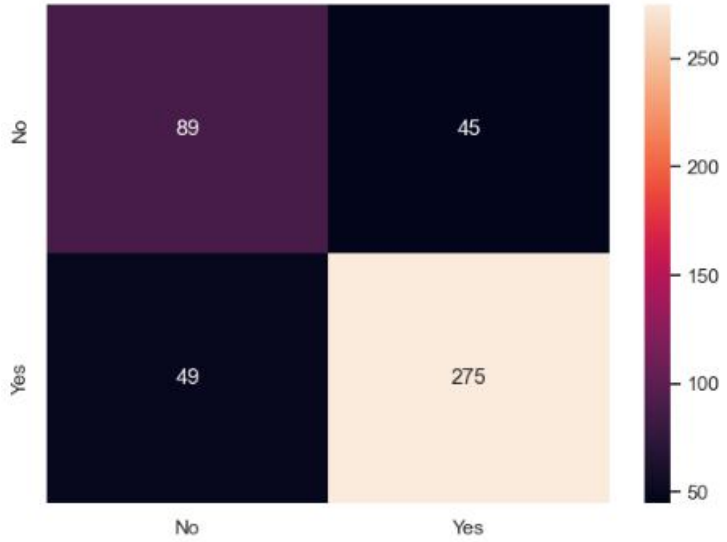
```
Area under Curve Gradient Boosting Classifier is 0.9634730368339922
```



Model Score/ Accuracy of Test set = 0.7947

```
0.7947598253275109
```

Confusion Matrix of Test Set =



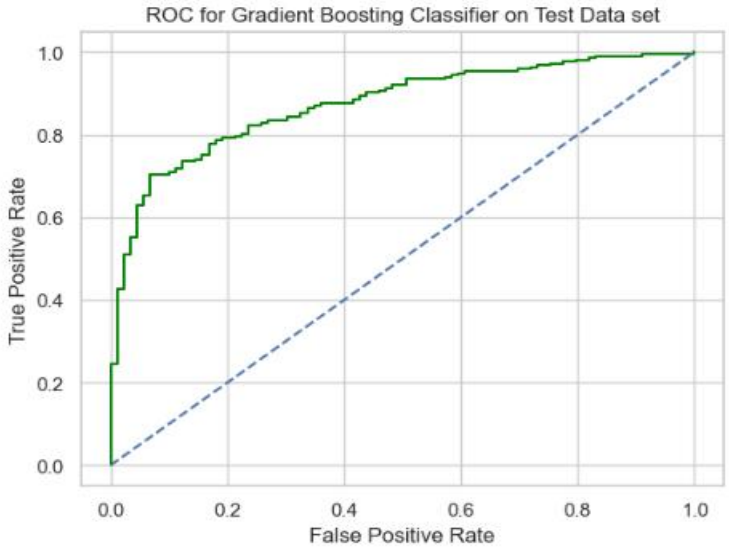
Classification report of Test set =

	precision	recall	f1-score	support
False	0.67	0.58	0.62	89
True	0.84	0.88	0.86	216
accuracy			0.79	305
macro avg	0.75	0.73	0.74	305
weighted avg	0.79	0.79	0.79	305

AUC of Test set = 0.8737

```
Area under Curve Gradient Boosting Classifier is 0.8737515605493134
```

