

**Machine Learning Project**

Summited by : Taniya Dubey

INDEX

|  |  |  |
| --- | --- | --- |
| no | title | Page no |
| 1.1 | 1.1 Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it. | 7-9 |
| 1.2 | 1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers. | 10-19 |
| 1.3 | 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 | 20-21 |
| 1.4 | 1.4 Apply Logistic Regression and LDA (linear discriminant analysis). | 22-27 |
| 1.5 | 1.5 Apply KNN Model and Naïve Bayes Model. Interpret the result | 28-32 |
| 1.6 | 1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting | 33-49 |
| 1.7 | 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. | 50-69 |
| 1.8 | 1.8 Based on these predictions, what are the insights? | 70 |
| 2.1 | 2.1 Find the number of characters, words, and sentences for the mentioned documents. | 71 |
| 2.2 | 2.2 Remove all the stopwords from all three speeches. | 71 |
| 2.3 | 2.3 Which word occurs the most number of times in his inaugural address for each president? Mention the top three words. ( | 72 |
| 2.4 | 2.4 Plot the word cloud of each of the speeches of the variable. (after removing the stopwords) | 73 |

List of tabels :

Table 1.1 : Data Dictionary of problem

Table 1.2 : top few rows of dataset

Table 1.3 : Last few rows of the dataset

Table 1.4 : info of dataset

Table 1.5 : stat summary of data

Table 1.6 : skewness of data

Table 1.7: nulls in data

Table 1.8 : encoded data

Table 1.9 : encoded gender

Table 1.10 : encoded vote

Table 1.11 : standard devatation and variance of data

Table 1.12 : standard devatation and variance after scaling age

Table 1.13 : feature importance lr model

Table 1.14 : feature importance lda model

Table 1.15 : feature importance knn model

Table 1.16 : feature importance naïve bayes model

Table 1.17 : feature importance bagging model

Table 1.18 : feature importance gradient boosting

Table1.19 accuracy LR

Table 1.20 : feature importance of beat lr model

Table 1.21 : accuracy of lda

Table 1.22 : feature importance of beat lda model

Table 1.23 : accuracy of knn

Table 1.24 : feature importance of best knn model

Table 1.25 : accuracy of NB

Table 1.26 : feature importance of best NB model

Table 1.27: accuracy of Bagging

Table 1.28 : feature importance of best NB model

Table 1.29: accuracy of best ada boosting

Table 1.30 : feature importance of best ada boosting model

Table 1.31: accuracy of best g boosting

Table 1.32 : feature importance of best g boosting model

Table 1.33 : Table of all models accuracy

Table 2.1 : count table

Table 2.2 : before, after word count

Table 2.3 : sentences before and after removing stop word

Table 2.4 : most comon words

List of fig :

Fig 1.1 : pairplot

Fig 1.2 : correlation plot and covariance plot

Fig 1.3 : countplot of variables with vote

Fig 1.4 : count plot of variables

Fig 1.5 : box plot of variables

Fig 1.6 : plot of variables

Fig 1.7 : barplot of variables with gender and vote

Fig 1.8: feature importanceof lr model

Fig1.9 : auc roc curve of lr model

Fig 1.10 : confusion matrix of lrmodel

Fig 1.11: feature importanceof lda model

Fig 1.12 : auc roc plot of lda model

Fig 1.13 : confusion matrixof the knn model

Fig 1.14: feature importanceof knn model

Fig 1.15 : aur roc curve of knn model

Fig 1.16 : confusion matrix of knn

Fig 1.17: feature importance of naïve bayes model

Fig 1.18 : aoc roc curve for naive bayes

Fig 1.19 : confusion matric of naïve bayes

Fig 1.20: feature importance of bagging model

Fig 1.21: aoc roc curve bagging model

Fig 1.22: feature importance of boosting model

Fig 1.23: aoc roc curve boosting model

Fig 1.24: feature importance of gradient boosting model

Fig 1.25: aoc roc curve boosting model

Fig 1.26 : feature importance of best lr model

Fig 1.27 : feature importance of best lda model

Fig 1.28 : feature importance of best knn model

Fig 1.29 : feature importance of best NB model

Fig 1.30 : feature importance of best Bagging model

Fig 1.31: feature importance of best Boostinging model

Fig 1.32: feature importance of best g Boosting model

Fig 1.33: confusion matrix of blrmodel

Fig 1.34 : roc and auc curve of best and default model

Fig 1.35: confusion matrix of blda model

Fig 1.36 : roc and auc curve of best and default model

Fig 1.37: confusion matrix of knn model

Fig 1.38 : roc and auc curve of best and default model

Fig 1.39: confusion matrix of best model

Fig 1.40: roc and auc curve of best and default model

Fig 1.41: confusion matrix of bestmodel

Fig 1.42 : roc and auc curve of best and default model

Fig 1.43: confusion matrix of bagging model

Fig 1.43: confusion matrix of bagging model

Fig 1.44 : roc and auc curve of best and default model

Fig 1.45: confusion matrix of best rmodel

Fig 1.46 : roc and auc curve of best and default model

Fig 2.1: word cloud of Roosevelt speech

Fig 2.2 : word cloud of Kennedy speech

Fig 2.3 : word cloud of Nixon Speech

---------------------------------------Problem 1:-----------------------------------------

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

**Data Dictionary :**

|  |  |  |
| --- | --- | --- |
| **1** | Vote : | Party choice: Conservative or Labour |
| **2** | age: | in years |
| **3** | economic.cond.national: | Assessment of current national economic conditions, 1 to 5. |
| **4** | economic.cond.household: | Assessment of current household economic conditions, 1 to 5. |
| **5** | Blair : | Assessment of the Labour leader, 1 to 5. |
| **6** | Hague: | Assessment of the Conservative leader, 1 to 5. |
| **7** | Europe: | an 11-point scale that measures respondents' attitudes toward European integration. High scores represent ‘Eurosceptic’ sentiment. |
| **8** | political.knowledge: | Knowledge of parties' positions on European integration, 0 to 3. |
| **9** | gender: | female or male. |

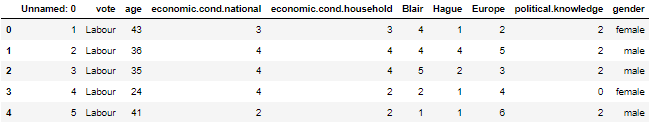
**Table 1.1 : Data Dictionary of problem**

## **1.1) Read the dataset. Describe the data briefly. Interpret the inferences for each. Initial steps like head() .info(), Data Types, etc . Null value check, Summary stats, Skewness must be discussed.**

Dataset Shape :

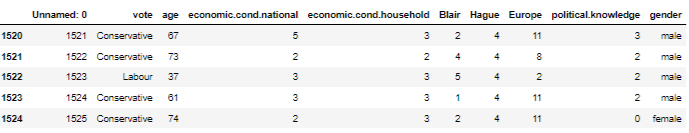
* The dataset has 1525 rows and 10 colunms .

Top few rows of dataset :



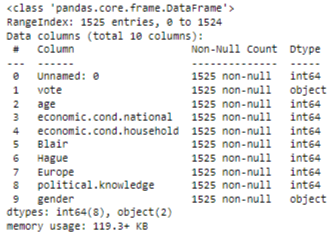
**Table 1.2 : top few rows of dataset**

Last few rows of dataset :



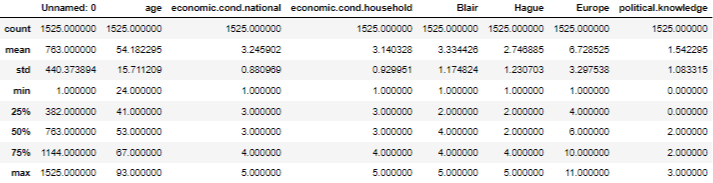
**Table 1.3 : Last few rows of the dataset**

Info of the dataset :



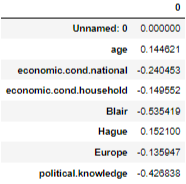
**Table 1.4 : info of dataset**

Stat summary of the dataset :



**Table 1.5 : stat summary of data**

Skewness of the dataset :



**Table 1.6 : skewness of data**

Null Value in the Dataset :

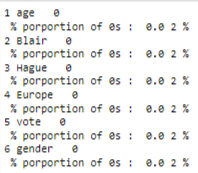


**Table 1.7: nulls in data**

Duplicate Value in the Dataset :

There is no duplicate value dataset.

0’s in the dataset :



Insights:

* Dataset has no alarming level of 0s values in the data.
* Data set has no null values and duplicate values.
* Data set a column named ‘Unnamed:0’ which is not need as its only a sch.no column.
* Age and hauge feature are positive skewed or right skewed and other features are left skewed.
* vote is target variable which means its dependent variable.

