**Election Results Prediction**

**Report**

Submission By: **Raj Keswani**

Contents

**Problem** **Statement**

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 must build a model, to predict which party a voter will vote for based on the given information, to create an exit poll that will help in predicting overall win and seats covered by a particular party.

**Data Ingestion**   
1.1 Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it.   
1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.

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

**Modeling**   
1.4 Apply Logistic Regression and LDA (linear discriminant analysis).   
1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results.   
1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging) and boosting.   
1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized.

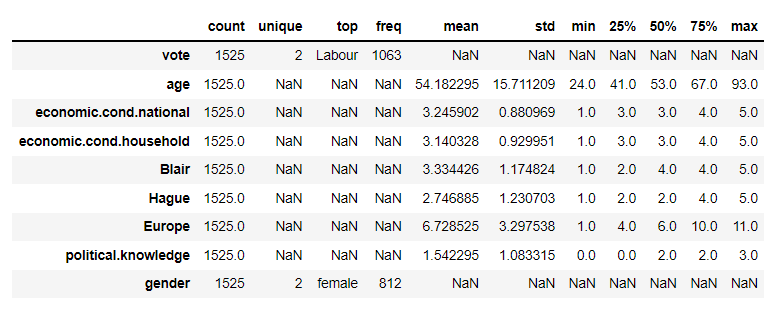
**Inference**1.8 Based on these predictions, what are the insights?

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

**Reading the dataset.**

The data was read into a data frame using Pandas library. It was observed that the data had an unnamed column what appeared to be the serial number. Since it was of no significance to us it was dropped from the data frame.

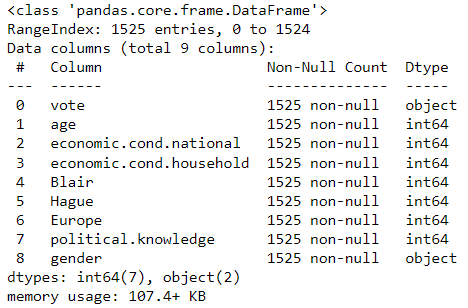
**Descriptive statistics.**



**Table I** Data Description

* It was observed that the minimum and maximum values of all the columns with numerical data type and the number of unique values for columns with categorical data type were as expected, removing any suspicion of corrupt data entries in the dataset.
* Mean and median for all the columns were observed to be close enough for the assumption of normal distribution.
* Two insights can be easily drawn by looking at the statistical description that are as follows:
  + Labor party won the elections as it got maximum votes.
  + Most number of the voters were female.

**Null value check.**



**Table II** Data Information

* All the columns have 1525 non-null values implying there are no null values in the dataset.
* The shape of the dataset is observed to be 1525 rows X 9 columns.
* 2 of the columns are of object datatype while other 7 are of int64 datatype.

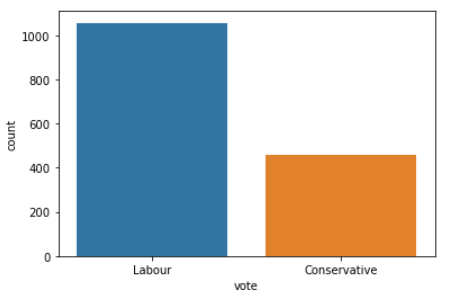
**NOTE**

There were 8 duplicate row entries found in the dataset. Even though it is distinctly possible for data duplicity to happen here i.e., different people having same attributes and voting preferences but here, for the sake of not introducing any bias into the machine learning models, a call was made to remove those entries as they were contributing similar information making new shape of data to be 1517 X 9.

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

**Univariate Data Analysis.**

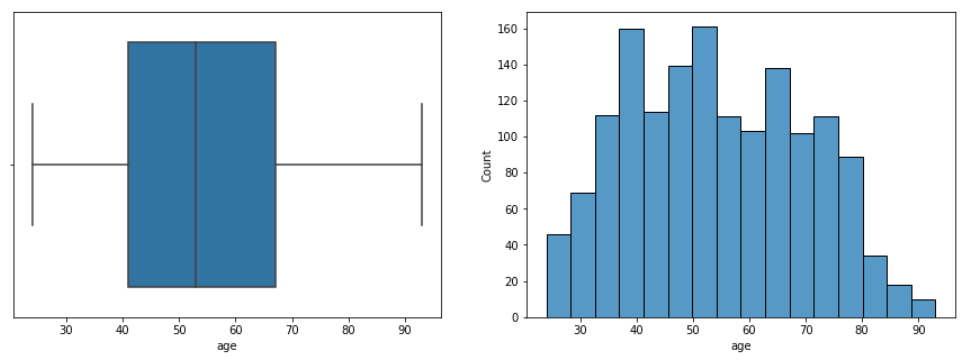
**Vote**



**Figure 1** Vote Distribution

* It is clearly visible that the Labour party won the elections with almost twice as many numbers of votes.

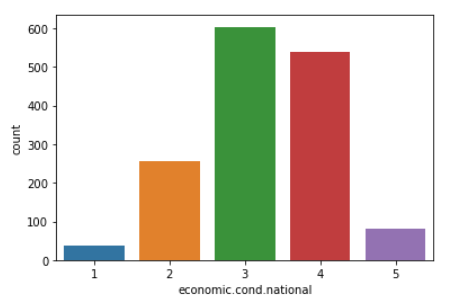
**Age**



**Figure 2** Age Distribution of Voters.

* The median age of voters is about 53 and 50% voters lie within the age range of 41 to 67.
* The age distribution can be better visualized with the histogram shown above.

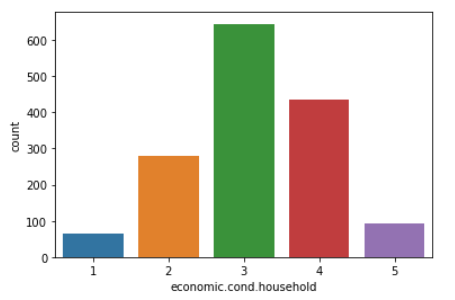
**Economic.cond.national**



**Figure 3** Distribution of national economic condition.

* National economic condition varies on a scale of 1 to 5.
* Most of the voters seem to be from nation with economic condition 3.

