Great Learning

MACHINE LEARNING

PROJECT REPORT

Student Name – Arpit Doneria

PGP-DSBA

Problem 1:

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

- **Read the dataset. Do the descriptive statisticsand do the null value condition check. Write an inference on it.**

<class 'pandas.core.frame.DataFrame'>

Int64Index: 1525 entries, 1 to 1525

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: 119.1+ KB

Data Information:

Observation:

- We have dropped the 'unnamed' column from the datasetas it isnot useful for our study.
- The dataset had 8 duplicated values. So, we are droppedthem.
- The data set had 1525 rows and 9 columns. After dropping the duplicate values, there are 1517 rows and 9columns.
- It has 7 numerical data types and 2 categorical data types.
- There is no null value in any column.

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

Checking for missing values:

There are no missing values.

8Checking for duplicated values:

There are 8 duplicated values. So, we are dropping them.

Data description:

Checking the skewness of the data:

The rule of thumb of skewness seems to be:

- If the skewness is between -0.5 and 0.5, the data arefairly symmetrical.
- If the skewness is between -1 and – 0.5 or between 0.5and1, the data are moderately skewed.
- If the skewness is less than -1 or greater than 1, the dataare highly skewed.

Insights:

- Here, we can see that there isn't much skewness in thedata.All the values seems to be between -0.5 and 0.5.
- The value of 'Blair' is a little bit higher than -0.5.
- The data overall, is fairly symmetrical.

- **Perform Univariate and Bivariate Analysis. Doexploratory data analysis. Check for Outliers.**

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

Exploratory Data Analysis:Null value check:

There are no null values present in the data.

Data types:

There are 7 numerical and 2 categorical data types in the data.

Shape of the data:

```
no. of rows: 1517
no. of columns: 9
```There are 1517 rows and 9 columns in the data.

## Univariate

## Analysis

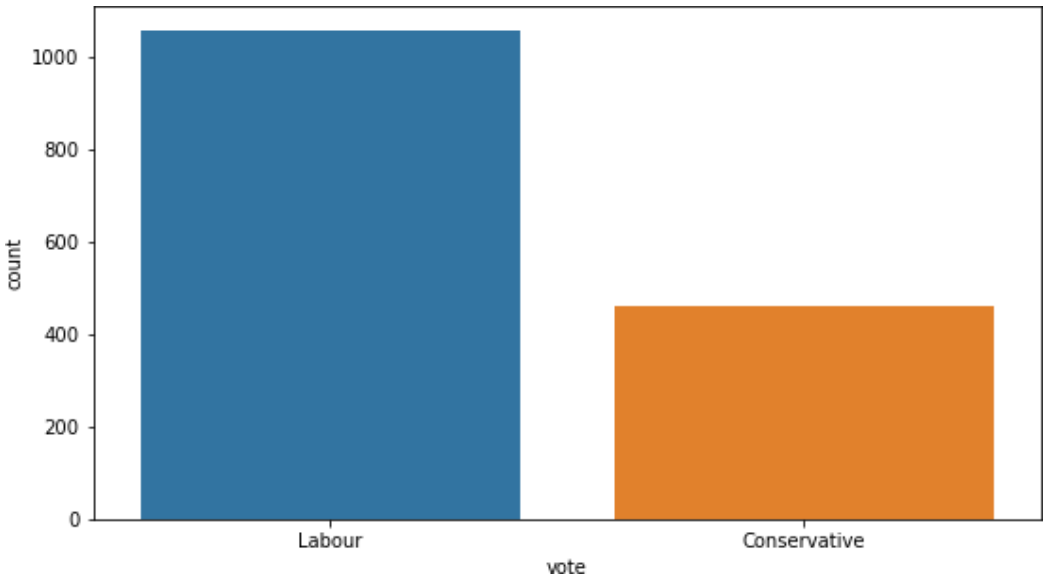
## s:Description of 'age':

## Histogram and box plot of 'age':

## Observation:

- The data is normally distributed.
- Maximum number of people are aged between 40 and 70.
- Outliers are not present.
- The minimum value is 24 and the maximum value is 93.
- The mean value is 54.241266

## Count plot of 'vote':




```
Labour          1057
Conservative     460
Name: vote, dtype: int64
```

Viewing the exact values of the variables of 'vote':

Observation:

- Labour party has higher number of votes. It has morethan doublethe votes of conservative party.
- Labour party has 1057 votes.
- Conservative party has 460 votes.

Count plot of 'economic.cond.national':

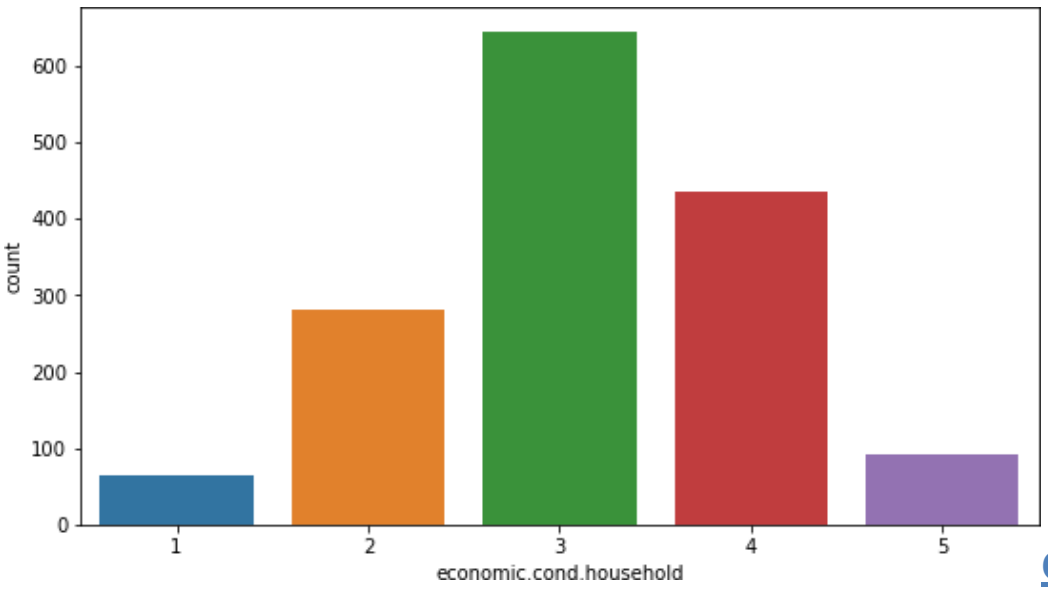
Viewing the exact values of the variables of'economic.c

Mean of 'economic.cond.national':

Observation:

- The top 2 variables are 3 and 4.
- 1 has the least value which is 37.
- 3 has the highest value which is 604.

- 3 is slightly higher than the 2nd highest variable 4 whosevalueis 538.
- The average score of 'economic.cond.national' is3.245221



Count plot of 'economic.cond.household':

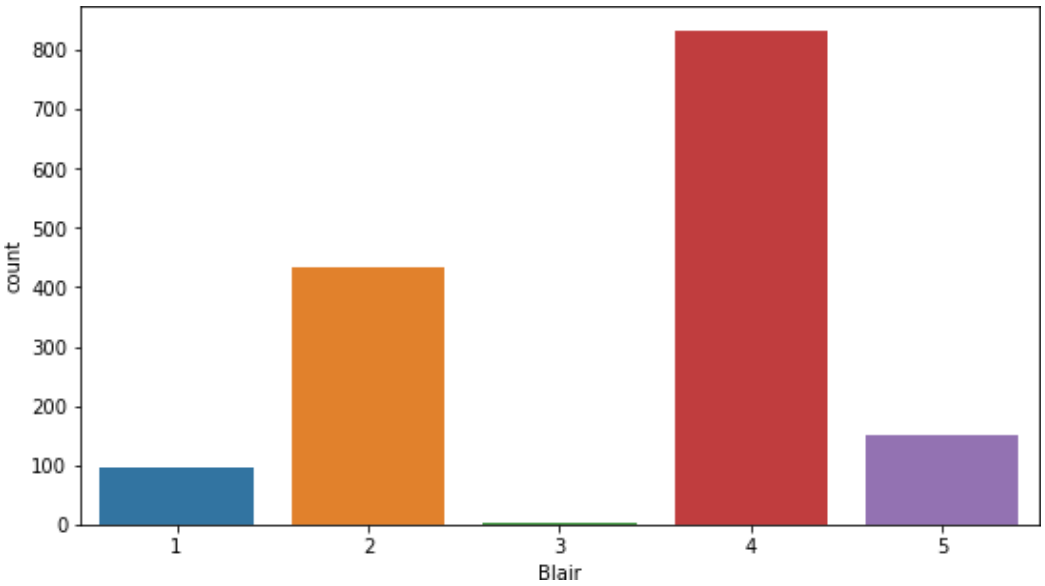
Viewing the exact values of the variables of'economic.c

Mean of 'economic.cond.household':

Observation:

- The top 2 variables are 3 and 4.
- 1 has the least value which is 65.
- 3 has the highest value which is 645.
- 3 is moderately higher than the 2nd highest variable 4whosevalue is 435.
- The average score of 'economic.cond.household' is3.137772

Count plot of 'Blair':



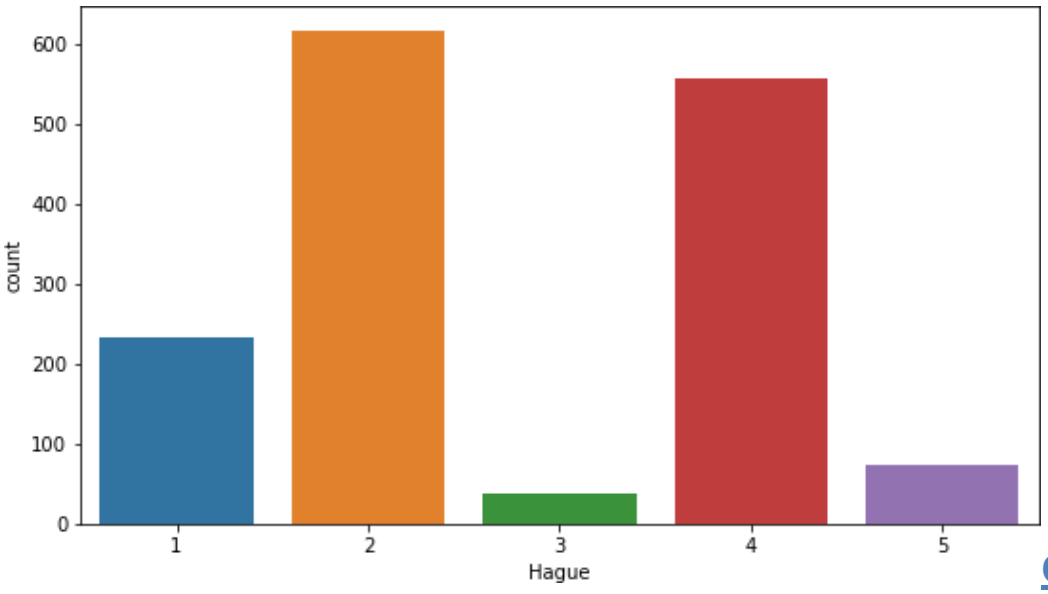
Viewing the exact values of the variables of 'Blair':

Mean of 'Blair':

Observation:

- The top 2 variables are 2 and 4.
- 3 has the least value which is 1.
- 4 has the highest value which is 833.
- 4 is much higher than the 2nd highest variable 2 whosevalueis 434.