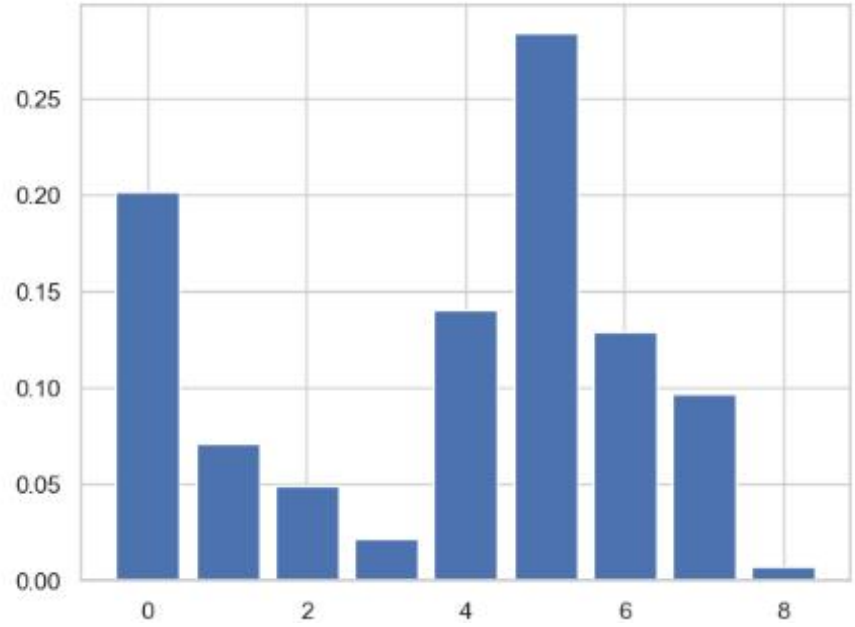
ROC of Test set =



ROC - AUC Scores: Train set 96% and Test set 87%

Calculating the score for all input features in a model to establish the importance of each feature in the decision-making process:-

```
Feature: 0, Score: 0.20179
Feature: 1, Score: 0.07087
Feature: 2, Score: 0.04898
Feature: 3, Score: 0.02130
Feature: 4, Score: 0.14029
Feature: 5, Score: 0.28412
Feature: 6, Score: 0.12895
Feature: 7, Score: 0.09664
Feature: 8, Score: 0.00705
```



1. Gender and economic condition (household) have minimal impact on predicting votes:
-The influence of both gender and household economic condition on vote prediction is negligible.
-These factors contribute very little to understanding voting patterns, based on the model.
-While other variables are more significant, gender and household financial status play a minor role in influencing votes.

2. Age remains a strong predictor of votes:
-Age continues to be a relevant factor in understanding voting behavior.
-The model finds that age is still a crucial variable for predicting votes.
-Amongst the analyzed variables, age retains its significance in influencing how people vote.

3. Untuned model exhibits overfitting:
-The model without hyperparameter tuning shows signs of overfitting.
-The unoptimized model struggles to generalize well, indicating overfitting.
-The lack of tuning has led to overfitting, which limits the model's performance on unseen data.