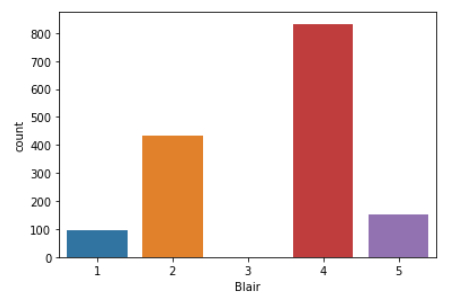
**Economic.cond.household**



**Figure 4** Distribution of national economic condition.

* Household economic condition follows the same trend as national economic condition.

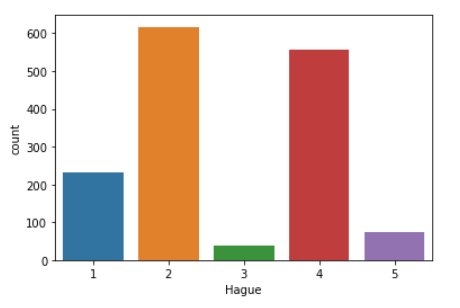
**Blair**



**Figure 5** Assessment of Labour leader

* Most of the voters assessed Labour Leader with a rating of 4.
* Average rating for Labour Leader is 3.33.

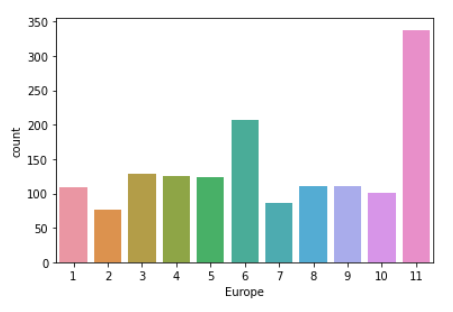
**Hague**



**Figure 6** Assessment of Conservative leader

* Most of the voters assessed Conservative leader with a rating of 2.
* Average rating for Conservative Leader is 2.75.
* One of the big reasons for Labour party to win seems to be the better leader acceptance by the voters.

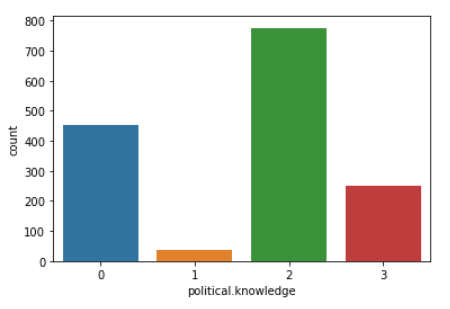
**Europe**



**Figure 7** Attitude towards European Integration

* Voters are uniformly distributed across the 11-point measure with a hike in voters for values 6 and 11

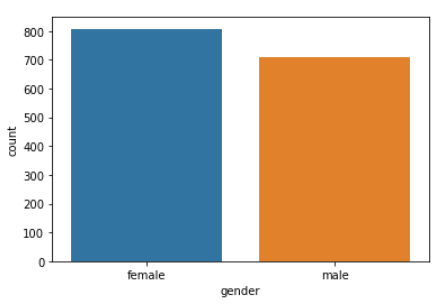
**Political.knowledge**



**Figure 8** Knowledge about party’s stand on European Integration

* Most of the voters are rated at a 2 for political knowledge implying most of the voters have decent amount of information regarding the parties stand on European Integration.

**Gender**

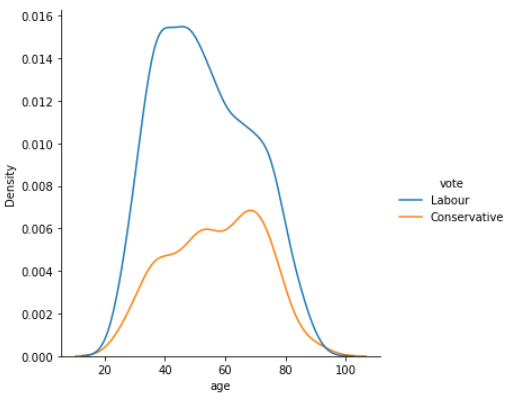
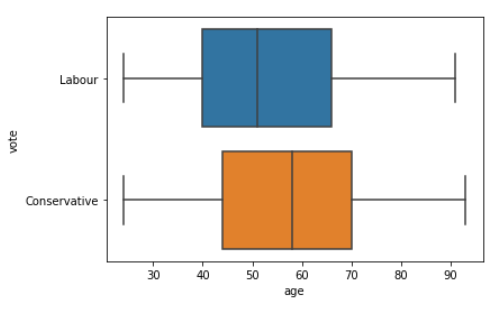


**Figure 9** Gender distribution of voters.

* More number of females voted as compared to males signifying a progressive society.
* Parties can also draw the insight to focus more on women issues to win the elections as they are the majority vote bank.

**Bivariate Data Analysis.**

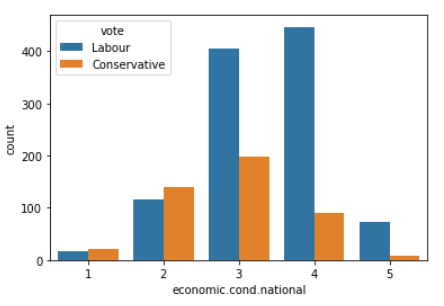
**Voting Pattern and Age.**

**Figure 10** Voting pattern and Age

* As age of voters increase the number of voters increase for Conservative party and number of voters decrease for Labour party.
* Conservative party can take the insight to focus on young generation voters’ issues if they wish to do better in the next elections.
* Median age of voters for conservative party s more than voters of labour party.

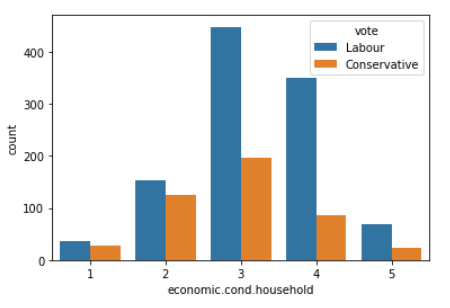
**Voting pattern and Economic.cond.national**



**Figure 11** Voting pattern and assessment of national economic condition.

* People who have rated the national economic condition to be bad have preferred conservative party and people who have rated the national economic condition to be good have preferred labour party suggesting the current ruling party is labour party before the elections happened.