- The average score of 'Blair' is 3.335531



Count plot of 'Hague':

Viewing the exact values of the variables of'Hague':

Mean of 'Hague':

Observation:

- The top 2 variables are 2 and 4.
- 3 has the least value which is 37.
- 2 has the highest value which is 617.

- 2 is slightly higher than the 2nd highest variable 4 whosevalueis 557.
- The average score of 'Blair' is 2.749506

Count plot of 'Europe':

Viewing the exact values of the variables of'Europe':

Mean of 'Europe':

Observation:

- The top 2 variables are 11 and 6.
- 2 has the least value which is 77.
- 11 has the highest value which is 338.
- 11 is moderately higher than the 2nd highest variable 6whosevalue is 207.
- The average score of 'Europe' is 6.740277

Count plot of 'political.knowledge':

Viewing the exact values of the variables of'political.kn

Mean of 'Europe':

Observation:

- The top 2 variables are 2 and 0.
- 1 has the least value which is 38.
- 2 has the highest value which is 776.
- 2 is much higher than the 2nd highest variable 0 whose value is454.
- We can see that, 454 out of 1517 people do not have any knowledge of parties' positions on European integration which is 29.93% of the total population.
- The average score of 'Europe' is 6.740277

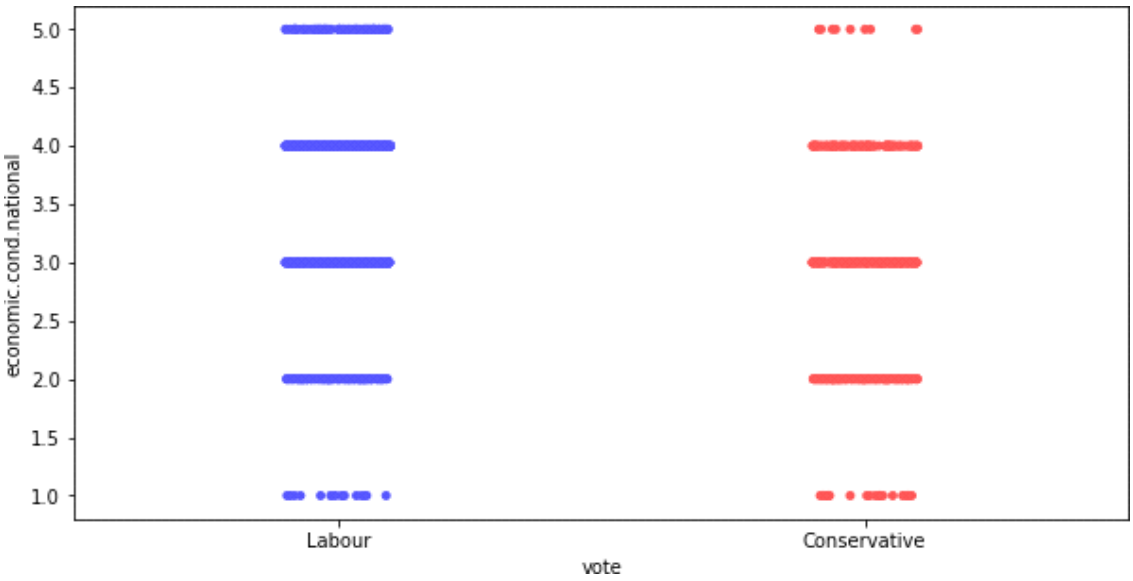
Bivariate Analysis:

Strip plot of 'vote' and 'age':

Viewing the exact values of the variables of 'vote'withrespect to 'gender':

Observation:

- We can clearly see that, the labour party has got morevotesthan the conservative party.
- In every age group, the labour party has got more votesthanthe conservative party.
- Female votes are considerably higher than the male votesin both parties.
- In both genders, the labour party has got more votes thanthe conservative party.



Strip plot of 'vote' and 'economic.cond.national':

Viewing the exact values of the variables of 'vote'withrespect to 'economic.cond.national':

Observation:

- Labour party has higher votes overall.
- Out of 82 people who gave a score of 5, 73 people have voted for the labour party.
- Out of 538 people who gave a score of 4, 447 people have voted for the labour party. This is the highest set of people in the labour party.

- Out of 604 people who gave a score of 3, 405 people have voted for the labour party. This is the 2nd highest set of people in the labour party. The remaining 199 people who have voted for the conservative party is the highest set of people in that party.
- Out of 256 people who gave a score of 2, 116 people have voted for the labour party. 140 people have voted for the conservative party. This is the instance where the conservative party has got more votes than the labourparty.
- Out of 37 people who gave a score of 1, 16 people have voted forthe labour party. 21 people have voted for the conservative party.
- The score of 3, 4 and 5 have more votes in the labour party.
- The score of 1 and 2 have more votes in the conservativeparty.

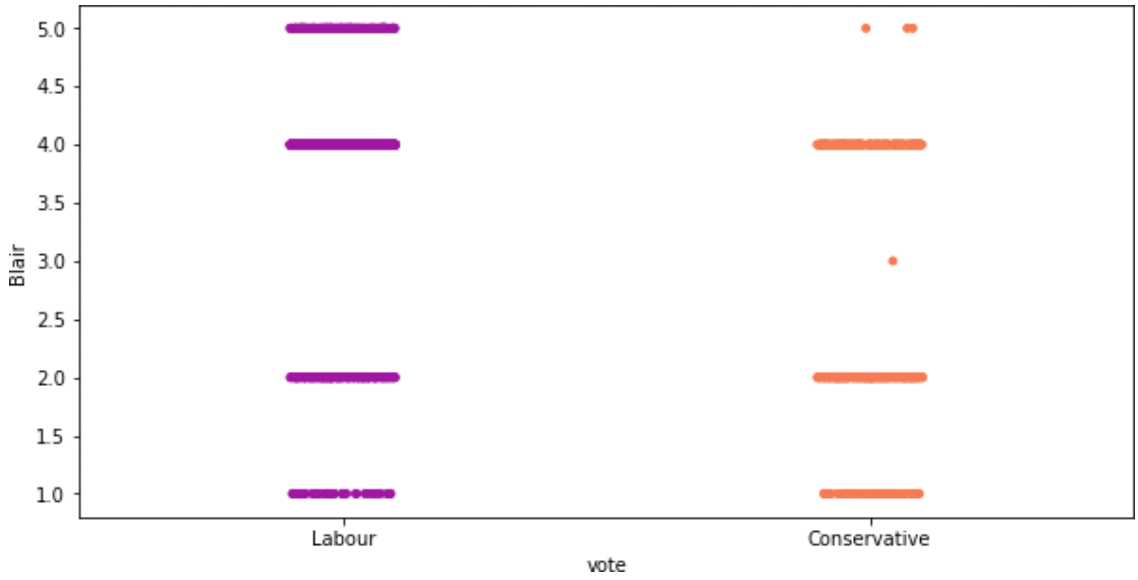
Strip plot of 'vote' and'economic.cond.household':

Viewing the exact values of the variables of 'vote'withrespect to 'economic.cond.household':

Observation:

- Labour party has higher votes overall.
- Out of 92 people who gave a score of 5, 69 people have voted for the labour party.
- Out of 435 people who gave a score of 4, 349 people have voted for the labour party. This is the 2nd highest set of people in the labour party.
- Out of 645 people who gave a score of 3, 448 people have voted for the labour party. This is the highest set of people in the labour party. The remaining 197 people who have voted for the conservative party is the highest set of people in that party.
- Out of 280 people who gave a score of 2, 154 people have voted for the labour party. 126 people have voted for the conservative party.

- Out of 65 people who gave a score of 1, 37 people have voted for the labour party. 28 people have voted for the conservative party.
- In all the instances, the labour party have more votes than the conservative party.



Strip plot of 'vote' and 'Blair':

Viewing the exact values of the variables of 'vote'withrespect to 'Blair':

```
vote          Blair
Conservative  2        240
              4        157
              1         59
              5          3
              3          1
Labour        4        676
              2        194
              5        149
              1         38
Name: Blair, dtype: int64
```

Observation:

- Labour party has higher votes overall.
- Out of 152 people who gave a score of 5, 149 people have voted for the labour party. The remaining 3 people, despite giving a score of 5 to the labour leader, have chosen to vote for the conservative party.

- Out of 833 people who gave a score of 4, 676 people have voted for the labour party. The remaining 157 people, despite giving a score of 4 to the labour leader, have chosen to vote for the conservative party.
- Only 1 person has given a score of 3 and that person hasvoted for the conservative party.
- Out of 434 people who gave a score of 2, 240 people have votedfor the conservative party. The remaining 194 people, despite giving an unsatisfactory score of 2 to the labour leader, have chosen to vote for the labour party.
- Out of 97 people who gave a score of 1, 59 people have votedfor the conservative party. The remaining 38 people, despite giving the lowest score of 1 to the labourleader, have chosento vote for the labour party.
- The score of 4 and 5 have more votes in the labour party.
- The score of 1, 2 and 3 have more votes in the conservativeparty.

Strip plot of 'vote' and 'Hague':

Viewing the exact values of the variables of 'vote'withrespect to 'Hague':

Observation:

- Labour party has higher votes overall.
- Out of 73 people who gave a score of 5, 59 people have voted forthe conservative party. The remaining 14people, despite giving a score of 5 to the conservative leader, have chosen to vote forthe labour party.
- Out of 557 people who gave a score of 4, 286 people have votedfor the conservative party. The remaining 271

people, despite giving a score of 4 to the conservative leader, have chosen to vote for the labour party.