1.5 Model Performance Improvement:-

- **MODEL TUNING:**

A. BOOSTING

Model Score/ Accuracy of Train set = 0.90

Confusion Matrix of Train set =

```
array([[273,  55],
       [ 47, 692]]
```

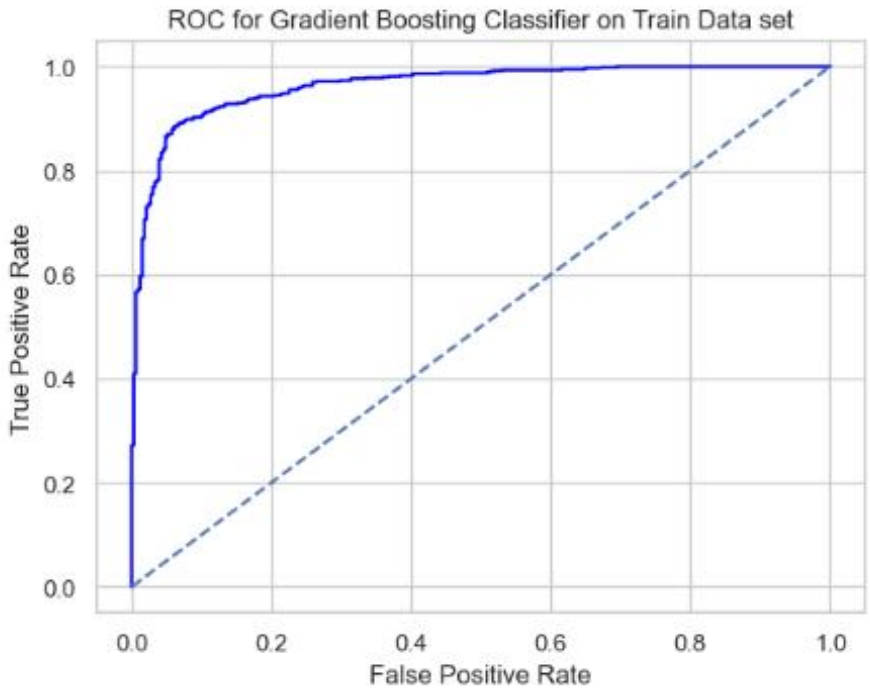
Classification Report of Train set =

	precision	recall	f1-score	support
False	0.85	0.83	0.84	328
True	0.93	0.94	0.93	739
accuracy			0.90	1067
macro avg	0.89	0.88	0.89	1067
weighted avg	0.90	0.90	0.90	1067

AUC of Train set = 0.9630

ROC of Train set =

```
Area under Curve Gradient Boosting Classifier is 0.9630433347635237
```



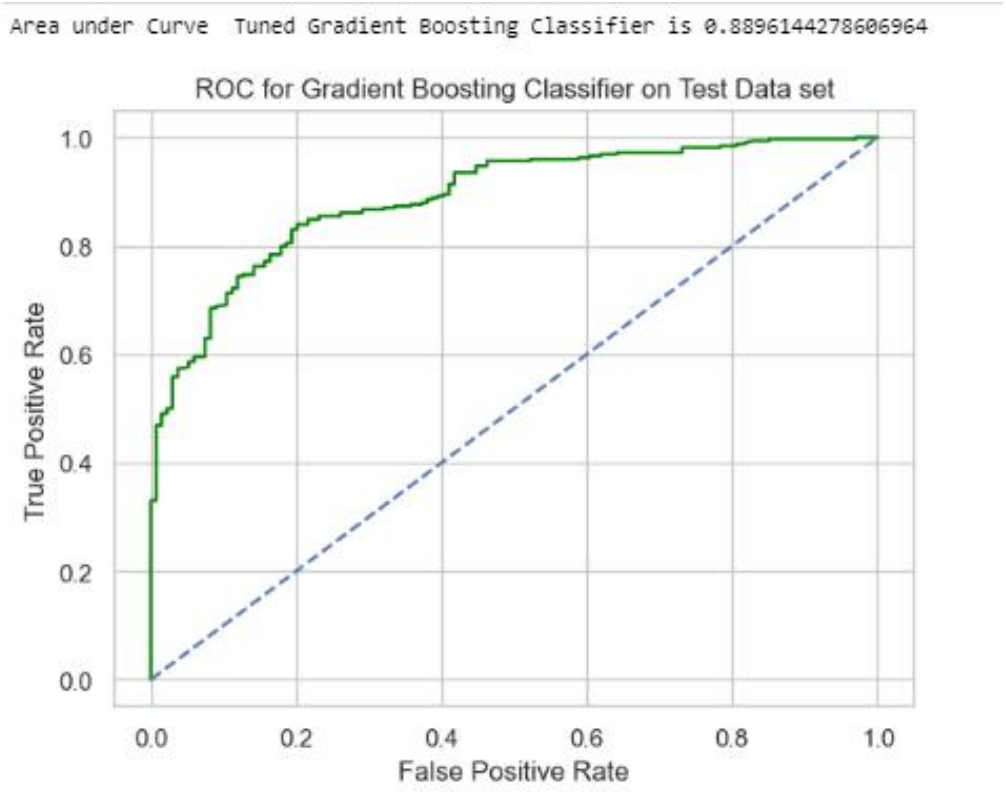
Model Score/ Accuracy of Test set = 0.82

Classification Report of Test set =

	precision	recall	f1-score	support
False	0.69	0.71	0.70	134
True	0.88	0.87	0.87	324
accuracy			0.82	458
macro avg	0.78	0.79	0.79	458
weighted avg	0.82	0.82	0.82	458