## **1.2) Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also check for outliers (4 pts). Interpret the inferences for each (3 pts) Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Outliers proportion should be discussed, and inferences from above used plots should be there.**

Null Value in the Dataset :



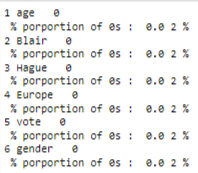
**Table 1.7: nulls in data**

**There are no null values**

Duplicate Value in the Dataset :

There is no duplicate value dataset.

0’s in the dataset :

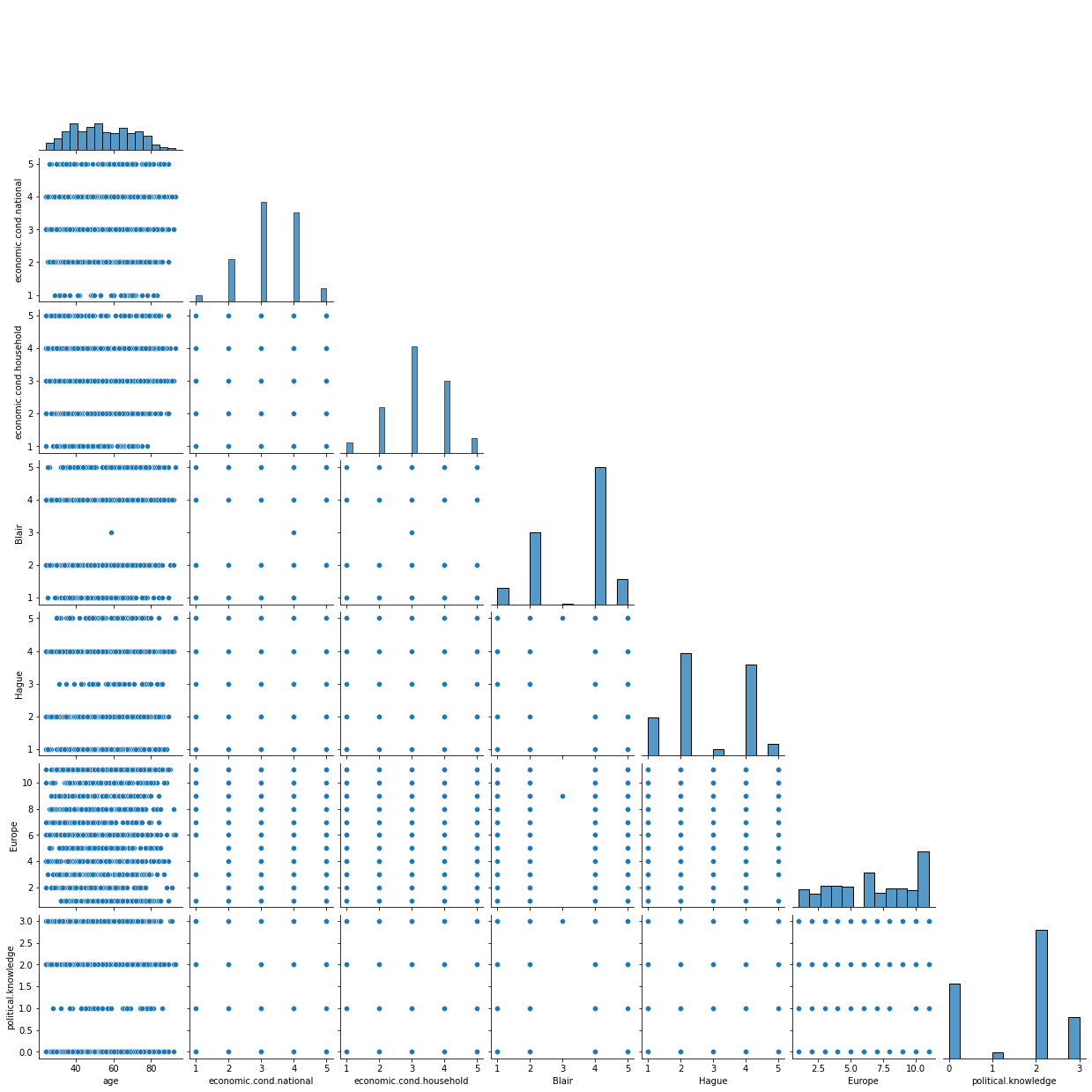


There are no 0s in dataset.

Drop column 'Unnamed' as its only a serial number column.

### **Bivariate Analysis :**

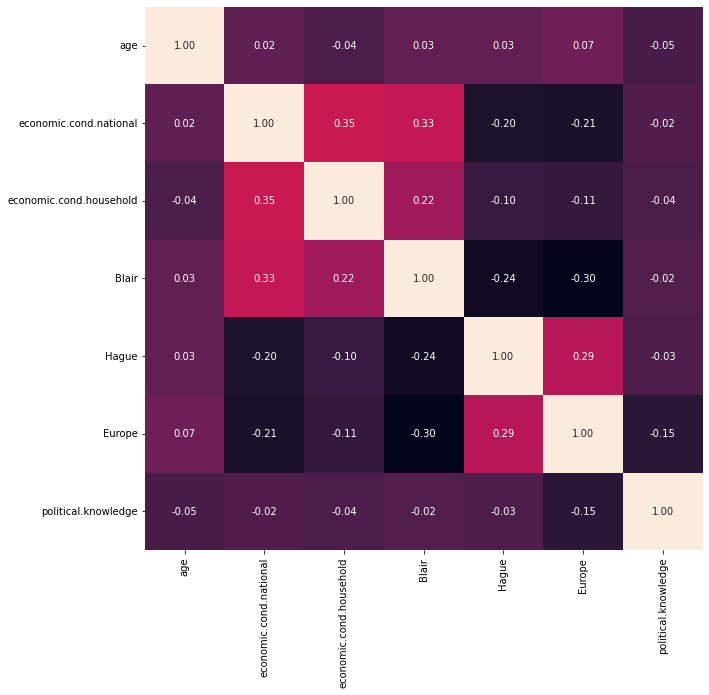
Pair Plot :

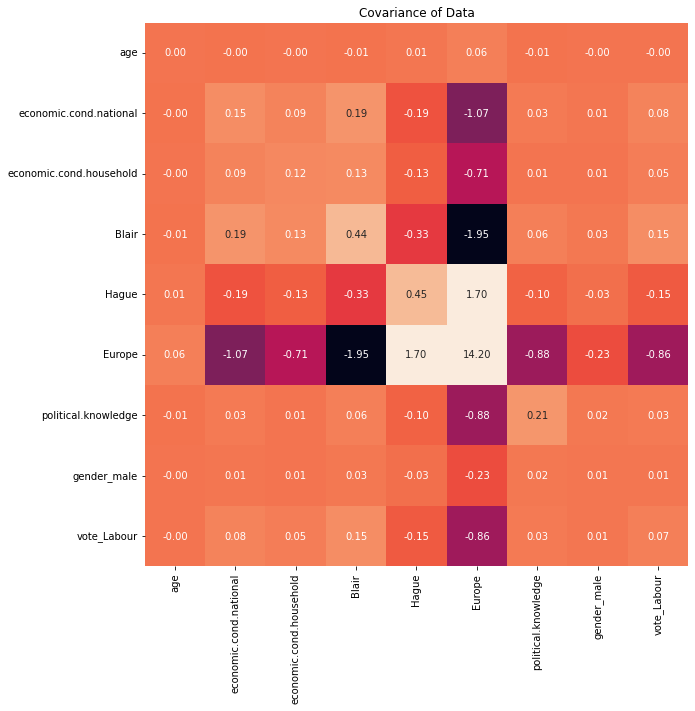


**Fig 1.1 : pairplot**

**There are not much correlation between variables.**

Correlation plot :