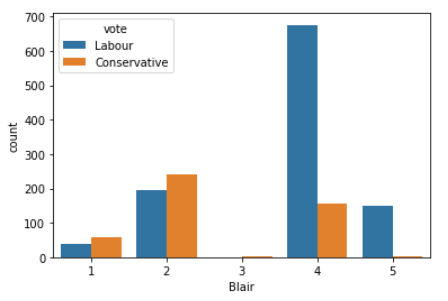
**Voting pattern and Economic.cond.household**



**Figure 12** Voting pattern and assessment of household economic condition.

* People irrespective of their household economic condition have preferred Labour party.

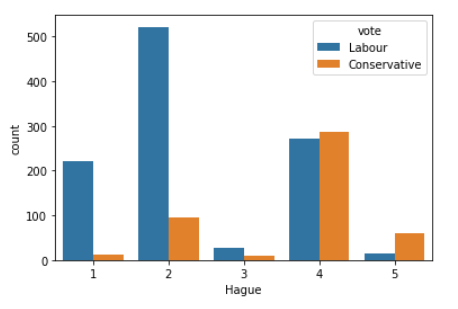
**Voting pattern and Blair**



**Figure 13** Voting pattern and assessment of Labour Leader

* People who have rated Labour Leader low have voted for the Conservative party and people have rated Labour Leader high have voted for Labour party.

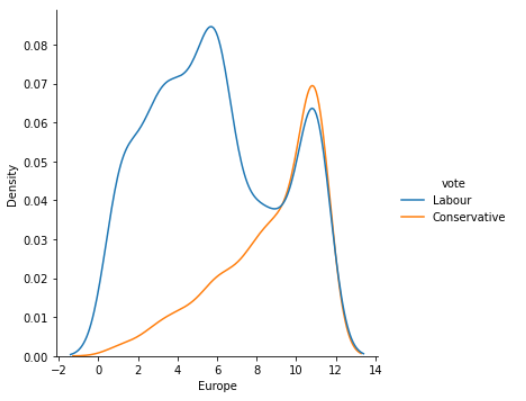
**Voting pattern and Hague**



**Figure 14** Voting pattern and assessment of Conservative Leader

* People who have rated Conservative Leader high have voted for the Conservative party and people have rated Conservative Leader low have voted for Labour party.
* It is very clear from the data that acceptance of Leader is a very significant factor in determining for whom an individual is going to vote for.

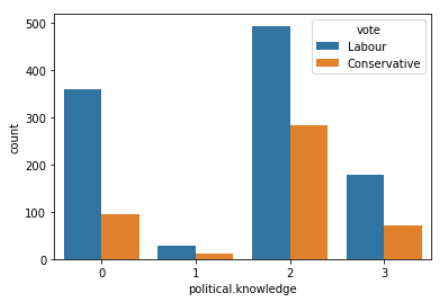
**Voting pattern and Europe**



**Figure 15** Voting pattern and Eurosceptic Sentiment.

* Except for a few exceptions, as the Eurosceptic Sentiment increases, more people tend to vote for the Conservative party as compared to the Labour party.

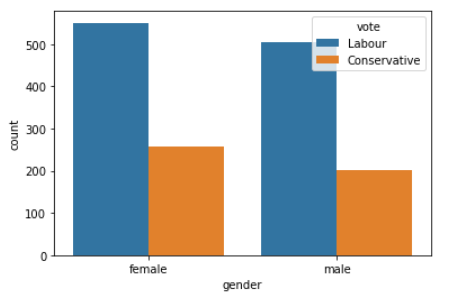
**Voting pattern and Political.knowledge**



**Figure 16** Voting pattern and Political Knowledge.

* Irrespective of their political knowledge voters seem to prefer Labour party as compared to Conservative Party.

**Voting pattern and Gender**

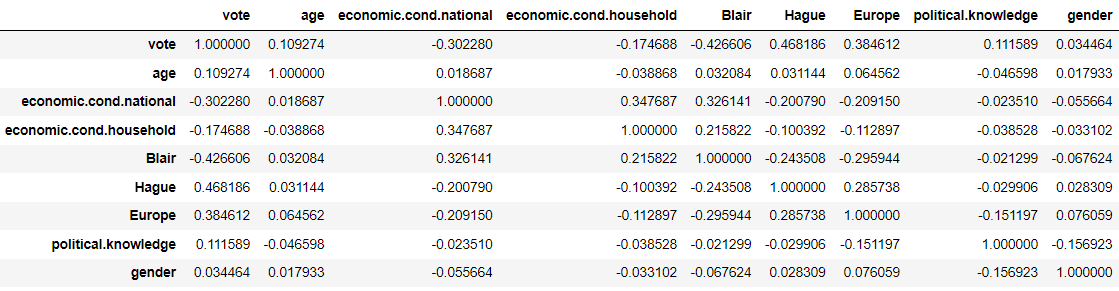


**Figure 17** Voting pattern and gender.

* Voters irrespective of their genders preferred Labour Party.

**Multivariate Data Analysis.**

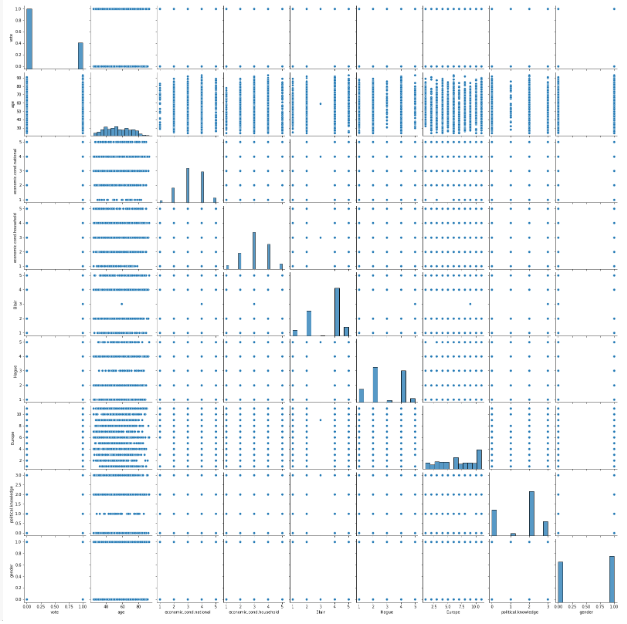
**Correlation Analysis.**



**Table III** Correlation Analysis.

* There is no obvious strong linear correlation between any two variables.