AUC of Test set = 0.889614

ROC of Test set =



B. BAGGING

Model Score/ Accuracy of Train set = 0.904404

0.9044048734770385

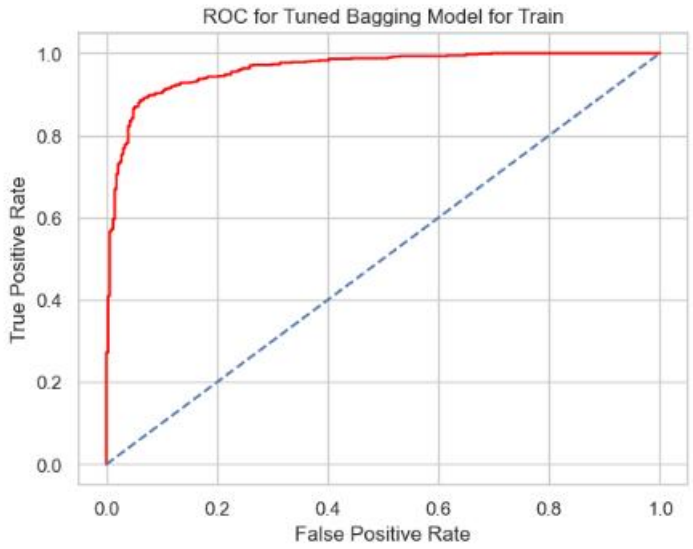
Classification Report of Train set =

	precision	recall	f1-score	support
False	0.85	0.83	0.84	328
True	0.93	0.94	0.93	739
accuracy			0.90	1067
macro avg	0.89	0.88	0.89	1067
weighted avg	0.90	0.90	0.90	1067

AUC of Train set = 0.9630

ROC of Train set =

```
Area under Curve is 0.9630433347635237
```



Model Score / Accuracy of Test set = 0.82

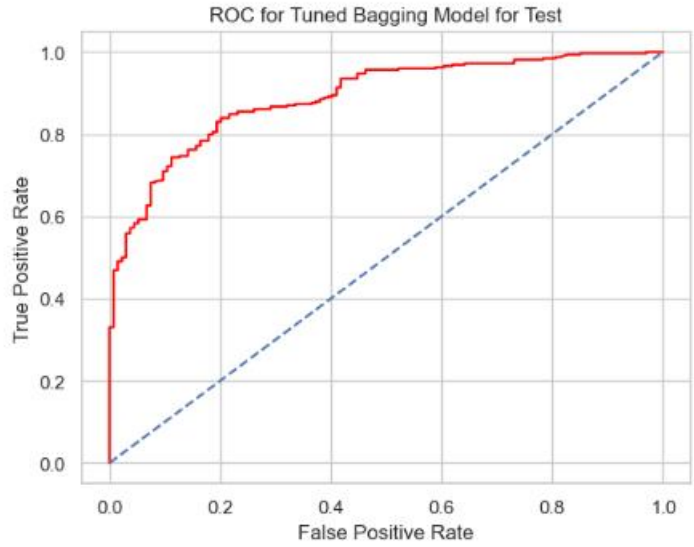
Classification Report of Test set =

	precision	recall	f1-score	support
False	0.69	0.71	0.70	134
True	0.88	0.87	0.87	324
accuracy			0.82	458
macro avg	0.78	0.79	0.79	458
weighted avg	0.82	0.82	0.82	458

AUC of Test set = 0.8906

ROC of Test set =

```
Area under Curve is 0.890650912106136
```



ROC - AUC Score: Train set is 96% and Test set is 89%

1. Overfitting in Untuned Bagging Classifier:

- The lack of hyperparameter tuning leads to severe overfitting in the bagging classifier, as evidenced by the substantial gap between training and test set accuracy and ROC-AUC scores. This indicates the model memorizes training data poorly generalizing to unseen data.
- Without adjusting hyperparameters, the bagging classifier heavily overfits, resulting in a significant performance difference between training and testing, rendering it unreliable for general use.