**Fig 1.2 : correlation plot and covariance plot**

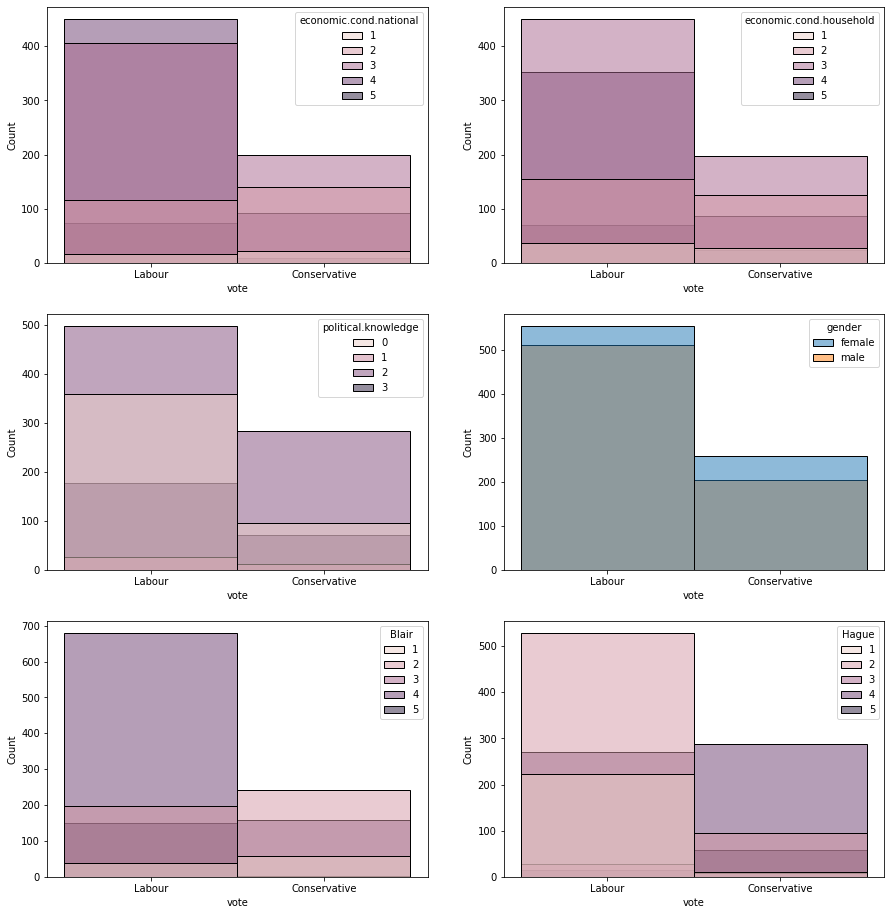
The variables with high positive covariance are:

* Hague and europe

The variables with high negative covariance are:

* Europe and economic.cond.national
* Europe and political knowledge
* Europe vote laboure
* Europe and econimic cond.household
* Europe and blair

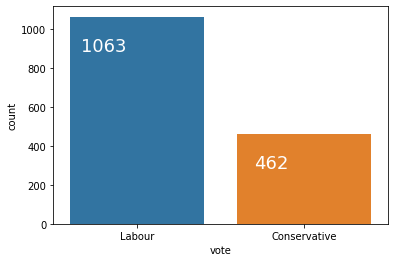
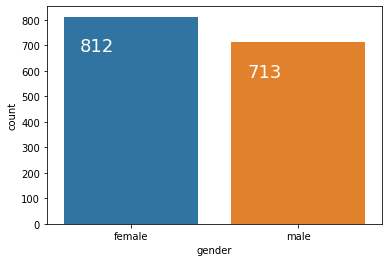
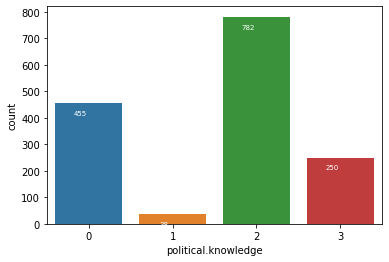
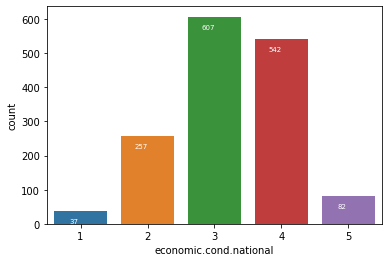
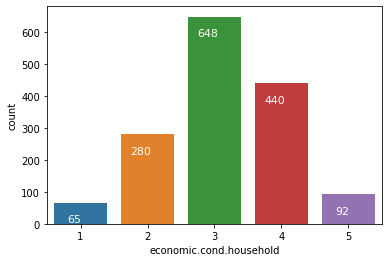
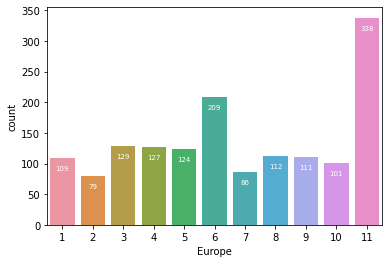
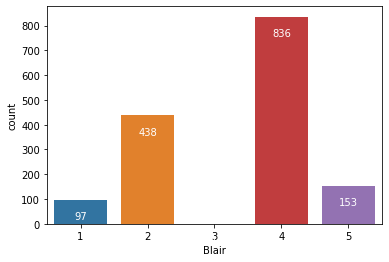
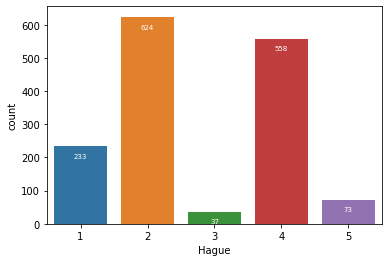
Countplot of other variable with vote :



**Fig 1.3 : countplot of variables with vote**

##### ***Univariate Analysis :***

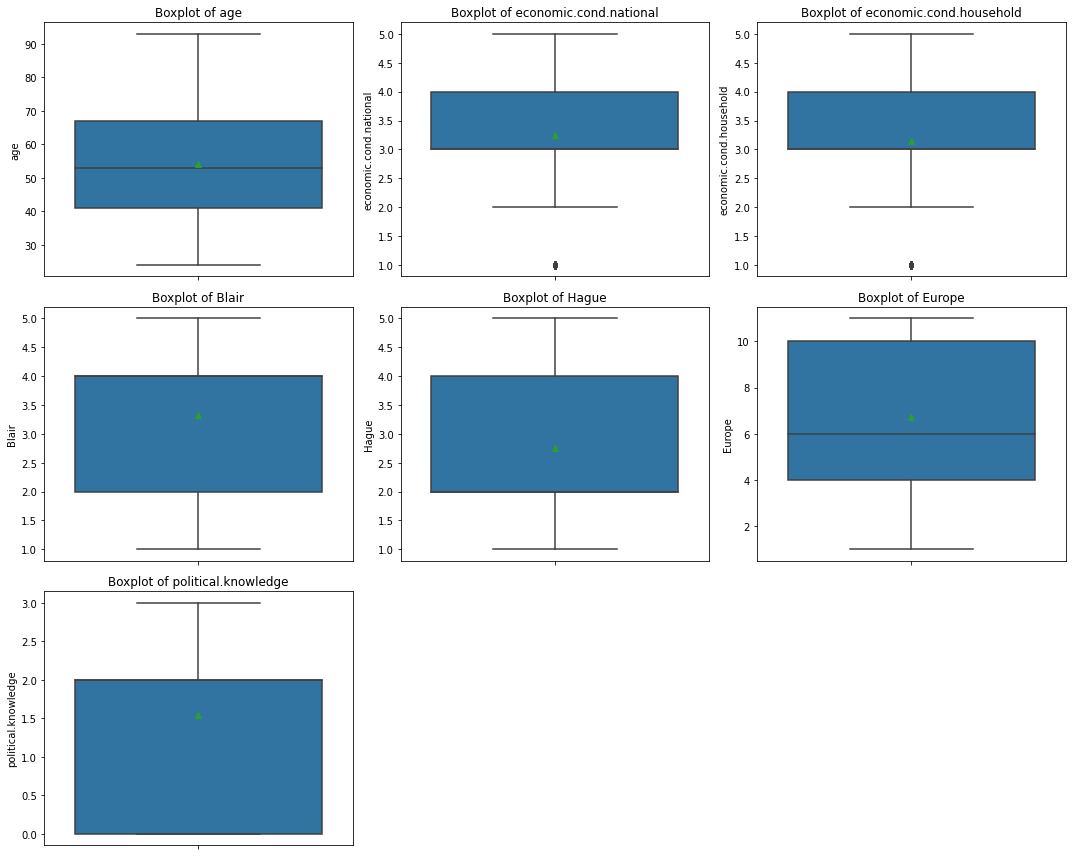
Count plot of categorical plot :



**Fig 1.4 : count plot of variables**

* **Gender is almost equal female and male**
* **Vote feature party choice for labour is more**
* **Most people have little (level 2 ) Political knowledge**