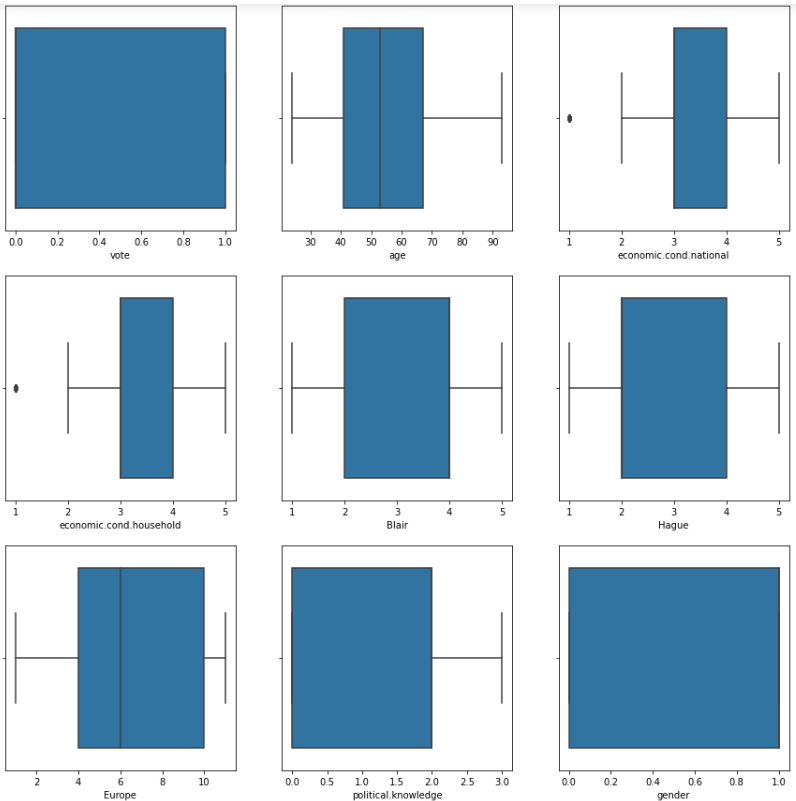
**Pairplot Analysis.**



**Figure 18** Pair-plot Analysis.

* There are no actionable insights that can be drawn from the pair-plot analysis.

**Outlier check.**



**Figure 19** Boxplot analysis to check for outliers.

* There are outliers in attributes *economic.cond.national* and *economic.cond.household*.
* Many ML algorithms can be sensitive to outliers but here we choose to not treat outliers as the value 1 itself has an ordinal meaning.

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

**Encoding**

* There were two attributes with object data type: vote and gender that required encoding.
* They were encoded using replace function of a data frame.
* For attribute vote, ‘Labour’ was encoded with 0 and ‘Conservative’ was encoded as 1.
* For attribute gender, ‘male’ was encoded as 0 and ‘female’ was encoded as 1.

**Scaling**

* Scaling is always a good idea for distance-based algorithms.
* It reduces the bias introduced towards a specific attribute which is high in magnitude.
* Since it was required to use the data set to train various models which were distance based, it is a good idea to scale the dataset.

**Splitting Dataset**

* The data set was first split into dependent and independent variables.
* Then the data was split into training and testing data using the function *train\_test\_split* from library *sklearn.model\_selection* such that 70% of the datapoints are used to train the data and remaining 30% of the datapoints are used for testing.

**1.4 Apply Logistic Regression and LDA (linear discriminant analysis).   
1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results.   
1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging) and boosting.   
1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized.**

**Note:** Questions 1.4 to 1.7 are solved together in the following section model wise.

**Model Tuning Strategy used.**

**Goal:**

To identify the correct number of votes each party is getting to predict number of seats/election result.

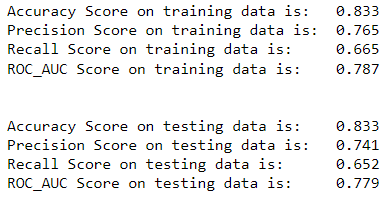
**Note:**

Here it is not that significant to make sure that each vote is predicted correctly. Else, if we can make sure that the number of votes incorrectly predicted for one class is equal to the number of votes incorrectly predicted for another class. i.e. **Number of False positives = Number of False Negatives**, we will still be able to 100% correctly predict the sum of votes each party is getting. This can be further described as **Recall = Precision**. Even though for this, Accuracy parameter might be less, but still the end goal to predict election result will be achieved in an optimized way.

**Logistic Regression**

Logistic Regression was imported from library sklearn.linear\_model and was trained using the training data and then tested with both training and testing data with default hyper-parameters and the model performance metrics were as follows.

**Performance Metrics**



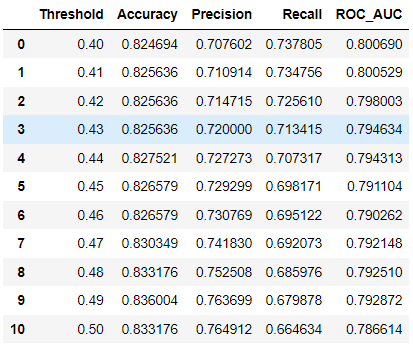
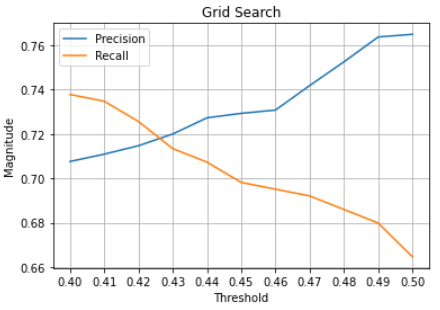
**Figure 20** Performance Metrics Logistic Regression

**Insights**

* Accuracy score is equal for training and testing data is equal, the model in not over/under fitted.
* There is a difference in Recall and Precision values, we need to tune the model.

**Model Tuning**

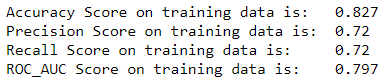
We are choosing to tune our model as per training data by doing a grid search on various values of threshold.

**Figure 21** Grid Search for optimum threshold.

A threshold value of **0.428** is finalized as our new threshold and the performance metrics for training and testing data are shared below.