2. Hyperparameter Tuning Rescues Overfitting:

- Hyperparameter tuning effectively addresses the overfitting issue, leading to a generalized model with consistent accuracy and ROC-AUC scores across both training and test sets. This signifies the model's ability to perform well on unseen data.
- Optimization of hyperparameters successfully mitigates overfitting in the bagging classifier, yielding a robust and generalizable model with near-identical performance on both training and testing data.

3. Hague Identified as Most Important Variable:

- According to the tuned bagging model, the "Hague" variable holds the most prominent influence on the predicted outcome. This suggests a strong association between this and the target variable.
- Hyperparameter tuning reveals "Hague" as the critical predictor within the bagging model, implying a substantial impact on the predicted outcome. Further investigation into this variable's nature and relationship with the target variable is recommended.

1.6 & 1.7 Final Model Selection & Actionable Insights & Recommendations

A. Compare all the model built:-

- **Before Tuning**

CONTENT	KNN	NAIVE BAYES	BOOSTING	BAGGING
Accuracy of Test set	0.84	0.82	0.79	0.83
Accuracy of Train set	0.85	0.85	0.99	0.96
AUC Score of Test set	0.85	0.88	0.87	0.84
AUC Score of Train set	0.90	0.89	0.96	0.99
Precision of Test set	0.88	0.89	0.84	0.87
Precision of Train set	0.90	0.88	0.92	0.96
Re-call of Test set	0.86	0.88	0.88	0.91

Re-call of Train	0.92	0.90	0.95	0.89
F1-Score of Test set	0.87	0.88	0.86	0.89
F1-Score of Train set	0.91	0.89	0.94	0.97

- **After Tuning**

CONTENT	BOOSTING	BAGGING
Accuracy of Test set	0.82	0.82
Accuracy of Train set	0.90	0.90
AUC Score of Test set	0.88	0.89
AUC Score of Train set	0.96	0.96
Precision of Test set	0.88	0.88
Precision of Train set	0.93	0.90
Re-call of Test-set	0.87	0.87
Re-call of Train set	0.94	0.94
F1-Score of Test set	0.87	0.87
F1-Score of Train Set	0.93	0.93

B. Select the final model with the proper justification

Before Tuning:

- Both Bagging and Boosting exhibit relatively high accuracy and AUC scores on the test set, with Bagging having slightly better precision and recall for class 1 (assuming it represents the positive class).
- Boosting shows significant overfitting with a large gap between training and test set performance in most metrics, despite higher training scores.
- Naive Bayes lags behind in various metrics compared to Bagging and Boosting.

After Tuning:

- Both Boosting and Bagging achieve similar performance across all metrics on the test set, indicating strong generalizability after addressing overfitting issues in Boosting.
- Precision remains slightly higher for Bagging, while recall for class 1 remains marginally better for Boosting. F1-scores are almost identical.

Considerations:

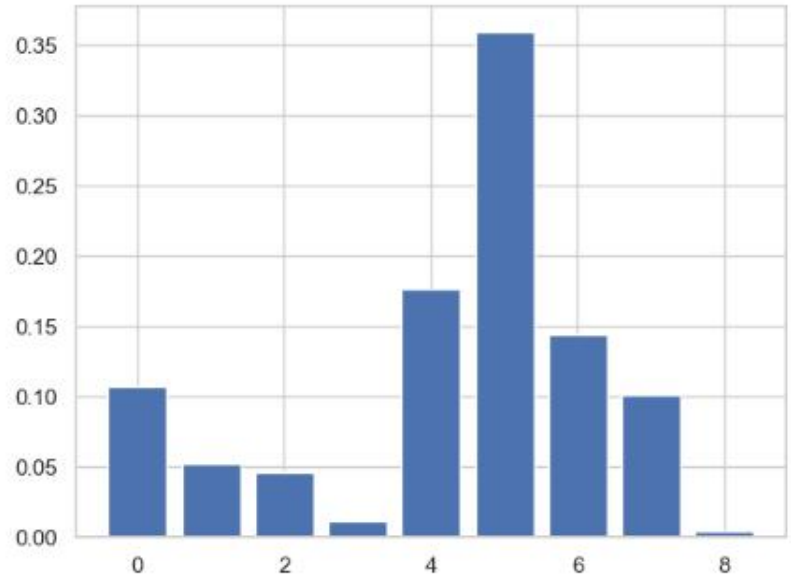
Class Imbalance: If class imbalance exists, the choice might lean towards the model performing better on the minority class (likely class 1 based on the higher recall).
Computational Resources: If training and prediction time are crucial, Bagging might be preferable due to its generally faster performance.
Interpretability: If understanding the model's decisions is essential, Bagging offers a simpler structure compared to Boosting's complexity.