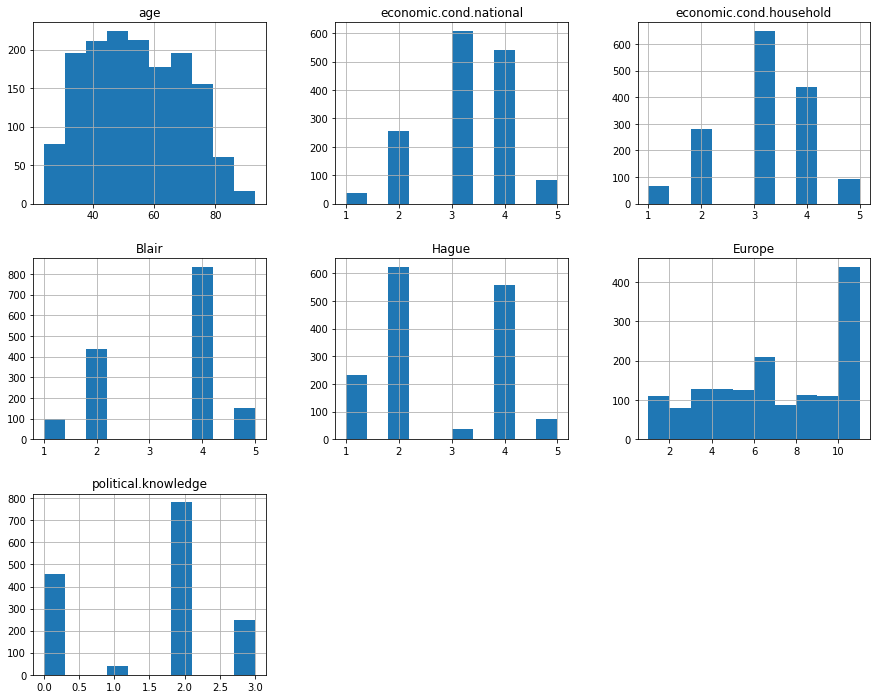
Boxplot of features :



**Fig 1.5 : box plot of variables**

Hague and Europe have outliers but its not alarming so we will not treat them.

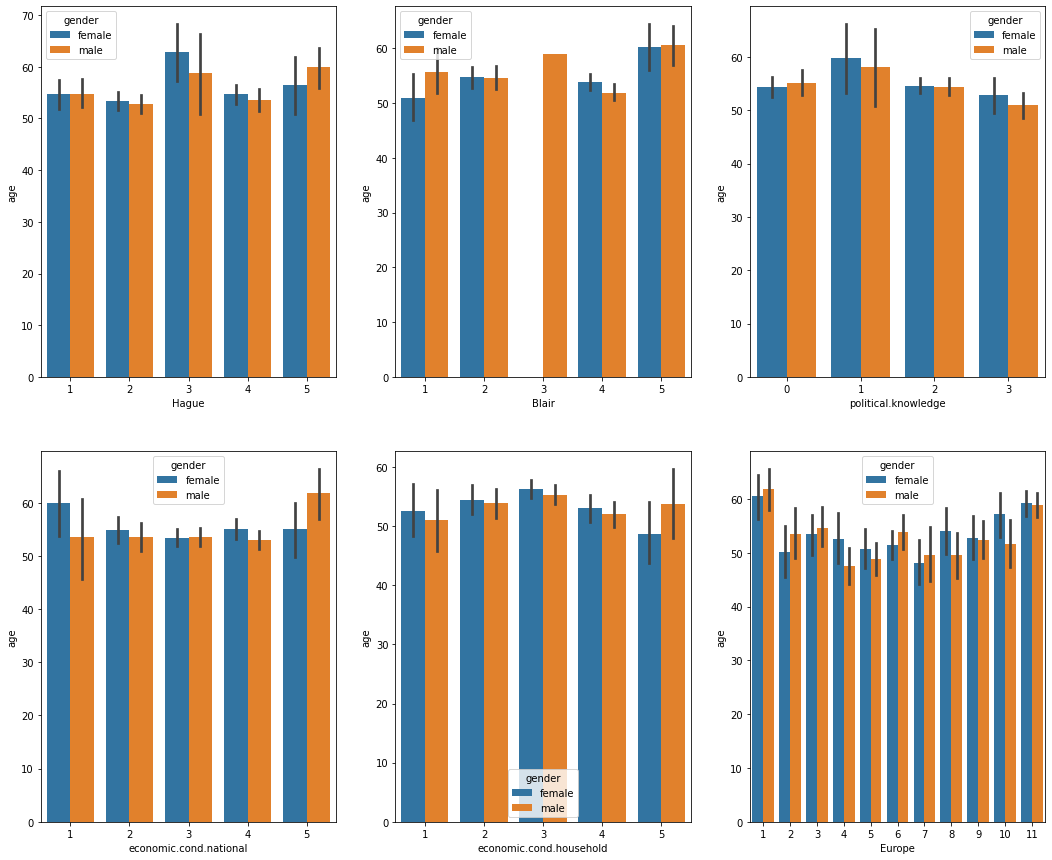
Histplot of features:



**Fig 1.6 : plot of variables**

Age is close to normal distribution.

Multivariate Analysis



**Fig 1.7 : barplot of variables with gender and vote**

Portion of outliers : (as there are only two feature with outliers)  
economic.cond.national :  
Number of outliers: 37  
Proportion of outliers: 2.43 %  
  
economic.cond.household :  
Number of outliers: 65  
Proportion of outliers: 4.26 %

## **1.3) Encode the data (having string values) for Modelling. Is Scaling necessary here or not?( 2 pts), Data Split: Split the data into train and test (70:30) (2 pts). The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for example std dev, variance, etc. Should justify whether there is a necessity for scaling. Object data should be converted into categorical/numerical data to fit in the models. (pd.categorical().codes(), pd.get\_dummies(drop\_first=True)) Data split, ratio defined for the split, train-test split should be discussed.**

Encoded data :



**Table 1.8 : encoded data**

|  |  |
| --- | --- |
| Gender | |
| **0** | **812** |
| **1** | **713** |

**Table 1.9 : encoded gender**

|  |  |
| --- | --- |
| Vote | |
| **0** | **462** |
| **1** | **1063** |

**Table 1.10 : encoded vote**

|  |  |  |
| --- | --- | --- |
| columns | **Standard deviation** | **variance** |
| **age** | 15.711209 | 246.842075 |
| **economic.cond.national** | 0.880969 | 0.776107 |
| **economic.cond.household** | 0.929951 | 0.864810 |
| **Blair** | 1.174824 | 1.380212 |
| **Hague** | 1.230703 | 1.514631 |
| **Europe** | 3.297538 | 10.873759 |
| **political.knowledge** | 1.083315 | 1.173571 |
| **gender\_male** | 0.499109 | 0.249110 |
| **vote\_Labour** | 0.459685 | 0.211310 |

**Table 1.11 : standard devatation and variance of data**

Need to scale data :

* As we can see that except age all features have similar standard deviation and variance.
* For normal logistic regression model scaling in the compulsory.
* For LDA, Naïve Bayes, KNN scaling is necessary.
* For bagging and boosting scaling is important.

|  |  |  |
| --- | --- | --- |
| columns | **Standard Deviation** | **vriance** |
| **age** | 0.227699 | 0.051847 |
| **economic.cond.national** | 0.880969 | 0.776107 |
| **economic.cond.household** | 0.929951 | 0.864810 |
| **Blair** | 1.174824 | 1.380212 |
| **Hague** | 1.230703 | 1.514631 |
| **Europe** | 3.297538 | 10.873759 |
| **political.knowledge** | 1.083315 | 1.173571 |
| **gender\_male** | 0.499109 | 0.249110 |
| **vote\_Labour** | 0.459685 | 0.211310 |

**Table 1.12 : standard deviation and variance after scaling age**

**Data split ( 70 : 30 ):**

Splitting data in to 70% training and 30% testing data.

X\_train : 1067 rows and 8 columns  
Y\_train : 1067 rows and 1 column  
X\_test : 458 rows and 8 columns  
Y\_test : 458 rows and 1 column

## **1.4) Apply Logistic Regression and LDA (Linear Discriminant Analysis) (2 pts). Interpret the inferences of both model s (2 pts). Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)**

**(i) Logical Regression model without tuning**

Default Parameters for logical Regression model :

'C': 1.0, (inverse if regularization strength must be positive float)  
 'class\_weight': None,  
 'dual': False,  
 'fit\_intercept': True,  
 'intercept\_scaling': 1,  
 'l1\_ratio': None,  
 'max\_iter': 100,  
 'multi\_class': 'auto',  
 'n\_jobs': None,  
 'penalty': 'l2',  
 'random\_state': None,  
 'solver': 'lbfgs',(multiclass pooblem only use this)  
 'tol': 0.0001,  
 'verbose': 0,  
 'warm\_start': False

Accuracy of the datasets :

score for train data: 0.833  
score for test data: 0.838

Features importance :

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | -1.00872 |
| **1** | **economic.cond.national** | 0.44905 |
| **2** | **economic.cond.household** | 0.02839 |
| **3** | **Blair** | 0.58223 |
| **4** | **Hague** | -0.75677 |
| **5** | **Europe** | -0.22496 |
| **6** | **political.knowledge** | -0.41711 |
| **7** | **gender\_male** | 0.12031 |