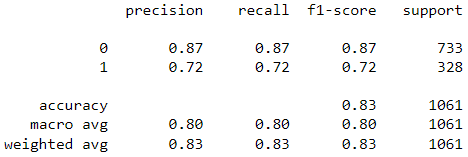
**Training Data**



**Figure 22** Performance metrics after tuning.



**Figure 23**  Confusion Matrix

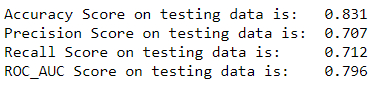


**Figure 24**  Classification Report



**Figure 25**  ROC Curve

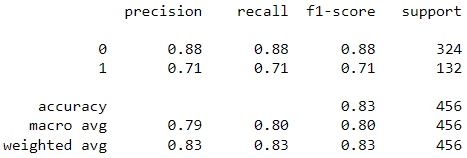
**Testing Data**



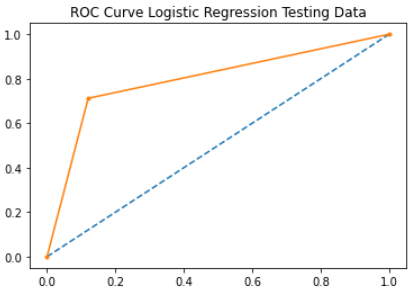
**Figure 26** Performance metrics after tuning.



**Figure 27**  Confusion Matrix



**Figure 28**  Classification Report



**Figure 29**  ROC Curve

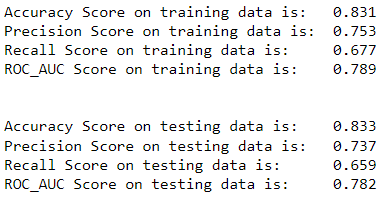
**Conclusion:**

Accuracy for testing and training data are comparable and number of false positives and false negatives is almost equal for testing data as well. i.e. we have a good model as per our requirement.

**Linear Discriminant Analysis**

Linear Discriminant Classifier was imported from library sklearn.discriminant\_analysis and was trained using the training data and then tested with both training and testing data with default hyper-parameters and the model performance metrics were as follows.

**Performance Metrics**



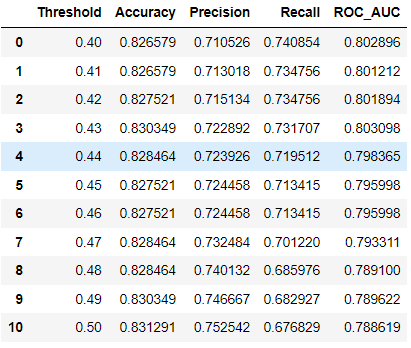
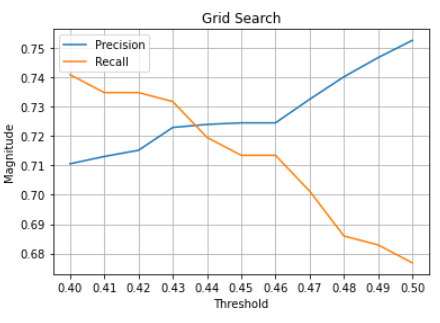
**Figure 30** Performance Metrics Logistic Regression

**Insights**

* Accuracy score is similar for training and testing data, the model in not over/under fitted.
* There is a difference in Recall and Precision values, we need to tune the model.

**Model Tuning**

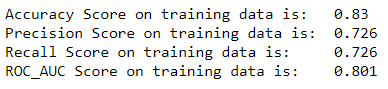
We tune our model using training data by doing a grid search on various values of threshold.

**Figure 31** Grid Search for optimum threshold.

A threshold value of **0.436** is finalized as our new threshold and the performance metrics for training and testing data are shared below.

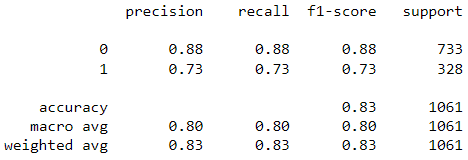
**Training Data**



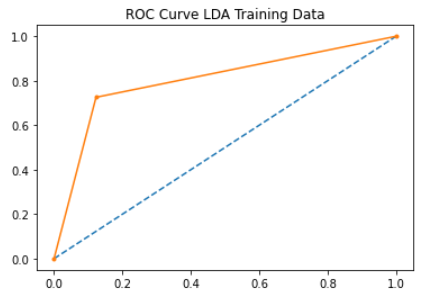
**Figure 32** Performance metrics after tuning.



**Figure 33**  Confusion Matrix

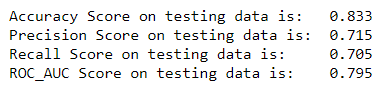


**Figure 34**  Classification Report



**Figure 35**  ROC Curve

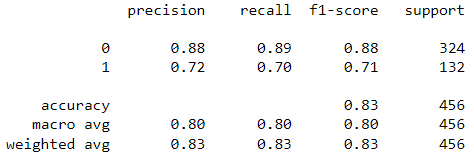
**Testing Data**



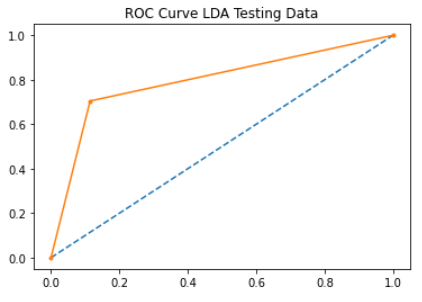
**Figure 36** Performance metrics after tuning.



**Figure 37**  Confusion Matrix



**Figure 38**  Classification Report



**Figure 39**  ROC Curve

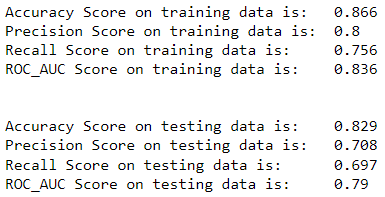
**Conclusion:**

Accuracy for testing and training data are comparable and number of false positives and false negatives is almost equal for testing data as well. i.e. we have a good model as per our requirement.

**K-Nearest Neighbors**

KNN Classifier was imported from library sklearn.neighbors and was trained using the training data and then tested with both training and testing data with default hyper-parameters and the model performance metrics were as follows.

**Performance Metrics**



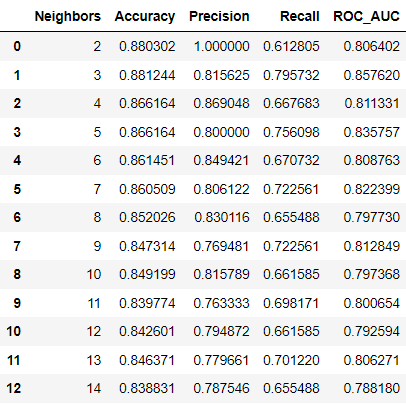
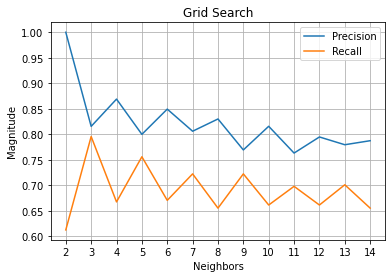
**Figure 40** Performance Metrics KNN

**Insights**

* Accuracy score is similar for training and testing data, the model in not over/under fitted.