- Out of 37 people who gave a score of 3, 28 have voted for the labour party. The remaining 9, despite giving an average score of 3 to the conservative party, have chosento vote for the conservative party.
- Out of 617 people who gave a score of 2, 522 people have voted for the labour party. The remaining 95 people, despite giving an unsatisfactory score of 2 to the conservative leader, have chosen to vote for the conservative party.
- Out of 233 people who gave a score of 1, 222 people havevoted for the labour party. The remaining 11 people,despite giving the lowest score of 1 to the conservative leader, have chosen to vote for the conservative party.
- The score of 4 and 5 have more votes in the conservative party, although in 4, the votes are almost equal in boththe parties. Conservative party gets slightly higher.
- The score of 1, 2 and 3 have more votes in the labourparty. Still, a significant percentage of people who gave a bad score to the conservative leader still chose to vote for 'Hague'.

Strip plot of 'vote' and 'Europe':

Viewing the exact values of the variables of 'vote'withrespect to 'Europe':

Observation:

- Out of 338 people who gave a score of 11, 166 peoplehave voted for the labour party and 172 people have voted for the conservative party.
- People who gave score of 7 to 10 have voted for labourand conservative almost equally. Conservative party seem to be slightly higher in these instances.
- Out of 207 people who gave a score of 6, 172 people have voted for the labour party and 35 people have voted for theconservative party.
- People who gave a score of 1 to 6 have predominantly voted for the labour party. As we can see, there are a totalof 770 people who have given scores from 1 to 6. Out of 770 people,
  672 people have voted for the labour party. So, 87.28% of the people have chosen labour party.
- So, we can infer that lower the 'Eurosceptic' sentiment, higher the votes for labour party.

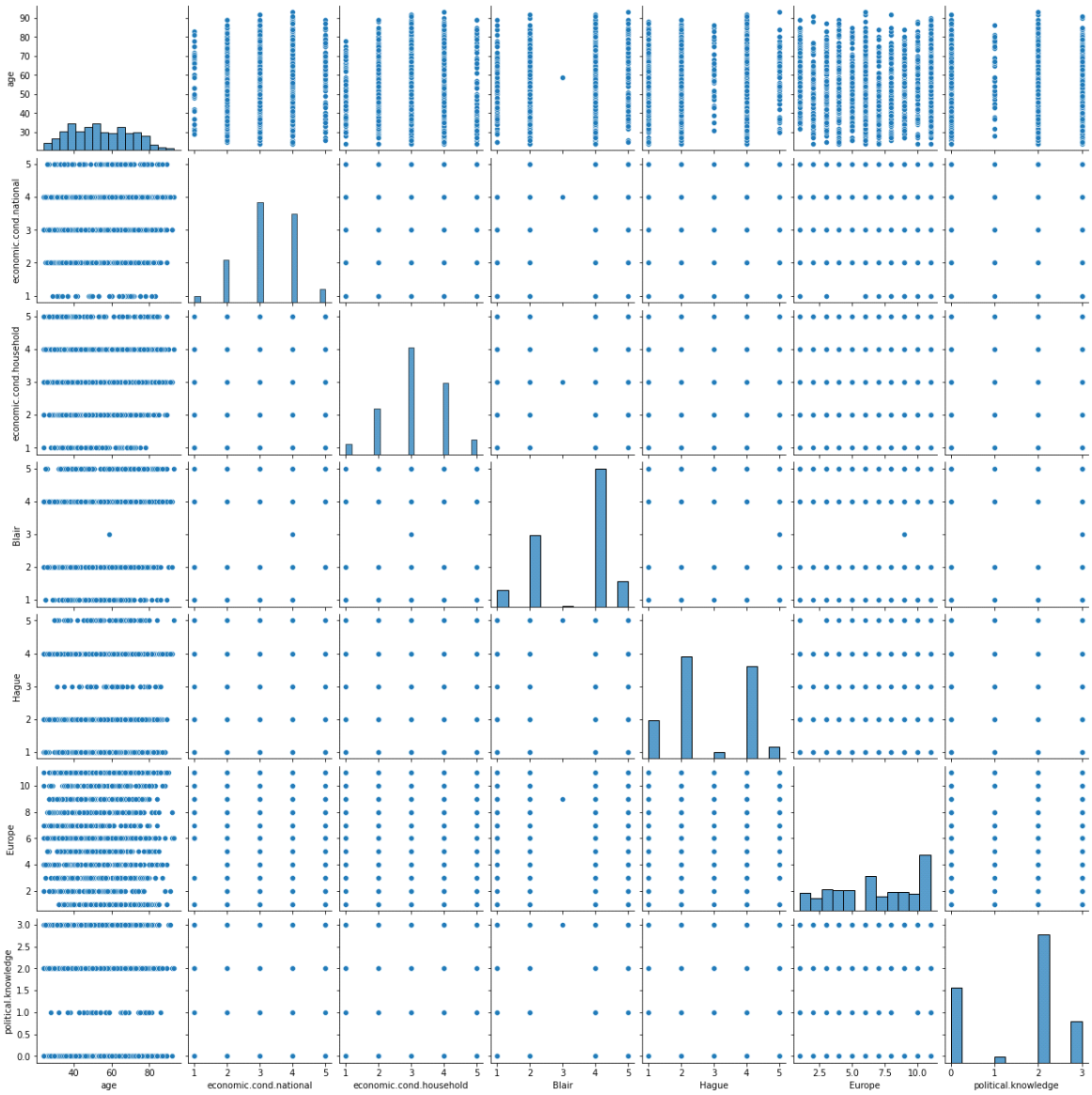
Strip plot of 'vote' and 'political.knowledge':

Viewing the exact values of the variables of 'vote'withrespect to 'political.knowledge':

Observation:

- Out of 249 people who gave a score of 3, 177 people have voted for the labour party and 72 people have voted for theconservative party.
- Out of 776 people who gave a score of 2, 493 people have voted for the labour party and 283 people have voted for the conservative party.
- Out of 38 people who gave a score of 1, 27 people have voted for the labour party and 11 people have voted for the conservative party.
- Out of 454 people who gave a score of 0, 360 people have voted for the labour party and 94 people have voted for theconservative party.
- We can see that, in all instances, labour party gets the higher number of votes.
- Out of 1517 people, 454 people gave a score of 0. So, this means that, 29.93% of the people are casting their votes without any political knowledge.

Checking pair-wise distribution of the continuousvariables:



Observation:

- Pair plot is a combination of histograms and scatter plots.
- From the histogram, we can see that, the 'Blair','Europe' and 'political.knowledge' variables are slightly left skewed.
- All other variables seem to be normally distributed.
- From the scatter plots, we can see that, there is mostly no correlation between the variables.

- We can use the correlation matrix to view them moreclearly.

Correlation matrix is a table which shows the correlationcoefficient between variables. Correlation values range from -1 to +1. For values closer to zero, it means that, there is no linear trend between two variables. Values close to 1 means that the correlationis positive.

The correlation heat map helps us to visualize the correlationbetweentwo variables.

Observation:

- We can see that, mostly there is no correlation in thedataset through this matrix. There are some variables that are moderately positively correlated and some thatare slightly negatively correlated.
- ‘economic.cond.national’ with ‘economic.cond.household’ have moderate positive correlation.
- ‘Blair' with 'economic.cond.national' and‘economic.cond.household’ have moderate positive correlation.
- ‘Europe’ with ‘Hague’ have moderate positive correlation.
- 'Hague' with 'economic.cond.national' and 'Blair' havemoderate negative correlation.
- 'Europe' with 'economic.cond.national' and 'Blair' have moderate negative correlation.

- **Encode the data (having string values) for Modelling. Is Scaling necessary here or not? Data Split: Split the data into train and test (70:30).**

Viewing the data after encoding:

Encoded data info:

Train-test-split:

Our model will use all the variables and 'vote_Labour' is the target variable. The train-test split is a technique for evaluating the performance of a machine learning algorithm.The procedure involves taking a dataset and dividing it into two subsets.

- **Train Dataset:** Used to fit the machine learning model.
- **Test Dataset:** Used to evaluate the fit machine learningmodel.

The data is divided into 2 subsets, training and testing set. Earlier, we have extracted the target variable 'vote_Labour' ina separate vector for subsets. Random state chosen as 1.

- **Training Set:** 70percent of data.
- **Testing Set:** 30 percent of the data.

```
x_train: (1061, 8)
y_train: (1061, 1)
x_test: (456, 8)
y_test: (456, 1)
```

Train-Test-Split Shape:

Why scaling?:

- The dataset contains features highly varying in magnitudes, units and range between the 'age' columnand other columns.
- But since, most of the machine learning algorithms use Eucledian distance between two data points in their computations, this is a problem.
- If left alone, these algorithms only take in the magnitude of features neglecting the units.
- The results would vary greatly between different units, 1km and 1000 metres.

- The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes.
- To supress this effect, we need to bring all features to the same level of magnitudes. This can be acheived by scaling.
- in this case, we have a lot of encoded, ordinal, categorical and continuous variables. So, we use the minmaxscaler technique to scale the data.

Viewing the data after scaling:

- **Apply Logistic Regression and LDA (lineardiscriminant analysis).**

Logistic Regression Model:

There are no outliers present in the continuous variable 'age'.The remaining variables are categorical in nature. Our modelwill use all the variables and 'vote_Labour' is the target variable.

Accuracy - Train data:

0.8341187558906692**Accuracy - Test data:**

Classification report - Train data:

Classification report - Test data:

Logistic Regression Model - ObservationTrain data:

- Accuracy: 83.41%
- Precision: 86%
- Recall: 92%
- F1-Score: 89%

Test data:

- Accuracy: 82.68%
- Precision: 86%
- Recall: 89%
- F1-Score: 87%

Validness of the model:

- The model is not over-fitted or under-fitted.
- The error in the test data is slightly higher than the train data, which is absolutely fine because the error margin is low and the error in both train and test data is not too high. Thus, the model is not over-fitted or under-fitted.

Linear Discriminant Analysis Model:

There are no outliers present in the continuous variable 'age'.The remaining variables are categorical in nature. Our modelwill use all the variables and 'vote_Labour' is the target variable.

Accuracy - Train data:

0.8341187558906692## Accuracy - Test data:

Classification report - Train data:

	precision	recall	f1-score	support
0	0.77	0.73	0.74	153
1	0.86	0.89	0.88	303
accuracy			0.83	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.83	0.83	456

Classification report - Test data:

Linear Discriminant Analysis Model - ObservationTrain data:

- Accuracy: 83.41%

- Precision: 86%
- Recall: 91%
- F1-Score: 89%

Test data:

- Accuracy: 83.33%
- Precision: 86%
- Recall: 89%
- F1-Score: 88%

Validness of the model:

- The model is not over-fitted or under-fitted.
- The error in the test data is slightly higher than the train data, which is absolutely fine because the error margin is low and the error in both train and test data is not too high. Thus, the modelis not over-fitted or under-fitted.