Table 1.13 : feature importance lr model

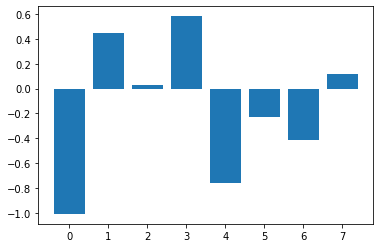


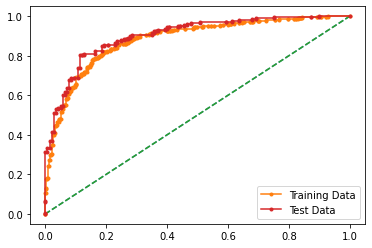
Fig 1.8: feature importance of lr model

Classification report of datasets :

Classification Report of the training data:  
  
 precision recall f1-score support  
  
 0 0.76 0.66 0.70 322  
 1 0.86 0.91 0.88 745  
  
 accuracy 0.83 1067  
 macro avg 0.81 0.78 0.79 1067  
weighted avg 0.83 0.83 0.83 1067  
   
  
----------------------------------------------------------   
  
Classification Report of the test data:  
  
 precision recall f1-score support  
  
 0 0.79 0.64 0.71 140  
 1 0.85 0.92 0.89 318  
  
 accuracy 0.84 458  
 macro avg 0.82 0.78 0.80 458  
weighted avg 0.83 0.84 0.83 458

AUC and ROC of the datasets:

AUC for the Training Data: 0.882  
AUC for the Test Data: 0.902



**Fig1.9 : aur roc curve of lr model**

Confusion matrix od datasets :

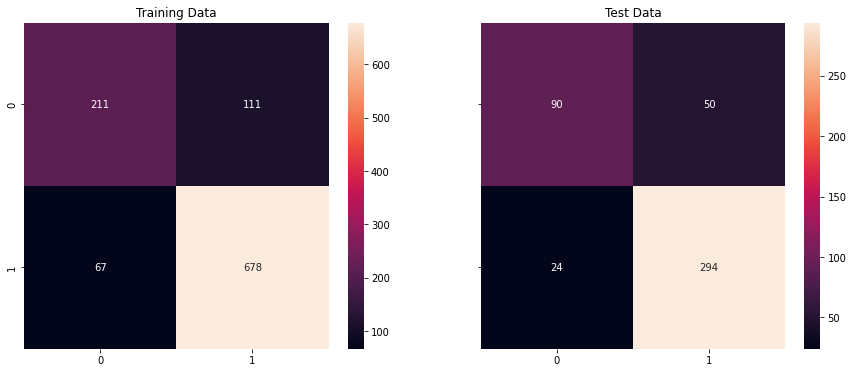


Fig 1.10 : confusion matrix of lrmodel

Inferences :

* From above classification report we can see that our model is neither over fit nor under it.
* Conservative party recall value in train and test data has difference of less than 10%.
* Labour party recall value in train and test dataset has difference of less then 10%.
* Conservative party precision value in train and test dataset difference is null.
* Labour party precision value in trian and test dataset difference is null.

-----------------------------------------------------------------------------------------------------------------------------

**(ii) Linear Descriminate Analysis (LDA)model without tuning**

Default Parameters for LDA model :

'covariance\_estimator': None,

'n\_components': None,

'priors': None,

'shrinkage': None,

'solver': 'svd',

'store\_covariance': False,

'tol': 0.0001

Accuracy of the model :

Accuracy - Training Data: 0.835  
Accuracy - Test Data: 0.829

Features importance :

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | -1.63859 |
| **1** | **economic.cond.national** | 0.42134 |
| **2** | **economic.cond.household** | 0.07214 |
| **3** | **Blair** | 0.76546 |
| **4** | **Hague** | -0.96352 |
| **5** | **Europe** | -0.23095 |
| **6** | **political.knowledge** | -0.49849 |
| **7** | **gender\_male** | 0.06753 |

Table 1.14 : feature importance lda model

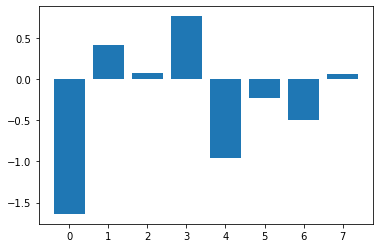


Fig 1.11: feature importanceof lda model

Classification report of model :

Classification Report of the training data:  
  
 precision recall f1-score support  
  
 0 0.75 0.69 0.72 322  
 1 0.87 0.90 0.88 745  
  
 accuracy 0.84 1067  
 macro avg 0.81 0.79 0.80 1067  
weighted avg 0.83 0.84 0.83 1067  
   
  
-------------------------------------------------------   
  
Classification Report of the test data:  
  
 precision recall f1-score support  
  
 0 0.75 0.66 0.70 140  
 1 0.86 0.90 0.88 318  
  
 accuracy 0.83 458  
 macro avg 0.80 0.78 0.79 458  
weighted avg 0.83 0.83 0.83 458

AUC and ROC of model:

AUC for the Training Data: 0.881  
AUC for the Test Data: 0.906

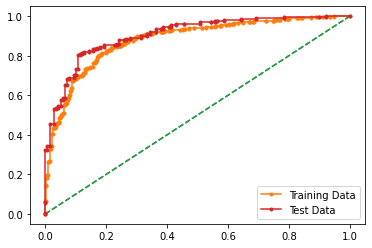


Fig 1.12: auc roc plot of lda model

Confusion matrix of model :

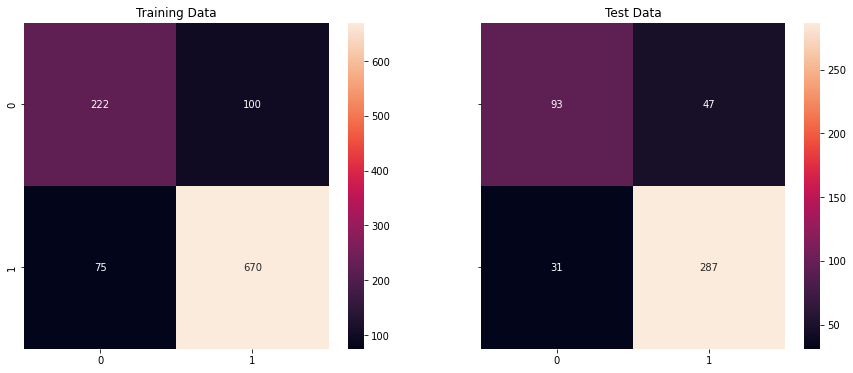


Fig 1.13 : confusion matrix of the lda model

Inferences :

* Model is neither underfitted nor overfitted.
* both conservative party and labour party recall value difference is less than 10%.
* Both conservative party and labour party precision value difference is null

-----------------------------------------------------------------------------------------------------------------------------

## **1.5) Apply KNN Model and Naïve Bayes Model (2pts). Interpret the inferences of each model (2 pts). Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)**

**(i) KNN model without tuning**

Default Parameters for KNN model :

'algorithm': 'auto',  
 'leaf\_size': 30,  
 'metric': 'minkowski',  
 'metric\_params': None,  
 'n\_jobs': None,  
 'n\_neighbors': 5,  
 'p': 2,  
 'weights': 'uniform'

Accuracy of the model :

score for train data: 0.852

score for test data: 0.840

Features importance :

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | 0.00261 |
| **1** | **economic.cond.national** | 0.01534 |
| **2** | **economic.cond.household** | 0.01377 |
| **3** | **Blair** | 0.04341 |
| **4** | **Hague** | 0.07449 |
| **5** | **Europe** | 0.07987 |
| **6** | **political.knowledge** | 0.04616 |
| **7** | **gender\_male** | -0.00393 |

Table 1.15 : feature importance knn model

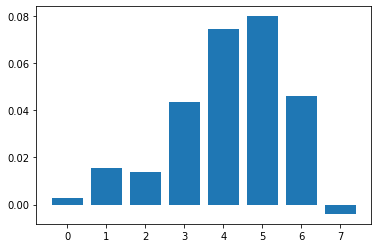


Fig 1.14: feature importance of knn model

Classification report of model:

train  
 precision recall f1-score support  
  
 0 0.78 0.71 0.74 322  
 1 0.88 0.91 0.90 745  
  
 accuracy 0.85 1067  
 macro avg 0.83 0.81 0.82 1067  
weighted avg 0.85 0.85 0.85 1067  
  
test  
 precision recall f1-score support  
  
 0 0.76 0.69 0.73 140  
 1 0.87 0.91 0.89 318  
  
 accuracy 0.84 458  
 macro avg 0.82 0.80 0.81 458  
weighted avg 0.84 0.84 0.84 458

AUC and ROC of model:

AUC for the Training Data: 0.928  
AUC for the Test Data: 0.865

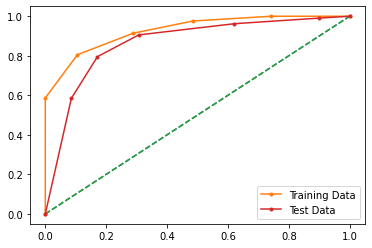


Fig 1.15 : aur roc curve of knn model

Confusion matrix of model :