**Model Tuning**

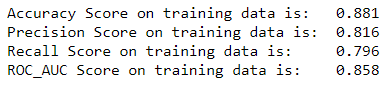
We tune our model using training data by doing a grid search on various values of neighbors.

**Figure 41** Grid Search for optimum number of neighbors.

As for 3 number of neighbors, we see highest accuracy and closest value of precision and recall, thus we pick the same for our KNN Model. Performance metric are shared for training and testing data.

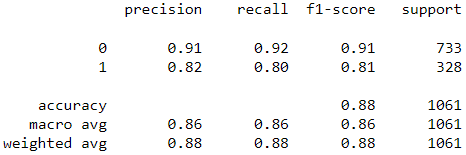
**Training Data**



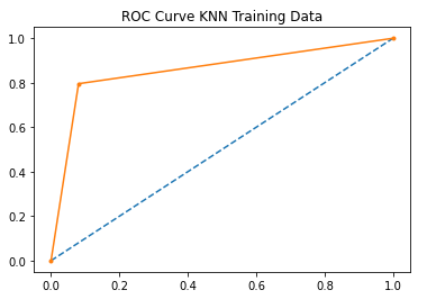
**Figure 42** Performance metrics after tuning.



**Figure 43**  Confusion Matrix

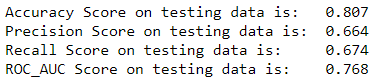


**Figure 44**  Classification Report



**Figure 45**  ROC Curve

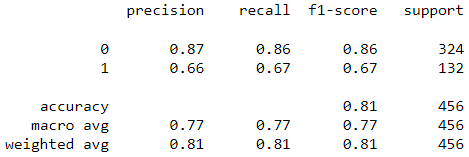
**Testing Data**



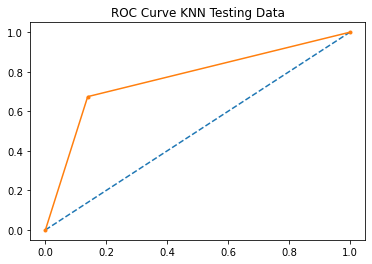
**Figure 46** Performance metrics after tuning.



**Figure 47**  Confusion Matrix



**Figure 48**  Classification Report



**Figure 49**  ROC Curve

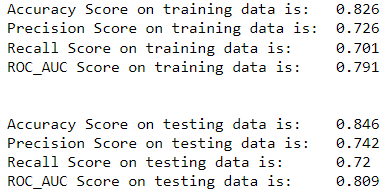
**Conclusion:**

There is a decrease in accuracy when we test on testing data. Even though number of False positive and false negatives are almost equal, still they are high in number. The model is slightly overfitted for training data and is not that good.

**Naïve Bayes**

Naïve Bayes Classifier was imported from library sklearn.naive\_bayes and was trained using the training data and then tested with both training and testing data with default hyper-parameters and the model performance metrics were as follows.

**Performance Metrics**



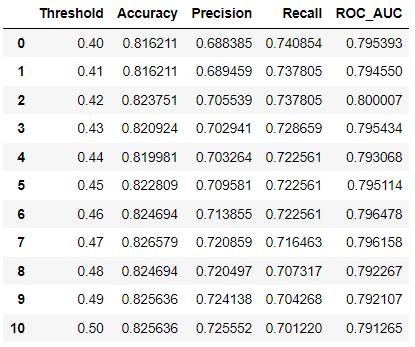
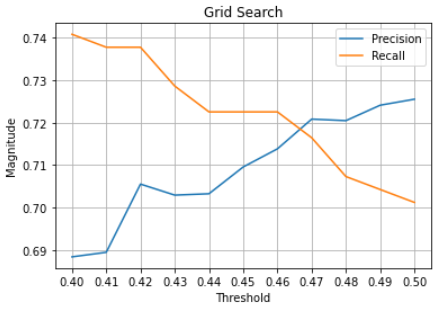
**Figure 50** Performance Metrics KNN

**Insights**

* Accuracy score is similar for training and testing data, the model in not over/under fitted.

**Model Tuning**

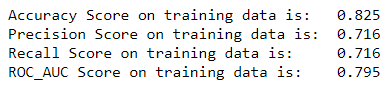
We tune our model using training data by doing a grid search on various values of thresholds.

**Figure 51** Grid Search for optimum number of neighbors.

A threshold value of **0.468** is finalized as our new threshold and the performance metrics for training and testing data are shared below.

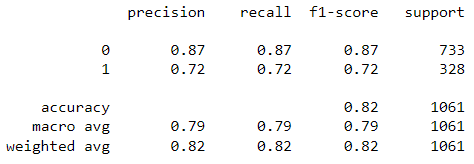
**Training Data**



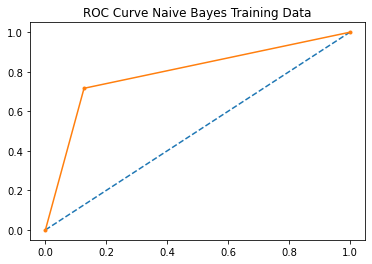
**Figure 52** Performance metrics after tuning.



**Figure 53**  Confusion Matrix

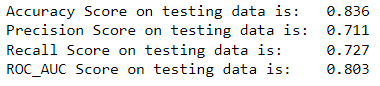


**Figure 54**  Classification Report



**Figure 55**  ROC Curve

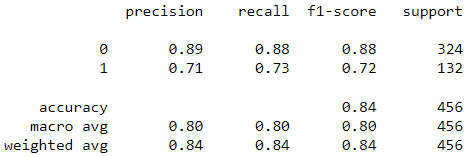
**Testing Data**



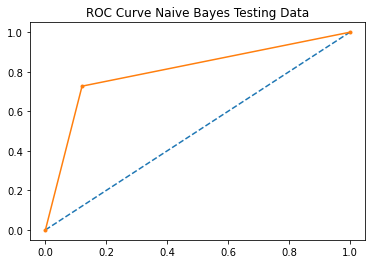
**Figure 56** Performance metrics after tuning.