- **Apply KNN Model . Interpret the results.**

K-Nearest Neighbor Model:

There are no outliers present in the continuous variable 'age'.The remaining variables are categorical in nature. Our modelwill use all the variables and 'vote_Labour' is the target variable. We take K valueas 7.

1.0## Accuracy - Train data:

Accuracy - Test data:

Classification report - Train data:

	precision	recall	f1-score	support
0	0.78	0.71	0.75	153
1	0.86	0.90	0.88	303
accuracy			0.84	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.84	0.84	0.84	456

Classification report - Test data:

K-Nearest Neighbor Model - ObservationTraindata:

- Accuracy: 100%
- Precision: 100%
- Recall: 100%
- F1-Score: 100%

Test data:

- Accuracy: 83.77%
- Precision: 86%
- Recall: 90%
- F1-Score: 88%

Validness of the model:

- The model is over-fitted.
- As we can see, the train data has a 100% accuracy andtest data has 84% accuracy. The difference is more than 10%. So, we can infer that the KNN model is over-fitted.

- **Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting.Logistic Regression Model Tuning:**

Best parameters:

0.8360037700282752**Accuracy - Train data:**

Accuracy - Test data:

Classification report - Train data:

Classification report - Test data.

Logistic Regression Model Tuned - Observation

Train data:

- Accuracy: 83.6%
- Precision: 86%
- Recall: 92%
- F1-Score: 89%

Test data:

- Accuracy: 84.21%

- Precision: 86%
- Recall: 90%
- F1-Score: 88%

Comparison on performance of both regular andtunedlogistic regression models:

	Regular Model (%)	Tuned Model (%)
Train:		
Accuracy	83.41	83.6
Precision	86	86
Recall	92	92
F1-score	89	89
Test:		
Accuracy	82.68	84.21
Precision	86	86
Recall	89	90
F1-score	87	88

- As we can see from the above tabular comparison, there is not much difference between the performance regular LR model and tuned LR model.
- The values are high overall and there is no over-fitting or under-fitting. Therefore both models are equally good models.

```
{'solver': 'svd', 'tol': 0.0001}
```

Linear Discriminant Analysis Model Tuning: Best parameters:

Accuracy - Train data:

0.8322337417530632## Accuracy - Test data:

Classification report - Train data:

Classification report - Test data:

LDA Model Tuned - Observation

Train data:

- Accuracy: 83.22%
- Precision: 87%
- Recall: 90%
- F1-Score: 88%

Test data:

- Accuracy: 83.99%
- Precision: 87%
- Recall: 89%
- F1-Score: 88%

Comparison on performance of both regular andtunedLDA models:

	Regular Model(%)	Tuned Model (%)
Train:		
Accuracy	83.41	83.22
Precision	86	87
Recall	91	90
F1-score	89	88
Test:		
Accuracy	83.33	83.99
Precision	86	87

Recall	89	89
F1-score	88	88

- As we can see from the above tabular comparison, thereis not much difference between the performance of regular LDA model and tuned LDA model.
- The values are high overall and there is no over-fitting or under-fitting. Therefore both models are equally good

  models.

K-Nearest Neighbour Model Tuning:Best parameters:

0.8435438265786993**Accuracy - Train data:**

0.8618421052631579**Accuracy - Test data:**

Classification report - Train data:

Classification report - Test data:

	precision	recall	f1-score	support
0	0.84	0.73	0.78	153
1	0.87	0.93	0.90	303
accuracy			0.86	456
macro avg	0.85	0.83	0.84	456
weighted avg	0.86	0.86	0.86	456

KNN Model Tuned - Observation

Train data:

- Accuracy: 84.35%
- Precision: 88%

- Recall: 91%
- F1-Score: 89%

Test data:

- Accuracy: 86.18%
- Precision: 87%
- Recall: 93%
- F1-Score: 90%

Comparison on performance of both regular andtunedKNN models:

	Regular Model(%)	Tuned Model (%)
Train:		
Accuracy	100	84.35
Precision	100	88
Recall	100	91
F1-score	100	89
Test:		
Accuracy	83.77	86.18
Precision	86	87
Recall	90	93
F1-score	88	90

- There is no over-fitting or under-fitting in the tuned KNNmodel. Overall, it is a good model.
- As we can see, the regular KNN model was over-fitted. But model tuning has helped the model to recover fromover- fitting.
- The values are better in the tuned KNN model.

- Therefore, the tuned KNN model is a better model.

```
        Imp
0  0.215348
5  0.185293
4  0.181437
3  0.136335
1  0.094462
2  0.077034
6  0.075300
7  0.034792
```

Ensemble Random Forest Classifier

Feature importances:

- Here,
- 0 = age
- 1 = economic.cond.national
- 2 = economic.cond.household
- 3 = Blair
- 4 = Hague
- 5 = Europe
- 6 = political.knowledge
- 7 = gender_male

Accuracy - Train data:

```
0.8267543859649122
```

Accuracy - Test data:

Classification report - Train data:

Classification report - Test data:

Random Forest Classifier -

ObservationTrain data:

- Accuracy: 100%
- Precision: 100%
- Recall: 100%
- F1-Score: 100%

Test data:

- Accuracy: 82.68%
- Precision: 84%
- Recall: 91%
- F1-Score: 88%

The model is over-fitted. We will use bagging to improve theperformance of the model.

Ensemble technique -

BaggingAccuracy -

Train data:

Accuracy - Test data:

The RF model even after using bagging technique, is still over-fitted.

0.8426013195098964**Ensemble technique -**

AdaBoostingAccuracy -

Train data:

Accuracy - Test data:

The model is not over-fitted. The values are good. Therefore,themodel is a good model.

Ensemble technique - Gradient BoostingAccuracy - Train data:

Accuracy - Test data:

The model is not over-fitted. The values are better than AdaBoosting model. The model is a good model.

```
{'criterion': 'gini',
 'max_depth': 8,
 'max_features': 5,
 'min_samples_leaf': 9,
 'min_samples_split': 50,
 'n_estimators': 100,
 'random_state': 1}
```

Random Forest model TuningBest parameters:

Bagging tune d:Accuracy - Train data:

0.8135964912280702Accuracy - Test data:

Classification report - Train data:

Classification report - Test data:

The tuning of the model has help the model recover from over-fitting.Now the model is a good model.

0.9349670122525919**Random Forest**

tuned - AdaBoosting

Accuracy - Train data:

Accuracy - Test data:

Classification Report - Train data:

	precision	recall	f1-score	support
0	0.90	0.87	0.89	307
1	0.95	0.96	0.95	754
accuracy			0.93	1061
macro avg	0.92	0.92	0.92	1061
weighted avg	0.93	0.93	0.93	1061

Classification Report - Test data:

There is no over-fitting. There is improvement from the regularmodel.The model is a good model.

Gradient Boosting Classifier

TunedBest parameters:

Accuracy - Train data:

Accuracy - Test data:

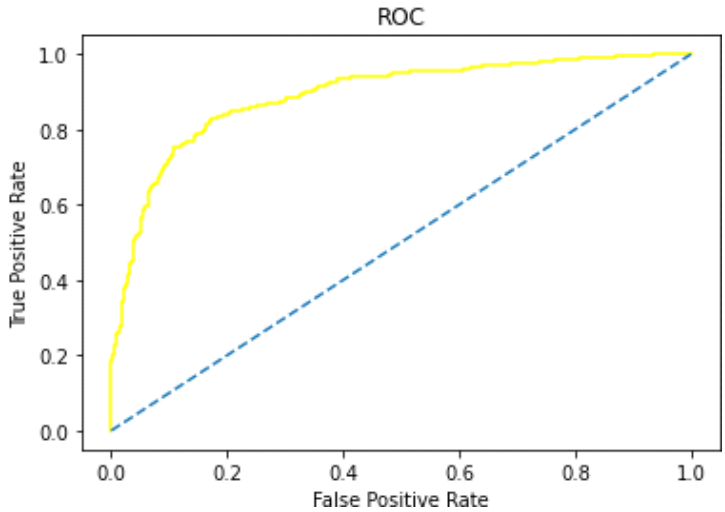
Classification Report - Train data:

	precision	recall	f1-score	support
0	0.86	0.75	0.80	153
1	0.88	0.94	0.91	303
accuracy			0.87	456
macro avg	0.87	0.84	0.85	456
weighted avg	0.87	0.87	0.87	456

Classification Report - Test data:

- The gradient boost classifier after tuning, has improvedthe model significantly.
- The difference between the train and test accuracies hasalsobeen reduced.
- Overall, the tuned Gradient Boost classifier is a bettermodel.

AUC: 0.8898383431686813



- 0.8341187558906692

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

Logistic Regression Model - Regular:

Predicted Class and probs:

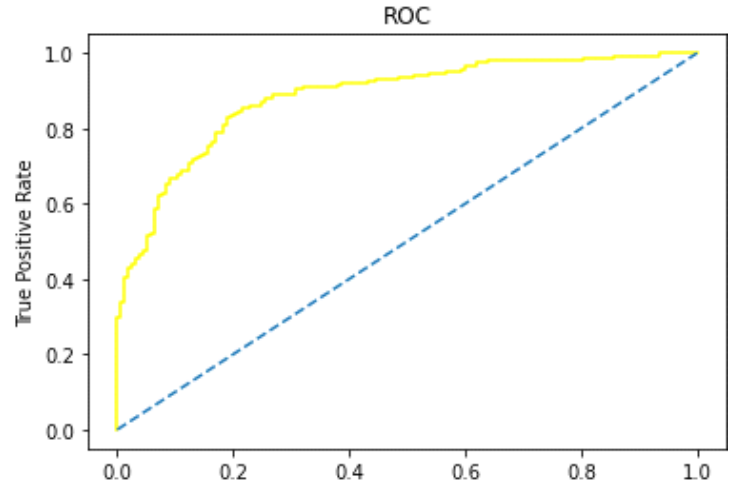
Accuracy - Train:

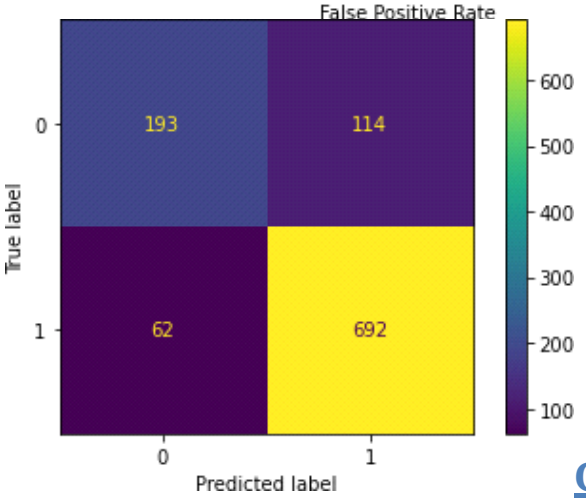
ROC and AUC - Train:

0.8267543859649122## Accuracy - Test:

ROC and AUC - Test:

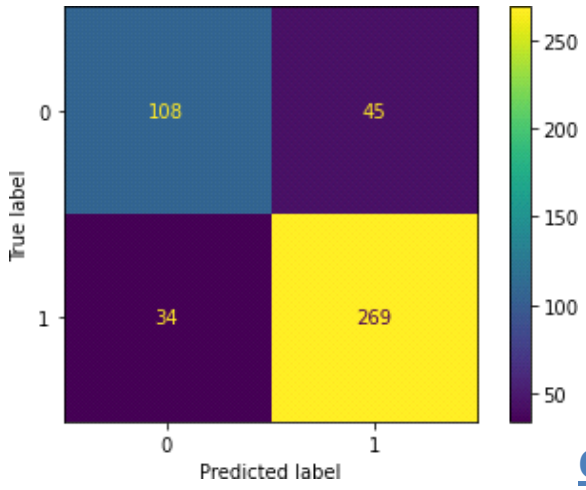
AUC: 0.8840786039388253





Confusion matrix - Train:

Classification report - Train:



Confusion matrix - Test:

Classification report - Test:

Observation:Train data:

- Accuracy: 83.41%
- Precision: 86%
- Recall: 92%
- F1-Score: 89%

- AUC: 88.98%

Test data:

- Accuracy: 82.68%
- Precision: 86%
- Recall: 89%
- F1-Score: 87%
- AUC: 88.4%

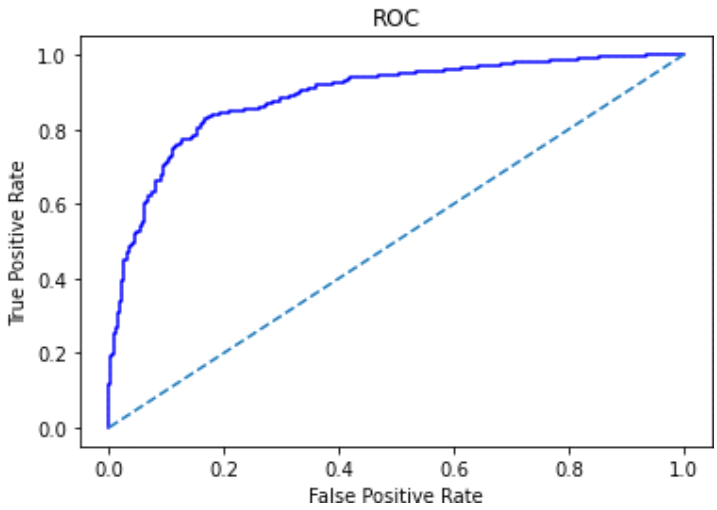
The model is not over-fitted or under-fitted. It is a good model.

Logistic Regression Model - Tuned:Predicted Class and probs:

Accuracy - Train:

ROC and AUC - Train:

```
AUC: 0.8888533683546601
```

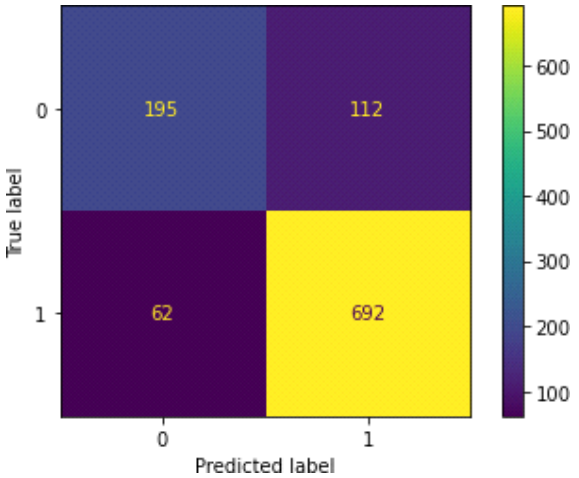



```
0.8421052631578947
```

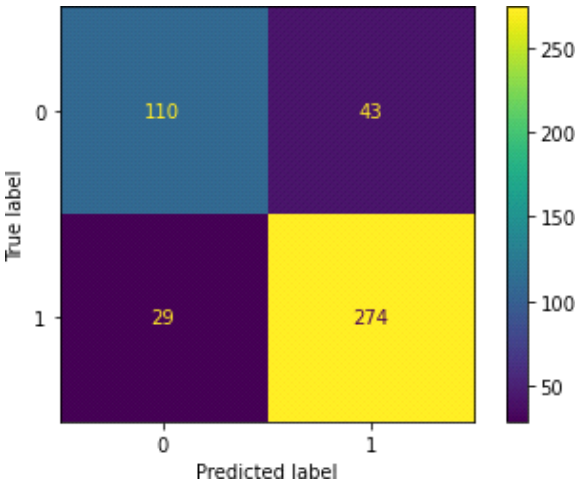
Accuracy - Test:

ROC and AUC - Test:

Confusion matrix - Train:



Classification report - Train:



Confusion matrix - Test:

Classification report - Test:

Observation:Train data:

- Accuracy: 83.6%
- Precision: 86%
- Recall: 92%
- F1-Score: 89%
- AUC: 88.89%

Test data:

- Accuracy: 84.21%
- Precision: 86%
- Recall: 90%
- F1-Score: 88%
- AUC: 89.05%

Comparison between the regular LR model andtuned LR model:

- As we can see, there is not much difference between the performance of regular LR model and tuned LR model.

- The values are high overall and there is no over-fitting orunder-fitting. Therefore both models are equally good models.

LDA Model - Regular:Predicted Class and probs:

Accuracy - Train:

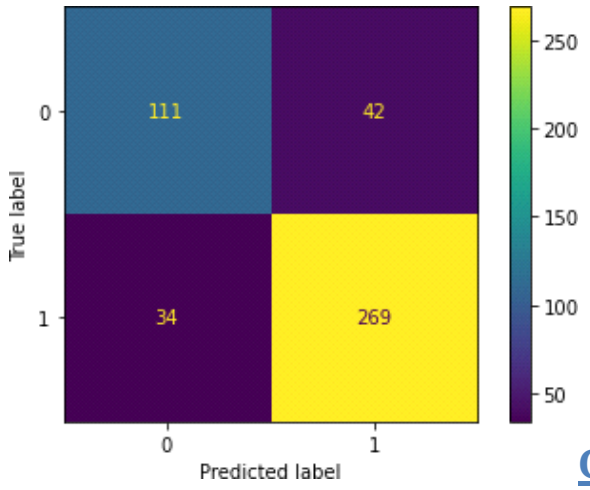
Accuracy - Test:

ROC and AUC - Train:

ROC and AUC - Test:

Confusion matrix - Train:

Classification report - Train:



Confusion matrix - Test:

Classification report - Test:

	precision	recall	f1-score	support
0	0.77	0.73	0.74	153
1	0.86	0.89	0.88	303
accuracy			0.83	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.83	0.83	456

Observation:

Train data:

- Accuracy: 83.41%
- Precision: 86%
- Recall: 91%
- F1-Score: 89%
- AUC: 88.94%

Test data:

- Accuracy: 83.33%
- Precision: 86%
- Recall: 89%
- F1-Score: 88%
- AUC: 88.76%

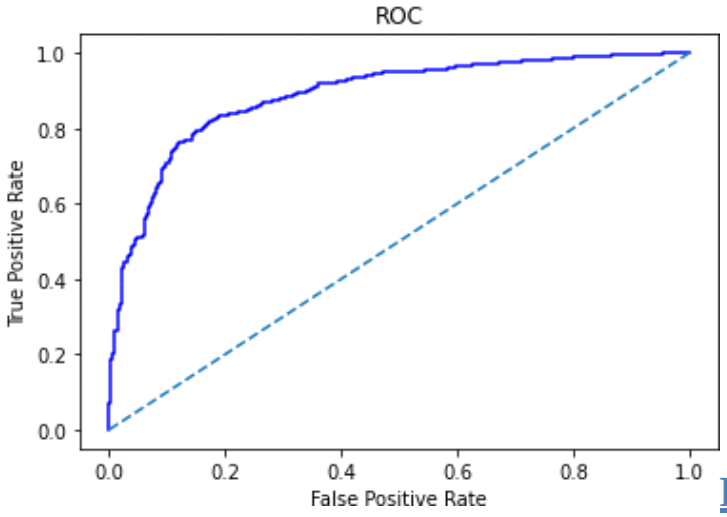
Validness of the model:

- The model is not over-fitted or under-fitted.
- The error in the test data is slightly higher than the train data, which is absolutely fine because the error margin is low and the error in both train and test data is not too high. Thus, the modelis not over-fitted or under-fitted

LDA Model - Tuned:Predicted Class and probs:

Accuracy - Train:

```
AUC: 0.8868186177520109
```

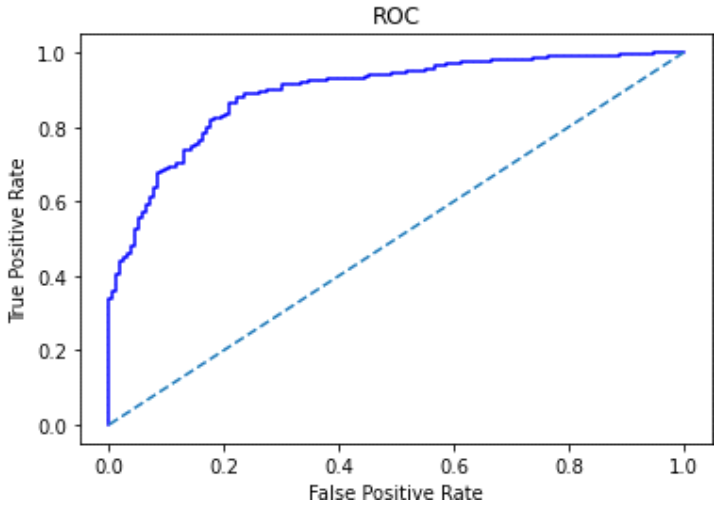


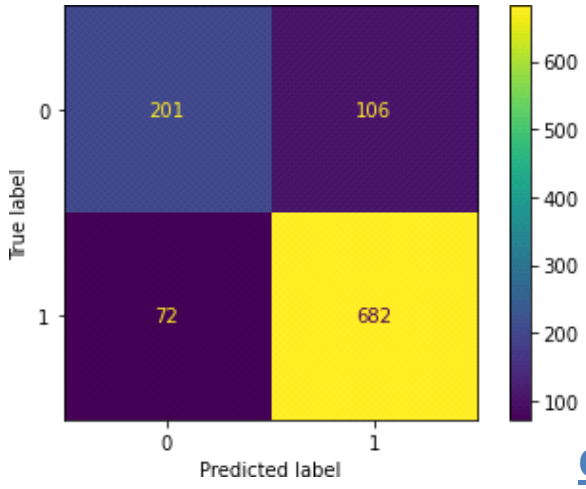
ROC and AUC - Train:

Accuracy - Test:

ROC and AUC - Test:

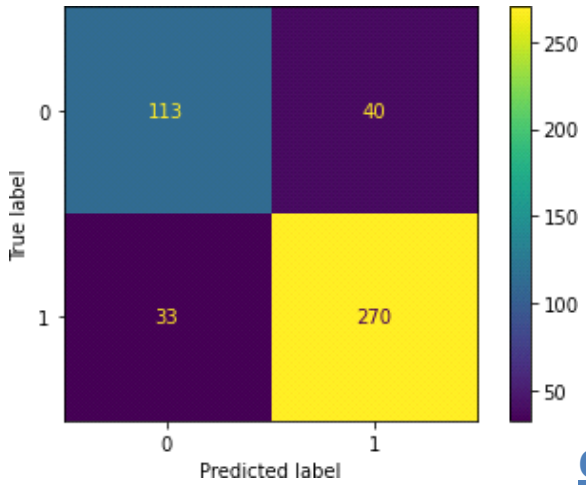
```
AUC: 0.8933324705019522
```





Confusion matrix - Train:

Classification report - Train:



Confusion matrix - Test:

Classification report - Test:

Observation:

Train data:

- Accuracy: 83.22%
- Precision: 87%
- Recall: 90%
- F1-Score: 88%
- AUC: 88.68%

Test data:

- Accuracy: 83.99%
- Precision: 87%
- Recall: 89%
- F1-Score: 88%
- AUC: 89.33%

There is no over-fitting or under-fitting in the tuned LDAmodel.Overall, it is a good model.

Comparison between the regular LDA model andtuned LDA model

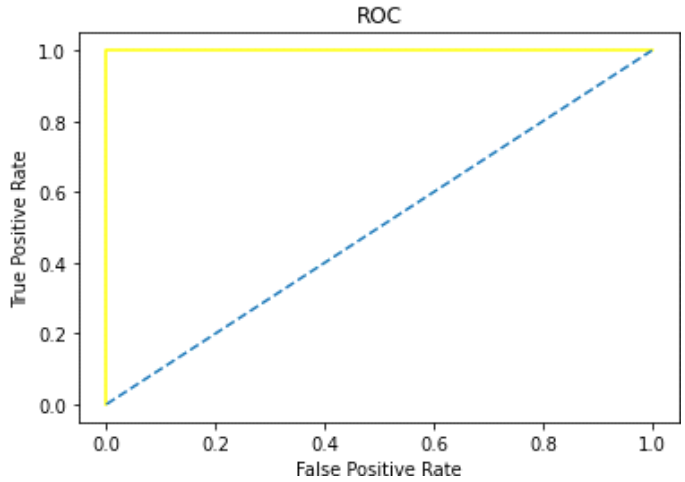
- As we can see, there is not much difference between the performance of regular LDA model and tuned LDA model.
- The values are high overall and there is no over-fitting orunder-fitting.
- Therefore both models are equally good models.

KNN Model - Regular:Predicted Class and probs:

Accuracy - Train:

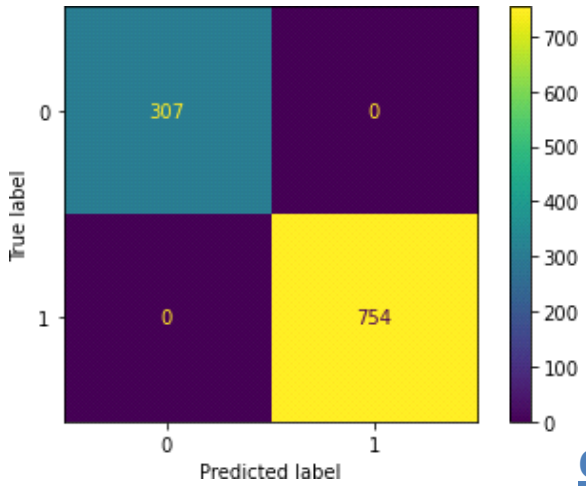
ROC and AUC - Train:

```
AUC: 1.0
```



Accuracy - Test:

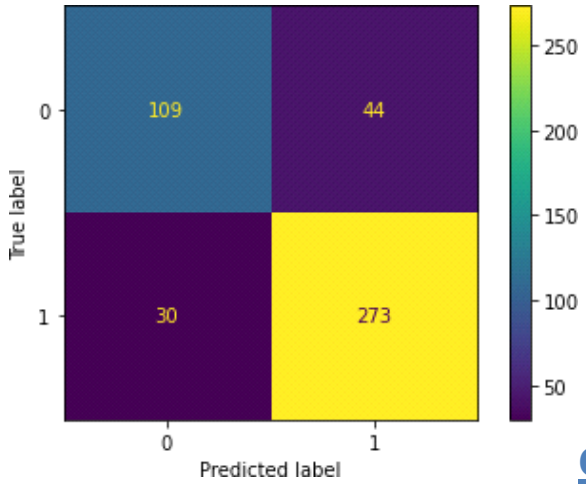
ROC and AUC - Test:



Confusion matrix - Train:

Classification report - Train:

Confusion matrix - Test:



Classification report - Test:

Observation:Train data:

- Accuracy: 100%
- Precision: 100%
- Recall: 100%
- F1-Score: 100%
- AUC: 100%

Test data:

- Accuracy: 83.77%
- Precision: 86%
- Recall: 90%
- F1-Score: 88%
- AUC: 88.46%

Validness of the model

- The model is over-fitted.
- As we can see, the train data has a 100% accuracy andtest data has 84% accuracy. The difference is more than 10%. So, we can infer that the KNN model is over-fitted.

KNN Model - Tuned:Predicted

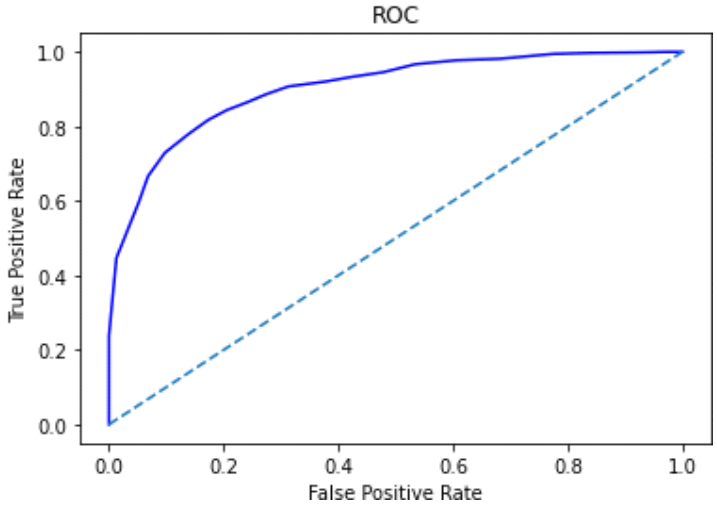
Class and probs:

Accuracy - Train:

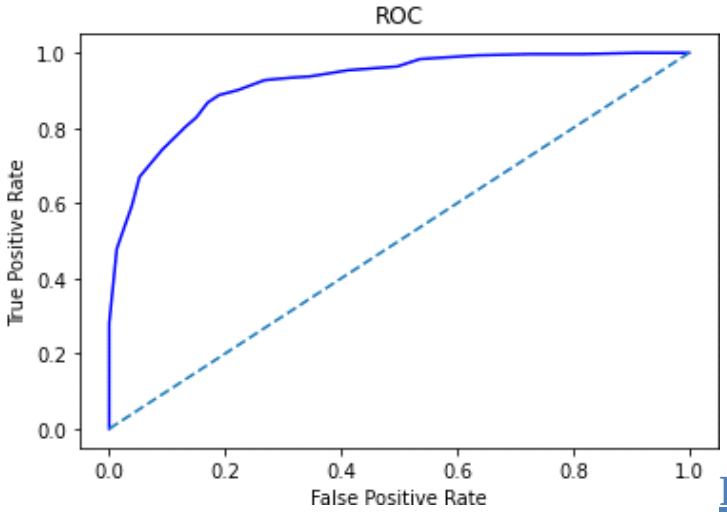
Accuracy - Test:

ROC and AUC - Train:

```
AUC: 0.9022498898383432
```

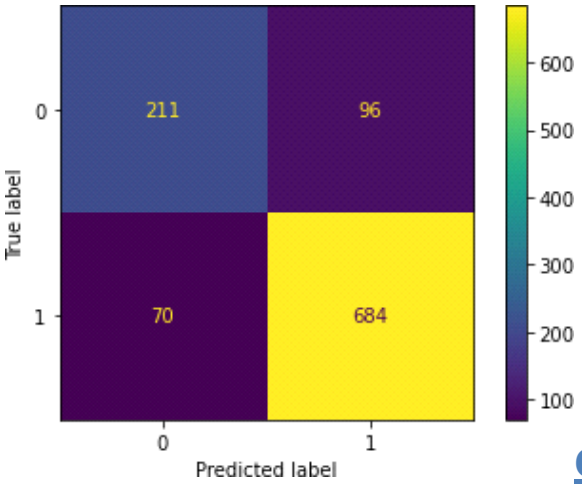


AUC: 0.922722664423305



ROC and AUC - Test:

Confusion matrix - Train:



Classification report - Train:

Confusion matrix - Test:

Classification report - Test:

Observation:

Train data:

- Accuracy: 84.35%
- Precision: 88%

- Recall: 91%
- F1-Score: 89%
- AUC: 90.23%

Test data:

- Accuracy: 86.18%
- Precision: 87%
- Recall: 93%
- F1-Score: 90%
- AUC: 92.27%

There is no over-fitting or under-fitting in the tuned KNNmodel.