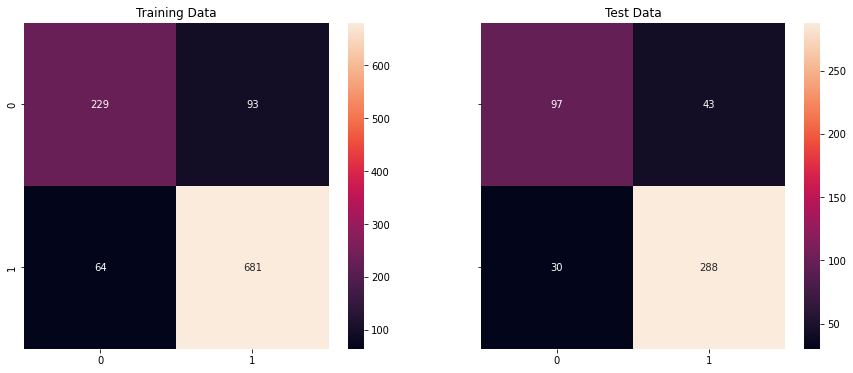


Fig 1.16 : confusion matrix of knn

Inferences :

* Model is neither overfit nor under fit.
* both conservative party and labour party recall value difference is less than 10%.
* Both conservative party and labour party precision value difference is null

-----------------------------------------------------------------------------------------------------------------------------

**(ii) Naïve Bayes model without tuning**

Default Parameters for LDA model :

'priors': None

'var\_smoothing': 1e-09

Accuracy of the model :

score for train data: 0.825  
score for test data: 0.842

Features importance :

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | 0.00616 |
| **1** | **economic.cond.national** | 0.01141 |
| **2** | **economic.cond.household** | 0.00157 |
| **3** | **Blair** | 0.04328 |
| **4** | **Hague** | 0.06807 |
| **5** | **Europe** | 0.04643 |
| **6** | **political.knowledge** | 0.01023 |
| **7** | **gender\_male** | 0.00026 |

Table 1.16 : feature importance naïve bayes model

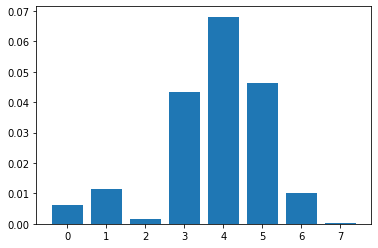


Fig 1.17: feature importance of naïve bayes model

Classification report of model :

train  
 precision recall f1-score support  
  
 0 0.71 0.70 0.71 322  
 1 0.87 0.88 0.88 745  
  
 accuracy 0.83 1067  
 macro avg 0.79 0.79 0.79 1067  
weighted avg 0.83 0.83 0.83 1067  
  
test  
 precision recall f1-score support  
  
 0 0.77 0.69 0.73 140  
 1 0.87 0.91 0.89 318  
  
 accuracy 0.84 458  
 macro avg 0.82 0.80 0.81 458  
weighted avg 0.84 0.84 0.84 458

AUC and ROC of model:

AUC for the Training Data: 0.878

AUC for the Test Data: 0.895

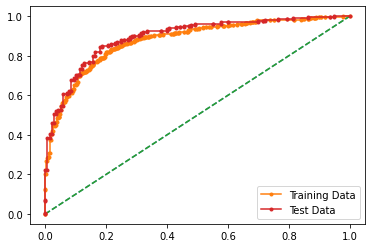


Fig 1.18 : aoc roc curve for naive bayes

Confusion matrix of model :

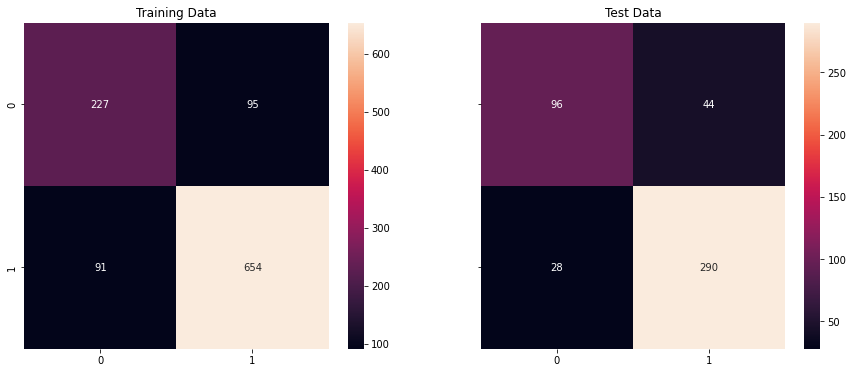


Fig 1.19 : confusion matric of naïve bayes

Inferences :

* Model is neither overfit nor underfit.
* both conservative party and labour party recall value difference is less than 10%.
* Both conservative party and labour party precision value difference is null

-----------------------------------------------------------------------------------------------------------------------------

## **1.6) Model Tuning (4 pts) , Bagging ( 1.5 pts) and Boosting (1.5 pts). Apply grid search on each model (include all models) and make models on best\_params. Define a logic behind choosing particular values for different hyper-parameters for grid search. Compare and comment on performances of all. Comment on feature importance if applicable. Successful implementation of both algorithms along with inferences and comments on the model performances.**

**(i) Bagging with Random Forest Classifier**

Default Parameters for Bagging with Random Forest Classifier :

'base\_estimator\_\_bootstrap': True,  
 'base\_estimator\_\_ccp\_alpha': 0.0,  
 'base\_estimator\_\_class\_weight': None,  
 'base\_estimator\_\_criterion': 'gini',  
 'base\_estimator\_\_max\_depth': 4,  
 'base\_estimator\_\_max\_features': 'auto',  
 'base\_estimator\_\_max\_leaf\_nodes': None,  
 'base\_estimator\_\_max\_samples': None,  
 'base\_estimator\_\_min\_impurity\_decrease': 0.0,  
 'base\_estimator\_\_min\_samples\_leaf': 1,  
 'base\_estimator\_\_min\_samples\_split': 2,  
 'base\_estimator\_\_min\_weight\_fraction\_leaf': 0.0,  
 'base\_estimator\_\_n\_estimators': 100,  
 'base\_estimator\_\_n\_jobs': None,  
 'base\_estimator\_\_oob\_score': False,  
 'base\_estimator\_\_random\_state': 1,  
 'base\_estimator\_\_verbose': 0,  
 'base\_estimator\_\_warm\_start': False,  
 'base\_estimator': RandomForestClassifier(max\_depth=4, random\_state=1),  
 'bootstrap': True,  
 'bootstrap\_features': False,  
 'max\_features': 1.0,  
 'max\_samples': 1.0,  
 'n\_estimators': 10,  
 'n\_jobs': None,  
 'oob\_score': False,  
 'random\_state': 1,  
 'verbose': 0,  
 'warm\_start': False

Accuracy of the model :

train: 0.845360824742268  
test: 0.8296943231441049

Features importance :

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | 0.01246 |
| **1** | **economic.cond.national** | 0.02400 |
| **2** | **economic.cond.household** | 0.00525 |
| **3** | **Blair** | 0.03738 |
| **4** | **Hague** | 0.06820 |
| **5** | **Europe** | 0.06977 |
| **6** | **political.knowledge** | 0.02007 |
| **7** | **gender\_male** | 0.00236 |

Table 1.17 : feature importance bagging model

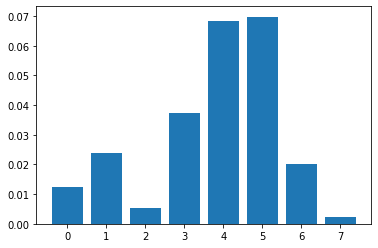


Fig 1.20: feature importance of bagging model

Classification report of model :

train  
 precision recall f1-score support  
  
 0 0.84 0.60 0.70 322  
 1 0.85 0.95 0.90 745  
  
 accuracy 0.85 1067  
 macro avg 0.84 0.78 0.80 1067  
weighted avg 0.84 0.85 0.84 1067  
  
test  
 precision recall f1-score support  
  
 0 0.84 0.54 0.66 140  
 1 0.83 0.96 0.89 318  
  
 accuracy 0.83 458  
 macro avg 0.84 0.75 0.77 458  
weighted avg 0.83 0.83 0.82 458

AUC and ROC of model:

AUC for the Training Data: 0.910

AUC for the Test Data: 0.909

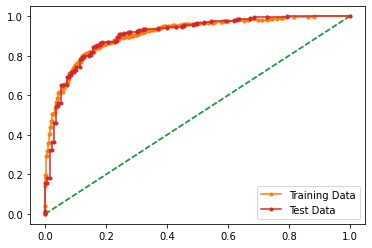


Fig 1.21: aoc roc curve boosting model

Inferences :