**Figure 57**  Confusion Matrix



**Figure 58**  Classification Report



**Figure 59**  ROC Curve

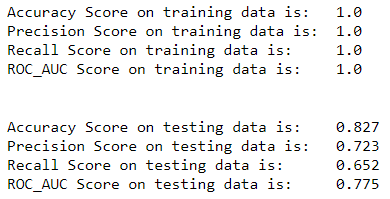
**Conclusion:**

Accuracy for testing and training data are comparable and number of false positives and false negatives is almost equal for testing data as well. i.e., we have a good model as per our requirement.

**Random Forest**

Random Forest Classifier was imported from library sklearn.ensemble and was trained using the training data and then tested with both training and testing data with default hyper-parameters and the model performance metrics were as follows.

**Performance Metrics**



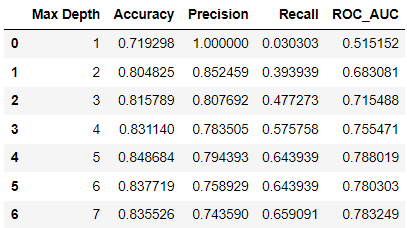
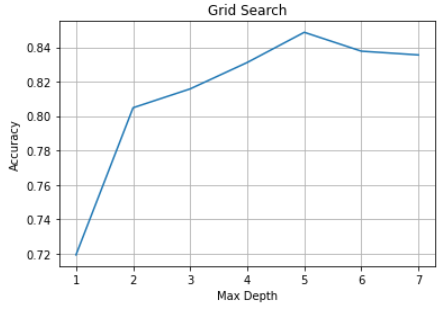
**Figure 60** Performance Metrics Random Forest

**Insights**

* There is a drop of more than 13% in accuracy from training to test data. The model is extremely over-fitted.

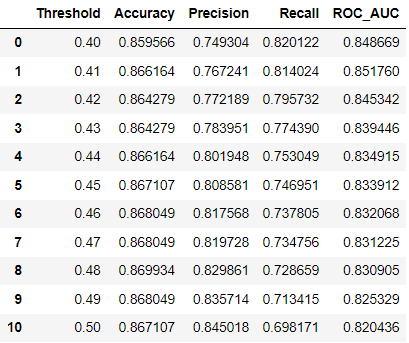
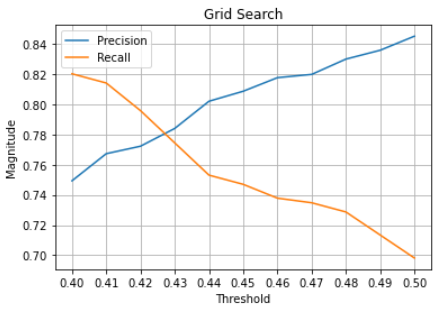
**Model Tuning**

We tune our model using testing data by doing a grid search on various values of max\_depths.

**Figure 61** Grid Search for optimal max\_depth.

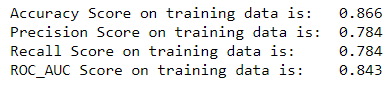
A max\_depth of **5** is finalized as our new max\_depth. Then we tune to an optimal threshold.

**Figure 62** Grid Search for optimal threshold.

A threshold value of **0.427** is finalized as our new threshold and the performance metrics for training and testing data are shared below.

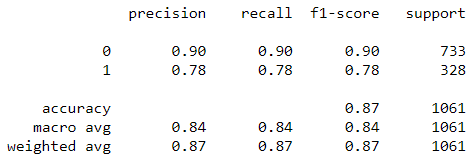
**Training Data**



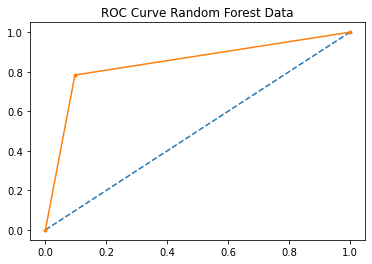
**Figure 63** Performance metrics after tuning.



**Figure 64**  Confusion Matrix

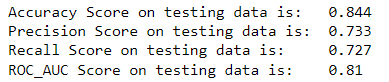


**Figure 65**  Classification Report



**Figure 66**  ROC Curve

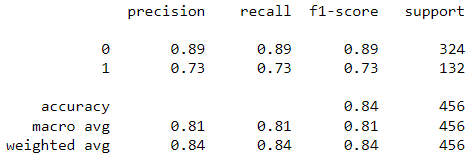
**Testing Data**



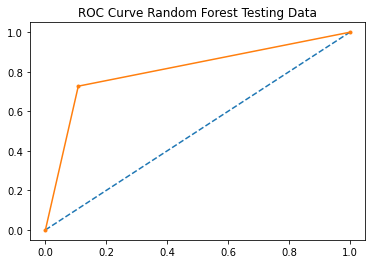
**Figure 67** Performance metrics after tuning.



**Figure 68**  Confusion Matrix



**Figure 69**  Classification Report



**Figure 70**  ROC Curve

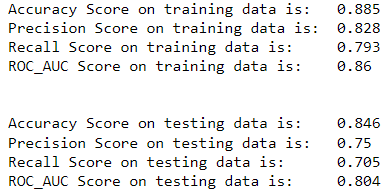
**Conclusion:**

Accuracy for testing and training data are comparable and number of false positives and false negatives is almost equal for testing data as well. i.e., we have a good model as per our requirement.

**Gradient Boosting**

Gradient Boosting Classifier was imported from library sklearn.ensemble and was trained using the training data and then tested with both training and testing data with default hyper-parameters and the model performance metrics were as follows.

**Performance Metrics**



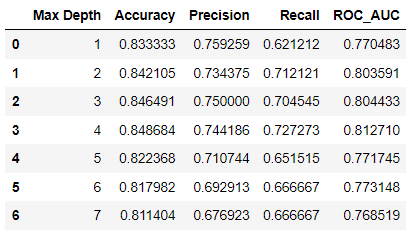
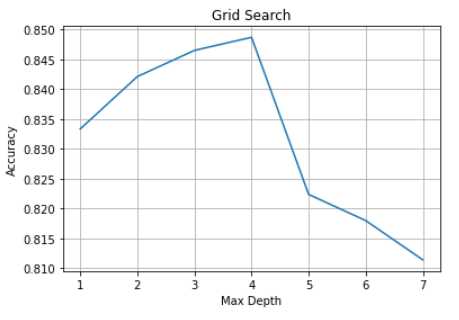
**Figure 71** Performance Metrics Random Forest

**Insights**

* Accuracies for training and testing data are comparable. No overfitting.