0.8223684210526315Overall, it is a good model.

- Comparison between the regular KNN model and tunedKNN model:
- As we can see, the regular KNN model was over-fitted.But model tuning has helped the model to recover from over-fitting.
- The values are better in the tuned KNN model.
- Therefore, the tuned KNN model is a better model.

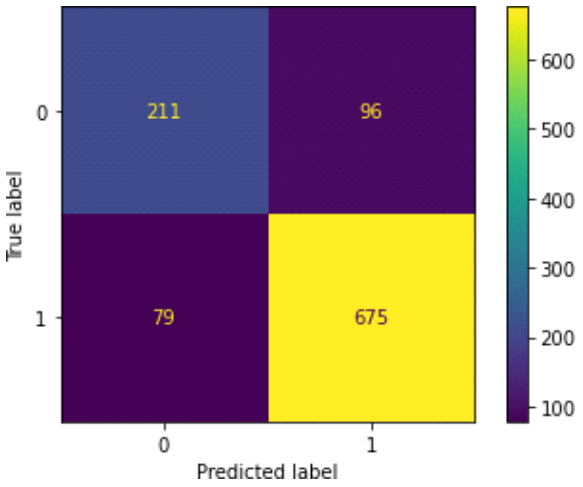
Naïve Bayes Model - Regular:Predicted

Class and probs:

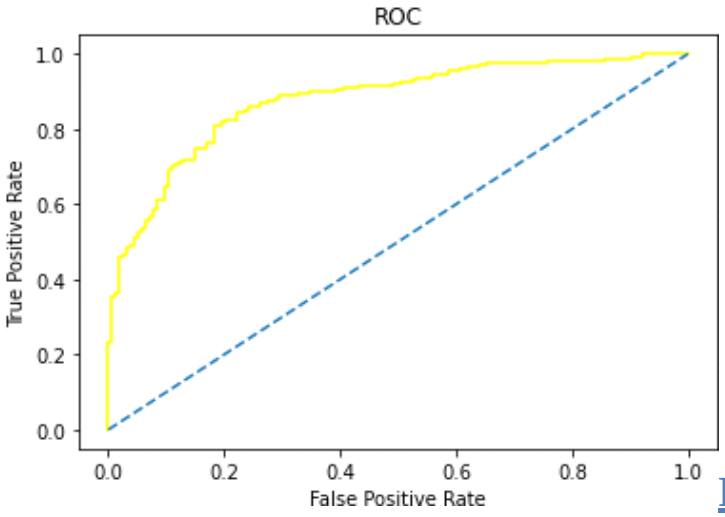
0.8350612629594723**Accuracy - Train:**

ROC and AUC - Train:

Accuracy - Test:



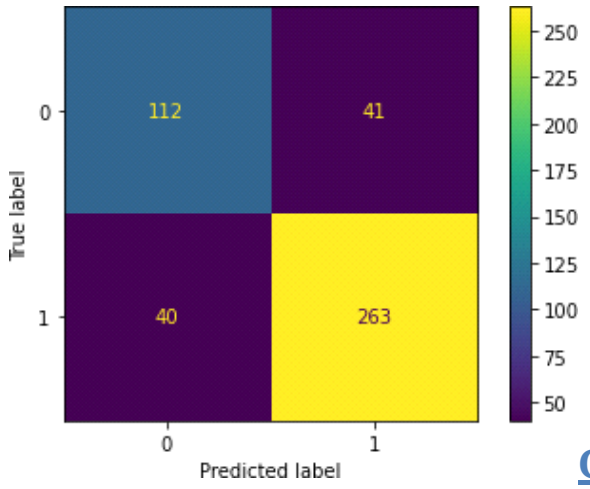
AUC: 0.8763562630772882



ROC and AUC - Test:

Confusion matrix - Train:

Classification report - Train:



Confusion matrix - Test:

Classification report - Test:

Observation:

Train data:

- Accuracy: 83.51%
- Precision: 88%
- Recall: 90%
- F1-Score: 89%AUC: 88.79%

Test data:

- Accuracy: 82.24%
- Precision: 87%
- Recall: 87%
- F1-Score: 87%
- AUC: 87.64%

Validness of the model

- The model is not over-fitted or under-fitted.
- The error in the test data is slightly higher than the train data, which is absolutely fine because the error margin is low and the

error in both train and test data is not too high. Thus, the modelis not over-fitted or under-fitted.

- There is no hyper-parameters to tune in Naive Bayes model. So,we cannot tune this model.

Ensemble Random Forest Classifier - Regular:Predicted Class and probs:

Accuracy - Train:

ROC and AUC - Train:

Accuracy - Test:

ROC and AUC - Test:

Confusion matrix - Train:

Confusion matrix - Test:

Classification report - Test:

Observation:

Train data:

- Accuracy: 100%
- Precision: 100%
- Recall: 100%
- F1-Score: 100%
- AUC: 100%

Test data:

- Accuracy: 82.68%
- Precision: 84%
- Recall: 91%
- F1-Score: 88%
- AUC: 88.62%

The model is over-fitted. So we will use bagging to improve the performance of the model.

Bagging -

Regular:

Predicted Class and probs:

Accuracy - Train:

ROC and AUC - Train:

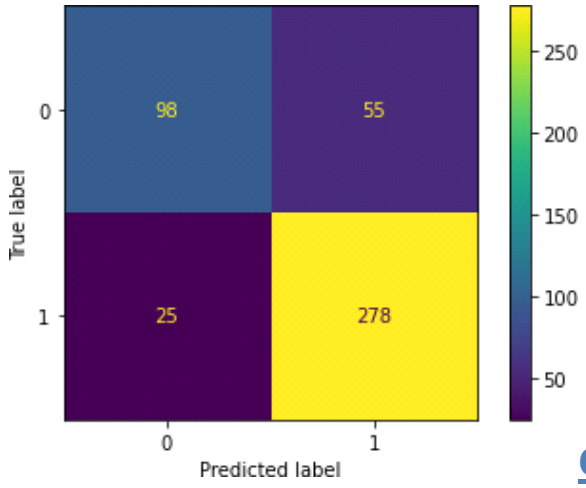
0.8245614035087719## Accuracy - Test:

ROC and AUC –

Test:

Confusion matrix - Train:

Classification report - Train:



Confusion matrix - Test:

Classification report - Test:

Observation:

Train data:

- Accuracy: 95.38%
- Precision: 95%
- Recall: 98%
- F1-Score: 97%
- AUC: 99.4%

Test data:

- Accuracy: 82.46%
- Precision: 83%
- Recall: 92%
- F1-Score: 87%
- AUC: 89.4%

After using bagging, the model is still over-fitted. The values are high.But the difference between the train and test accuracy is high.

Bagging - Tuned: Predicted

Class and probs:

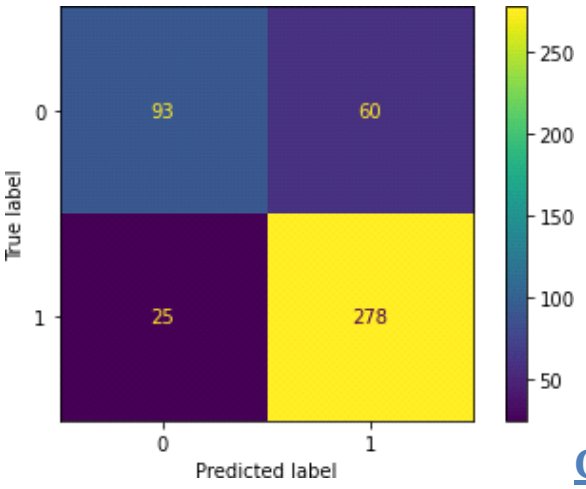
Accuracy - Train:

Accuracy - Test:

ROC and AUC - Test:

Confusion matrix - Train:

Classification report - Train:



Confusion matrix - Test:

Classification report - Test:

Observation:

Train data:

- Accuracy: 84.45%
- Precision: 86%
- Recall: 94%
- F1-Score: 90%
- AUC: 90.41%

Test data:

- Accuracy: 81.36%
- Precision: 82%
- Recall: 92%
- F1-Score: 87%
- AUC: 88.58%

The tuning of the model has help the model recover from over-fitting.Now the model is a good model.

AdaBoosting -

Regular:

Predicted Class and probs:

Accuracy - Train:

ROC and AUC - Train:

Accuracy - Test:

ROC and AUC – Test

Confusion matrix - Train:

Classification report - Train:

Confusion matrix - Test:

Classification report - Test:

Observation:

Train data:

- Accuracy: 84.26%
- Precision: 87%
- Recall: 91%
- F1-Score: 89%
- AUC: 89.79%

Test data:

- Accuracy: 82.02%
- Precision: 86%
- Recall: 87%
- F1-Score: 87%
- AUC: 87.81%

The tuning of the model has help the model recover from over-fitting.Now the model is a good model.