Model is overfit

-----------------------------------------------------------------------------------------------------------------------------

**(ii) ADA Boosting**

Default Parameters for Bagging with Random Forest Classifier:

'algorithm': 'SAMME.R',  
 'base\_estimator': None,  
 'learning\_rate': 1.0,  
 'n\_estimators': 50,  
 'random\_state': 1

Accuracy of the model :

train: 0.837863167760075  
test: 0.8406113537117904

Features importance :

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | 0.01403 |
| **1** | **economic.cond.national** | 0.01705 |
| **2** | **economic.cond.household** | 0.00210 |
| **3** | **Blair** | 0.04852 |
| **4** | **Hague** | 0.07030 |
| **5** | **Europe** | 0.03987 |
| **6** | **political.knowledge** | 0.00879 |
| **7** | **gender\_male** | 0.00000 |

Table 1.18 : feature importance ada boosting model

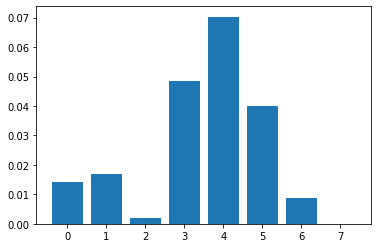


Fig 1.22: feature importance of boosting model

Classification report of model :

train  
 precision recall f1-score support  
  
 0 0.75 0.69 0.72 322  
 1 0.87 0.90 0.89 745  
  
 accuracy 0.84 1067  
 macro avg 0.81 0.80 0.80 1067  
weighted avg 0.83 0.84 0.84 1067  
  
test  
 precision recall f1-score support  
  
 0 0.77 0.69 0.72 140  
 1 0.87 0.91 0.89 318  
  
 accuracy 0.84 458  
 macro avg 0.82 0.80 0.81 458  
weighted avg 0.84 0.84 0.84 458

AUC and ROC of model:

AUC for the Training Data: 0.903

AUC for the Test Data: 0.901

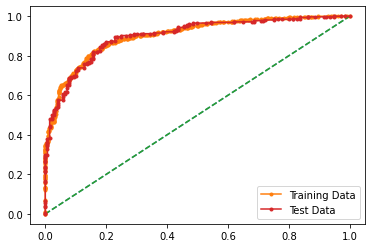


Fig 1.23: aoc roc curve boosting model

Inferences :

***Model is nor under fit nor overfit.***

**(iii) Gradient Boosting**

Default Parameters for Gradient Boosting :

'ccp\_alpha': 0.0,  
 'criterion': 'friedman\_mse',  
 'init': None,  
 'learning\_rate': 0.1,  
 'loss': 'deviance',  
 'max\_depth': 3,  
 'max\_features': None,  
 'max\_leaf\_nodes': None,  
 'min\_impurity\_decrease': 0.0,  
 'min\_samples\_leaf': 1,  
 'min\_samples\_split': 2,  
 'min\_weight\_fraction\_leaf': 0.0,  
 'n\_estimators': 100,  
 'n\_iter\_no\_change': None,  
 'random\_state': None,  
 'subsample': 1.0,  
 'tol': 0.0001,  
 'validation\_fraction': 0.1,  
 'verbose': 0,  
 'warm\_start': False

Accuracy of the model :

train: 0.8856607310215557  
test: 0.8537117903930131

Features importance :

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | 0.03856 |
| **1** | **economic.cond.national** | 0.02728 |
| **2** | **economic.cond.household** | 0.00551 |
| **3** | **Blair** | 0.05390 |
| **4** | **Hague** | 0.08433 |
| **5** | **Europe** | 0.07252 |
| **6** | **political.knowledge** | 0.03279 |
| **7** | **gender\_male** | 0.00210 |

Table 1.18 : feature importance gradient boosting

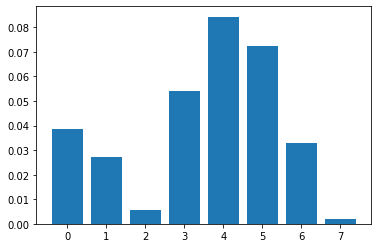


Fig 1.24: feature importance of gradient boosting model

Classification report of model :

train  
 precision recall f1-score support  
  
 0 0.83 0.78 0.80 322  
 1 0.91 0.93 0.92 745  
  
 accuracy 0.89 1067  
 macro avg 0.87 0.85 0.86 1067  
weighted avg 0.88 0.89 0.88 1067  
  
test  
 precision recall f1-score support  
  
 0 0.82 0.66 0.74 140  
 1 0.86 0.94 0.90 318  
  
 accuracy 0.85 458  
 macro avg 0.84 0.80 0.82 458  
weighted avg 0.85 0.85 0.85 458

AUC and ROC of model:

AUC for the Training Data: 0.949

AUC for the Test Data: 0.907

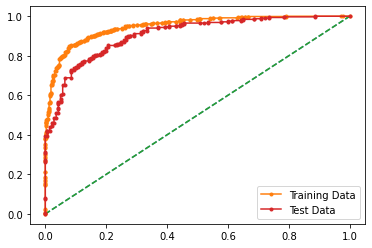


Fig 1.25: aoc roc curve boosting model

Inferences :

***Model is nor under fit nor overfit .***

-----------------------------------------------------------------------------------------------------------------------------

## **------------------------------tunning model-------------------------------------**

**(i) Logical Regression model with Hyper –Tune Parameter**

Hyper Tune Parameters for LDA model:

'C':[-1,1,10,0.001, 0.01, 0.1, 1, 10, 100, 1000],

'max\_iter':[100,200,300],

'penalty':['l1','l2','none'],

'solver':['saga','newton-cg'],

'tol':[0.0001,0.00002

Grid search parameter :

GridSearchCV(cv=5, estimator=LogisticRegression(max\_iter=10000, n\_jobs=2), n\_jobs=-1,  
 param\_grid={'C': [-1, 1, 10, 0.001, 0.01, 0.1, 1, 10, 100, 1000],  
 'max\_iter': [100, 200, 300],  
 'penalty': ['l1', 'l2', 'none'],  
 'solver': ['saga', 'newton-cg'],  
 'tol': [0.0001, 2e-05]},  
 scoring='accuracy')

Best Parameter :

‘C': 0.1

'max\_iter': 100

'penalty': 'l2'

'solver': 'newton-cg'

'tol': 0.0001

Best Score:

0.8331

Accuracy:

|  |  |
| --- | --- |
| Best LR model | LR model |
| Training Data: 0.8331771321462043  Test Data: 0.834061135371179 | score for train data: 0.8331771321462043 score for test data: 0.8384279475982532 |

Table1.19 accuracy LR

Features Importance:

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | -0.46391 |
| **1** | **economic.cond.national** | 0.40848 |
| **2** | **economic.cond.household** | 0.05538 |
| **3** | **Blair** | 0.04615 |
| **4** | **Hague** | -0.71602 |
| **5** | **Europe** | -0.22793 |
| **6** | **political.knowledge** | -0.38655 |
| **7** | **gender\_male** | 0.10026 |

Table 1.20 : feature importance of beat lr model

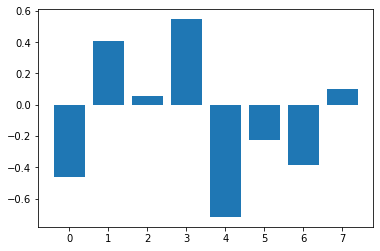


Fig 1.26 : feature importance of best lr model

Inference:

Accuracy of train and test has not much of a differrence.

----------------------------------------------------------------------------------------------------------------------------

**(ii) LDA model with Hyper –Tune Parameter**

Hyper Tune Parameters for LDA model:

'solver':['svd','eigen','lsqr'],

'shrinkage':[0.01,0.001,'none','auto'],

'n\_components':[1,2,3,4,5,6,7],

'tol':[0.0001,0.00001]

Grid search parameter:

GridSearchCV(cv=5, estimator=LinearDiscriminantAnalysis(), n\_jobs=-1,  
 param\_grid={'n\_components': [1, 2, 3, 4, 5, 6, 7],  
 'shrinkage': [0.01, 0.001, 'none', 'auto'],  
 'solver': ['svd', 'eigen', 'lsqr'],  
 'tol': [0.0001, 1e-05]},  
 scoring='f1')

Best Parameter:

'n\_components': 1

'shrinkage': 0.01

'solver': 'eigen'

'tol': 0.0001

Best Score :

0.8854657263013802

Accuracy:

|  |  |
| --- | --- |
| Best LDA model | LDA model |
| Training Data: 0.8359887535145267 Test Data: 0.834061135371179 | Training Data: 0.8359887535145267 Test Data: 0.8296943231441049 |

Table 1.21 : accuracy of lda

Features Importance:

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | -1.16073 |
| **1** | **economic.cond.national** | 0.44242 |
| **2** | **economic.cond.household** | 0.05407 |
| **3** | **Blair** | 0.69380 |
| **4** | **Hague** | -0.86297 |
| **5** | **Europe** | -0.23847 |
| **6** | **political.knowledge** | -0.50095 |
| **7** | **gender\_male** | 0.07363 |

Table 1.22 : feature importance of beat lda model

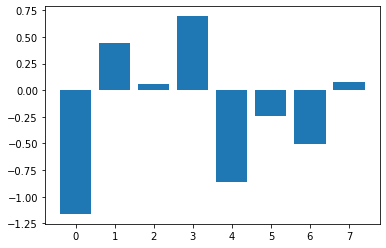


Fig 1.27 : feature importance of best lda model

Inference:

Both models of lda have approx. same accuracy.which shows that tuning model does not affectes the model much.