BAGGING might be preferred choice in this specific scenario.

C. Important features in the final model and draw inferences:-

Calculating the score for all input features in a model to establish the importance of each feature in the decision-making process:-

```
Feature: 0, Score: 0.10683
Feature: 1, Score: 0.05175
Feature: 2, Score: 0.04531
Feature: 3, Score: 0.01085
Feature: 4, Score: 0.17667
Feature: 5, Score: 0.35945
Feature: 6, Score: 0.14364
Feature: 7, Score: 0.10113
Feature: 8, Score: 0.00439
```



B. 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:-

President Franklin D. Roosevelt in 1941;
President John F. Kennedy in 1961
President Richard Nixon in 1973

2.1 Define the problem and Perform Exploratory Data Analysis:

A. Find the number of Character:-

```
print(len(inaugural.raw('1941-Roosevelt.txt')))
```

```
7571
```

The number of character in '1941 Roosevelt.txt'of Inaugural is 7571.

```
print(len(inaugural.raw('1961-Kennedy.txt')))
```

```
7618
```

The number of character in "1961 Kennedy.txt' of Inaugural is 7618.

```
print(len(inaugural.raw('1973-Nixon.txt')))
```

```
9991
```

The number of character in '1961 Kennedy.txt'of Inaugural is 9991.

B. Find the number of Words:-

```
print(len(inaugural.words('1941-Roosevelt.txt')))
```

```
1536
```

The number of words in '1941-Roosevelt.txt' are 1536.

```
print(len(inaugural.words('1961-Kennedy.txt')))
```

```
1546
```

The number of words in '1961-Kennedy.txt' are 1546

```
print(len(inaugural.words('1973-Nixon.txt')))
```

```
2028
```

The number of words in '1973-Nixon.txt' are 2028.

C. Find the Number of Sentences:-

```
print(len(inaugural.sents('1941-Roosevelt.txt')))
```

```
68
```

The number of sentences in '1941-Roosevelt' are 68.

```
print(len(inaugural.sents('1961-Kennedy.txt')))
```

```
52
```

The number of sentences in '1961-Kennedy.txt' are 52.

```
print(len(inaugural.sents('1973-Nixon.txt')))
```

```
69
```

The number of sentences in '1973-Nixon.txt' are 69.

2.2 Text Cleaning:

A. Stop-word removal & Stemming:-

-We used nltk library's stopwords module For get all "english" stopwords.

-Then we run the command to build a new list of words for each speech (Clean list) which does not contain stop words.

-We used nltk library's Porter Stemmer For get Stemming.

Stopwords Removal

```
import random
import string
from nltk.corpus import stopwords
nltk.download('stopwords')
from nltk.stem.porter import PorterStemmer
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\rahul\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Create a object named PS for the PorterStemmer

```
PS = PorterStemmer()
```

```
stopwords = nltk.corpus.stopwords.words('english') +list(string.punctuation)
stopwords.extend(["--"]) #,"us"])
```

Removing the stopword on '1941-Roosevelt's.txt' speech:

```
Roosevelt_lower = (w.lower() for w in inaugural.words('1941-Roosevelt.txt'))
Roosevelt_clean = [word for word in Roosevelt_lower if word not in stopwords]
Roosevelt_clean_stem = [PS.stem(word) for word in Roosevelt_clean]
Roosevelt_freq = nltk.FreqDist(Roosevelt_clean_stem)
```

Removing the stopword on '1961-Kennedy's.txt'speech:

```
Kennedy_lower = (w.lower() for w in inaugural.words('1961-Kennedy.txt'))
Kennedy_clean = [word for word in Kennedy_lower if word not in stopwords]
Kennedy_clean_stem = [PS.stem(word) for word in Kennedy_clean]
Kennedy_freq = nltk.FreqDist(Kennedy_clean_stem)
```

Removing the stopword on'1973-Nixon's.txt'speech:

```
Nixon_lower = (w.lower() for w in inaugural.words('1973-Nixon.txt'))
Nixon_clean = [word for word in Nixon_lower if word not in stopwords]
Nixon_clean_stem = [PS.stem(word) for word in Nixon_clean]
Nixon_freq = nltk.FreqDist(Nixon_clean_stem)
```

```
df_stopwords = pd.DataFrame(stopwords)
df_stopwords.head(5)
```

	0
0	i
1	me
2	my
3	myself
4	we

B. Find the 3 most common words used in all three speeches:-

```
print("The top three words in Roosevelt's Speech are :\n",Roosevelt_freq.most_common(3))
```

```
The top three words in Roosevelt's Speech are :
 [('nation', 17), ('know', 10), ('peopl', 9)]
```

```
print("The top three words in Kennedy's Speech are :\n",Kennedy_freq.most_common(3))
```

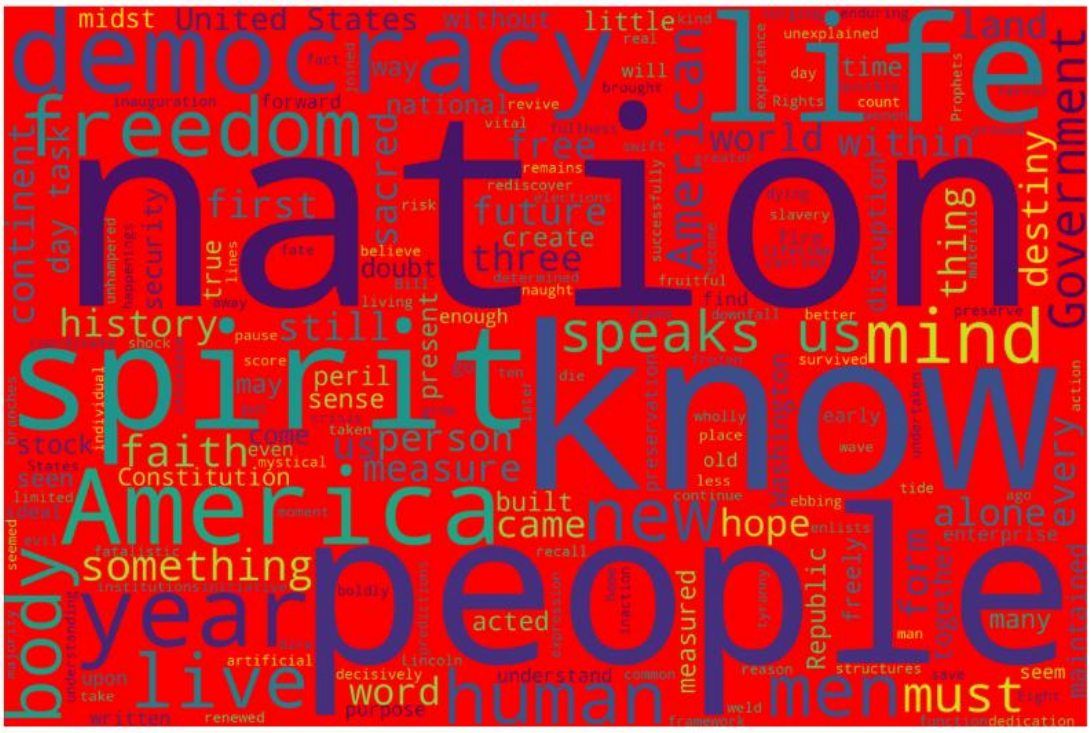
```
The top three words in Kennedy's Speech are :
 [('let', 16), ('us', 12), ('power', 9)]
```