AdaBoosting - Tuned:

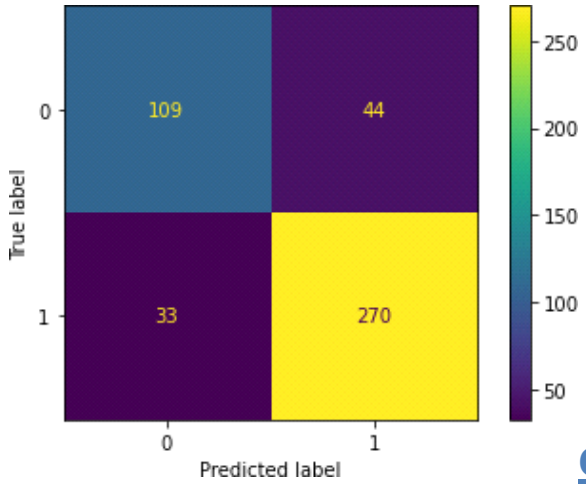
Predicted Class and probs:

Accuracy - Train:

ROC and AUC - Train:

Accuracy - Test:

ROC and AUC - Test:



Confusion matrix - Train:

Classification report - Train:

Confusion matrix - Test:

Classification report - Test:

Observation:

Train data:

- Accuracy: 93.5%
- Precision: 95%
- Recall: 96%
- F1-Score: 95%
- AUC: 98.62%

Test data:

- Accuracy: 83.11%
- Precision: 86%
- Recall: 89%
- F1-Score: 88%
- AUC: 89.99%

The model is a good model. There is no over-fitting. There isimproveme

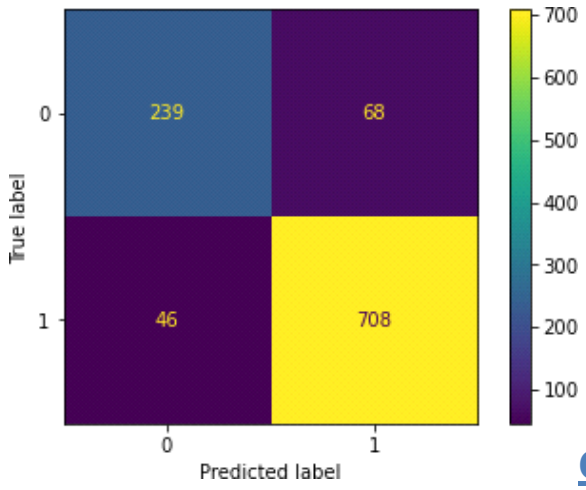
Gradient Boosting - Regular:Predicted Class and probs:

Accuracy - Train:

ROC and AUC - Train:

0.8333333333333334**Accuracy - Test:**

ROC and AUC - Test:



Confusion matrix - Train:

Classification report - Train:

Confusion matrix - Test:

Classification report - Test:

Observation:

Train data:

- Accuracy: 89.26%
- Precision: 91%
- Recall: 94%
- F1-Score: 93%
- AUC: 95.11%

Test data:

- Accuracy: 83.33%
- Precision: 85%
- Recall: 91%
- F1-Score: 88%
- AUC: 89.87%

The values are high. There is no over-fitting of any sorts. Themodelis a good model.

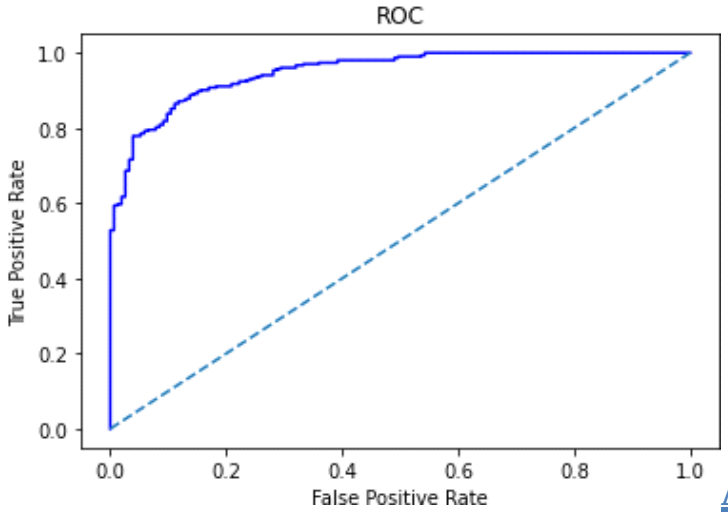
Gradient Boosting - Tuned:Predicted

Class and probs:

0.883129123468426## Accuracy - Train:

ROC and AUC - Train:

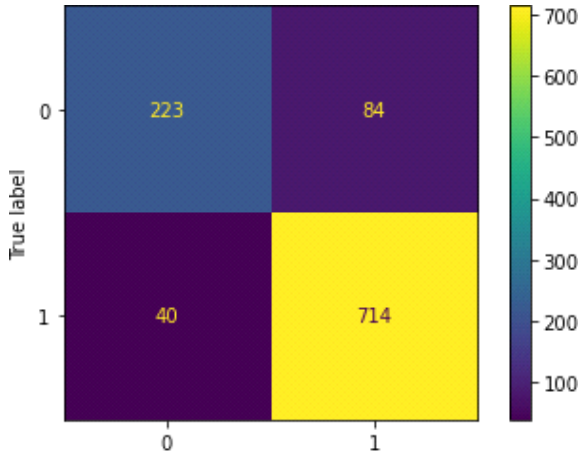
```
AUC: 0.9496861450850966
```




```
0.8728070175438597
```

Accuracy - Test:

ROC and AUC - Test:




```
              precision    recall  f1-score   support

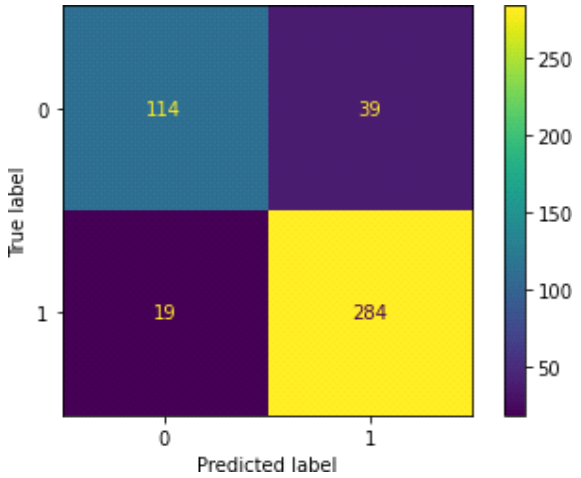
           0       0.85      0.73      0.78       307
           1       0.89      0.95      0.92       754

    accuracy                           0.88      1061
   macro avg       0.87      0.84      0.85      1061
weighted avg       0.88      0.88      0.88      1061
```

Confusion matrix - Train:

Classification report - Train:

Confusion matrix - Test:



	precision	recall	f1-score	support
0	0.86	0.75	0.80	153
1	0.88	0.94	0.91	303
accuracy			0.87	456
macro avg	0.87	0.84	0.85	456
weighted avg	0.87	0.87	0.87	456

Classification report - Test:

Observation:

Train data:

- Accuracy: 88.31%
- Precision: 89%
- Recall: 95%
- F1-Score: 92%
- AUC: 94.69%

Test data:

- Accuracy: 87.28%
- Precision: 88%
- Recall: 94%
- F1-Score: 91%
- AUC: 94.97%

The tuning of the Gradient Boost model has improved the model further. The values are high. The better is better than the regular model.

Comparison of train data of all models in a structured tabular manner:

	Accuracy	Precision	Recall	F1-Score	AUC
LR - Regular	83.41%	86%	92%	89%	88.98 %
LR - Tuned	83.6%	86%	92%	89%	88.89 %
LDA - Regular	83.41%	86%	91%	89%	88.94 %
LDA - Tuned	83.22%	87%	90%	88%	88.68 %
KNN - Regular	100%	100%	100%	100%	100%
KNN - Tuned	84.35%	88%	91%	89%	90.23 %
Naïve Bayes - Regular	83.51%	88%	90%	89%	88.79 %
Random Forest - Regular	100%	100%	100%	100%	100%
Bagging - Regular	95.38%	95%	98%	97%	99.4%
Bagging - Tuned	84.45%	86%	94%	90%	90.41 %
AdaBoosting - Regular	84.26%	87%	91%	89%	89.79 %
AdaBoosting - Tuned	93.5%	95%	96%	95%	98.62 %
Gradient Boosting - Regular	89.26%	91%	94%	93%	95.11 %
Gradient Boosting - Tuned	88.31%	89%	95%	92%	94.69 %

Comparison of test data of all models in astructured

	Accuracy	Precision	Recall	F1-Score	AUC
LR - Regular	82.68%	86%	89%	87%	88.4%
LR - Tuned	84.21%	86%	90%	88%	89.05 %

LDA - Regular	83.33%	86%	89%	88%	88.76 %
LDA - Tuned	83.99%	87%	89%	88%	89.33 %
KNN - Regular	83.77%	86%	90%	88%	88.46 %
KNN - Tuned	86.18%	87%	93%	90%	92.27 %
Naïve Bayes - Regular	82.24%	87%	87%	87%	87.64 %
Random Forest - Regular	82.68%	84%	91%	88%	88.62 %
Bagging - Regular	82.46%	83%	92%	87%	89.4%
Bagging - Tuned	81.36%	82%	92%	87%	88.58 %
AdaBoosting - Regular	82.02%	86%	87%	87%	87.81 %
AdaBoosting - Tuned	83.11%	86%	89%	88%	89.99 %
Gradient Boosting - Regular	83.33%	85%	91%	88%	89.87 %
Gradient Boosting - Tuned	87.28%	88%	94%	91%	94.97 %

Comparing the AUC, ROC curve on the train data of all the tuned models:

In all the models, tuned ones are better than the regular models. So, we compare only the tuned models and describe which model is the best/optimized.

Comparing the AUC, ROC curve on the test data of all the tuned models:

In all the models, tuned ones are better than the regular models. So, we compare only the tuned models and describe which model is the best/optimized.

Conclusion:

- There is no under-fitting or over-fitting in any of thetunedmodels.
- All the tuned models have high values and every model is good. But as we can see, the most consistent tuned model in both train and test data is the Gradient Boost model.
- The tuned gradient boost model performs the best with 88.31% accuracy score in train and 87.28% accuracy score in test. Also it has the best AUC score of 94% inboth train and test data which is the highest of all the models.
- It also has a precision score of 88% and recall of 94% which is also the highest of all the models. So, we conclude that Gradient Boost Tuned model is the best/optimized model.

- **Based on these predictions, what are theinsights?**

Insights:

- Labour party has more than double the votes of conservative party.
- Blair has higher number of votes than Hague and the scores are much better for Blair than for Hague.
- On a scale of 0 to 3, about 30% of the total population has zero knowledge about politics/parties.
- People who gave a low score of 1 to a certain party, still decided to vote for the same party instead of voting for the other party. This can be because of lack of political knowledgeamong the people.
- People who have higher Eurosceptic sentiment, has voted for the conservative party and lower the Eurosceptic sentiment, higher the votes for Labour party.
- Out of 454 people who gave a score of 0 for political knowledge, 360 people have voted for the labour partyand 94 people have voted for the conservative party.

- All models performed well on training data set as well as test dat set. The tuned models have performed better than the regular models.
- There is no over-fitting in any model except Random Forest and Bagging regular models.
- Gradient Boosting model tuned is the best/optimized model.

Business recommendations:

- Hyper-parameters tuning is an import aspect of modelbuilding. There are limitations to this as to process these combinations, huge amount of processing power is required. But if tuning can be done with many sets ofparameters, we might get even better results.
- Gathering more data will also help in training the models and thus improving the predictive powers.
- We can also create a function in which all the modelspredict the outcome in sequence. This will helps in better understanding and the probability of what the outcome will be.
- Using Gradient Boosting model without scaling for predicting the outcome as it has the best optimized performance.