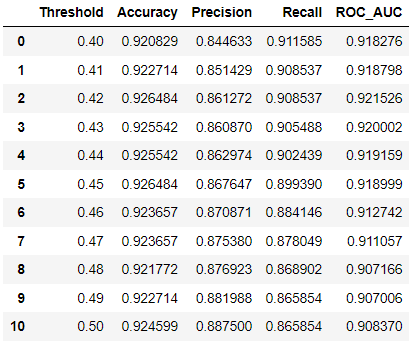
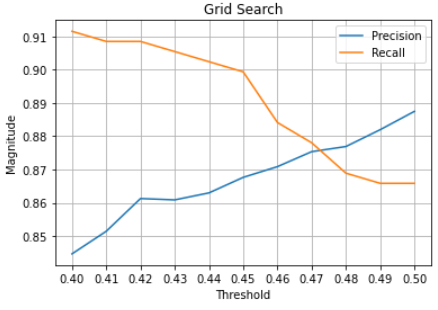
**Model Tuning**

We tune our model using testing data by doing a grid search on various values of max\_depths.

**Figure 72** Grid Search for optimal max\_depth.

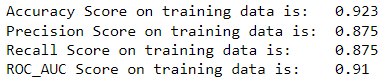
A max\_depth of **4** is finalized as our new max\_depth. Then we tune to an optimal threshold.

**Figure 73** Grid Search for optimal threshold.

A threshold value of **0.475** is finalized as our new threshold and the performance metrics for training and testing data are shared below.

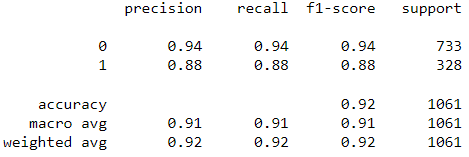
**Training Data**



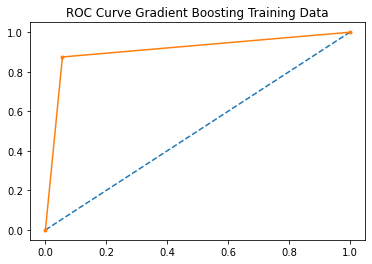
**Figure 74** Performance metrics after tuning.



**Figure 75**  Confusion Matrix

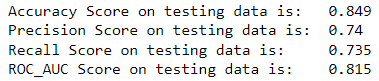


**Figure 76**  Classification Report



**Figure 77**  ROC Curve

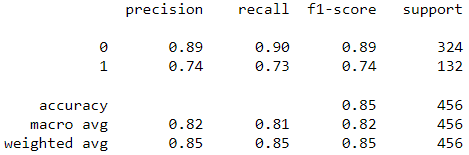
**Testing Data**



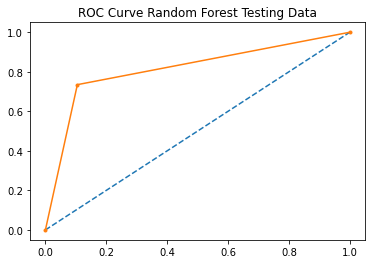
**Figure 78** Performance metrics after tuning.



**Figure 79**  Confusion Matrix



**Figure 80**  Classification Report



**Figure 81**  ROC Curve

**Conclusion:**

Accuracy for testing and training data are comparable and number of false positives and false negatives is almost equal for testing data as well. i.e., we have a good model as per our requirement.

**Comparison**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Model*** | **Accuracy** | **Roc Auc Score** | **False Positives** | **False Negatives** |
| *Logistic Regression* | 0.831 | 0.796 | 39 | 38 |
| *Linear Discriminant Analysis* | 0.833 | 0.795 | 37 | 39 |
| *Naïve Bayes* | 0.836 | 0.803 | 39 | 36 |
| *K-Nearest Neighbors* | 0.807 | 0.768 | 45 | 43 |
| *Random Forest* | 0.844 | 0.81 | 35 | 36 |
| *Gradient Boosting* | 0.849 | 0.815 | 34 | 35 |

**Table IV**  Performance Comparison Table.

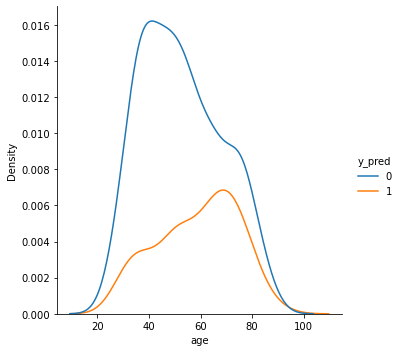
Even though the performance of all the models is comparable, they have been rated from top to bottom in the following list:

* Gradient Boosting
* Random Forest
* Logistic regression
* Linear Discriminant Analysis
* Naïve Bayes
* K-Nearest Neighbors

The order is on the basis for correct prediction of total number of seats won by each party and not the accuracy by which the model predicts each voter’s vote.

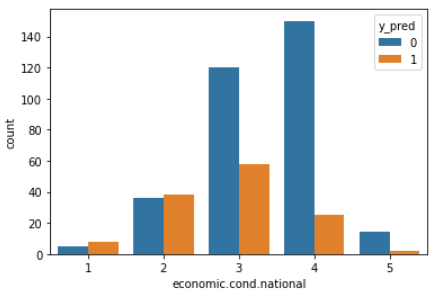
**1.8 Based on these predictions, what are the insights?**

* Based on the predictions, younger people are more likely to vote for Labour Party and elderly population is more likely to vote for Conservative party.



**Figure 82**  Age distribution for voters.

* Based on predictions, people who rate national economic condition to be good are going to vote for labour party and people who rate national economic consition to be bad are going to vote for conservative party.



**Figure 83**  Rating of national economic condition by voters.

* People predicted to vote for Labour party on an average rate Blair more than Hauge.



**Figure 84**  Rating of party leader by voters

* People predicted to vote for Conservative party on an average rate Blair less than Hauge.



**Figure 85**  Rating of party leader by voters

* People with a higher Eurosceptic sentiment are predicted to vote for Conservative Party.



**Figure 86**  Rating of Eurosceptic sentiment