```
print("The top three words in Nixon's Speech are :\n",Nixon_freq.most_common(3))
```

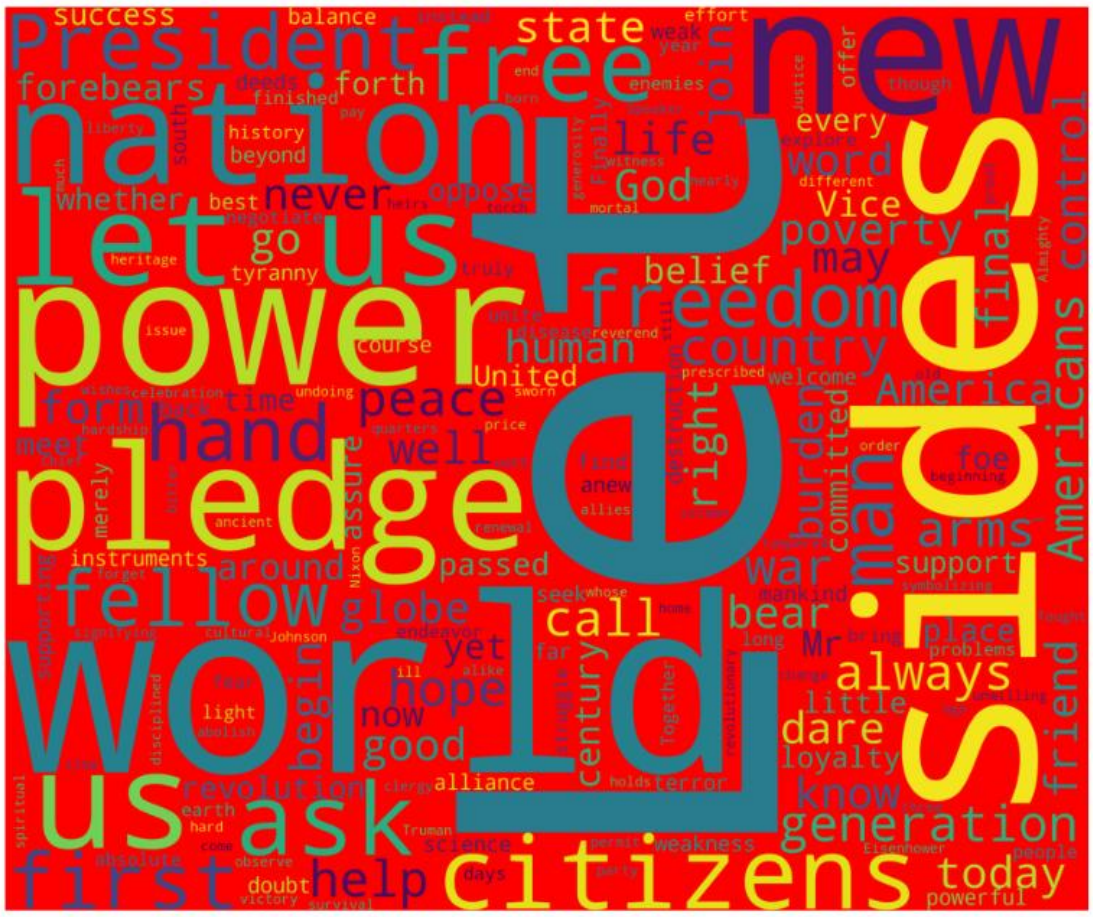
```
The top three words in Nixon's Speech are :
 [('us', 26), ('let', 22), ('america', 21)]
```

2.3 Plot Word cloud of all three speeches:

A. Word cloud for President "Rossevelt" inaugural address in 1941.



B. Wordcloud for President "Kennedy" inaugural address in 1961.



C. Word cloud for President "Nixon" inaugural address in 1973