----------------------------------------------------------------------------------------------------------------------------

**(iii) KNN model with Hyper –Tune Parameter**

Hyper Tune Parameters for KNN model:

'n\_neighbors':[1,3,5,7,9,11,13,15,17,19,21],

'weights':['uniform','distance'],

'algorithm':['auto','ball\_tree', 'kd\_tree', 'brute'],

'leaf\_size':[30,60,90],

'metric':['euclidean','manhattan','minkowski']

Grid search parameter:

GridSearchCV(cv=3, estimator=KNeighborsClassifier(), n\_jobs=2,  
 param\_grid={'algorithm': ['auto', 'ball\_tree', 'kd\_tree', 'brute'],  
 'leaf\_size': [30, 60, 90],  
 'metric': ['euclidean', 'manhattan', 'minkowski'],  
 'n\_neighbors': [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21],  
 'weights': ['uniform', 'distance']},  
 scoring='f1')

Best Parameter:

'algorithm': 'ball\_tree'

'leaf\_size': 60,

'metric': 'manhattan',

'n\_neighbors': 17,

'weights': 'uniform'

Best Score :

0.8779886230664916

Accuracy:

|  |  |
| --- | --- |
| Best LDA model | LDA model |
| Training Data: 0.8537956888472352 Test Data: 0.8493449781659389 | train data: 0.8528584817244611 test data: 0.8406113537117904 |

Table 1.23 : accuracy of knn

Features Importance:

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | 0.01010 |
| **1** | **economic.cond.national** | 0.02374 |
| **2** | **economic.cond.household** | 0.00892 |
| **3** | **Blair** | 0.03843 |
| **4** | **Hague** | 0.06610 |
| **5** | **Europe** | 0.08066 |
| **6** | **political.knowledge** | 0.03698 |
| **7** | **gender\_male** | 0.00590 |

Table 1.24 : feature importance of best knn model

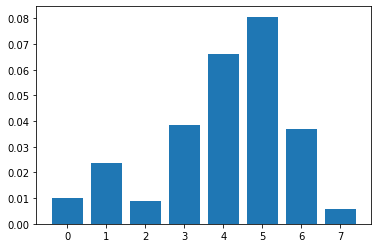


Fig 1.28 : feature importance of best knn model

Inference:

Model accuracy have not changed much after tuning .

Model is also not overfit or underfit.

----------------------------------------------------------------------------------------------------------------------------

**(iv) Naïve Bayes model with Hyper –Tune Parameter**

Hyper Tune Parameters for logical Regression model:

'var\_smoothing':[1e-01,1e-03,1e-05,1e-07,1e-09]

Grid search parameter :

GridSearchCV(cv=5, estimator=GaussianNB(), n\_jobs=2,  
 param\_grid={'var\_smoothing': [0.1, 0.001, 1e-05, 1e-07, 1e-09]},  
 scoring='accuracy')

Best Parameter :

{'var\_smoothing': 1e-05}

Best Score:

0.8209863542626475

Accuracy:

|  |  |
| --- | --- |
| Best NB model | NB model |
| Training Data: 0.8256794751640113 Test Data: 0.8427947598253275 | train data: 0.8256794751640113 test data: 0.8427947598253275 |

Table 1.25: accuracy of NB

Features Importance:

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | 0.00577 |
| **1** | **economic.cond.national** | 0.01587 |
| **2** | **economic.cond.household** | 0.00315 |
| **3** | **Blair** | 0.03659 |
| **4** | **Hague** | 0.06452 |
| **5** | **Europe** | 0.05207 |
| **6** | **political.knowledge** | 0.01062 |
| **7** | **gender\_male** | 0.00052 |

Table 1.26 : feature importance of best NB model

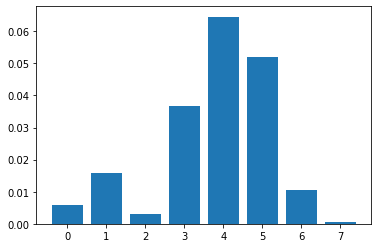


Fig 1.29 : feature importance of best NB model

Inference:

Tuned model is performing better then normal model.

----------------------------------------------------------------------------------------------------------------------------

**(v) Bagging Classifier model with Hyper –Tune Parameter**

Hyper Tune Parameters for logical Regression model:

'n\_estimators':[10,50,100],

'max\_samples':[50,100,1000],

'max\_features':[2,4,5,6,7,8,9],

'bootstrap':['True','False'],

'bootstrap\_features':['True','False'],

'oob\_score':['True','False']

Grid search parameter :

GridSearchCV(cv=3, estimator=BaggingClassifier(), n\_jobs=2,  
 param\_grid={'bootstrap': ['True', 'False'],  
 'bootstrap\_features': ['True', 'False'],  
 'max\_features': [2, 4, 5, 6, 7, 8, 9],  
 'max\_samples': [50, 100, 1000],  
 'n\_estimators': [10, 50, 100],  
 'oob\_score': ['True', 'False']},  
 scoring='f1')

Best Parameter :

'bootstrap': 'True',

'bootstrap\_features': 'False',

'max\_features': 7,

'max\_samples': 100,

'n\_estimators': 50,

'oob\_score': 'False'

Best score :

0.883571628443057

Accuracy:

|  |  |
| --- | --- |
| Best bagging model | Bagging model |
| Training Data: 0.8575445173383318  Test Data: 0.8275109170305677 | train: 0.845360824742268 test: 0.8296943231441049 |

Table 1.27: accuracy of Bagging

Features Importance:

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | 0.02308 |
| **1** | **economic.cond.national** | 0.03541 |
| **2** | **economic.cond.household** | 0.00944 |
| **3** | **Blair** | 0.02872 |
| **4** | **Hague** | 0.05718 |
| **5** | **Europe** | 0.07948 |
| **6** | **political.knowledge** | 0.03161 |
| **7** | **gender\_male** | 0.00616 |

Table 1.28 : feature importance of best NB model

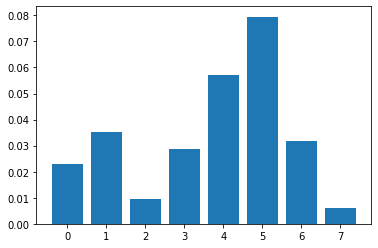


Fig 1.30 : feature importance of best Bagging model

Inference:

Bagging model is not good for our data set as compared to other models.

----------------------------------------------------------------------------------------------------------------------------

**(vi) ADA Boosting model with Hyper –Tune Parameter**

Hyper Tune Parameters for ADA Boosting model:

‘n\_estimators':[10,50,100,200],

'algorithm':['SAMME','SAMME.R']

Grid search parameter :

GridSearchCV(cv=3, estimator=AdaBoostClassifier(random\_state=1), n\_jobs=2,  
 param\_grid={'algorithm': ['SAMME', 'SAMME.R'],  
 'n\_estimators': [10, 50, 100, 200]},  
 scoring='f1')

Best Parameter :

'algorithm': 'SAMME', 'n\_estimators': 100

Best score :

0.8755085053883992

Accuracy:

|  |  |
| --- | --- |
| Best ADA Boosting model | ADA Boosting model |
| Training Data: 0.8284910965323337  Test Data: 0.851528384279476 | train: 0.837863167760075 test: 0.8406113537117904 |

Table 1.29: accuracy of best ada boosting

Features Importance:

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | 0.01075 |
| **1** | **economic.cond.national** | 0.01338 |
| **2** | **economic.cond.household** | -0.00170 |
| **3** | **Blair** | 0.05049 |
| **4** | **Hague** | 0.06492 |
| **5** | **Europe** | 0.04905 |
| **6** | **political.knowledge** | 0.00852 |
| **7** | **gender\_male** | 0.00000 |

Table 1.30 : feature importance of best ada boosting model

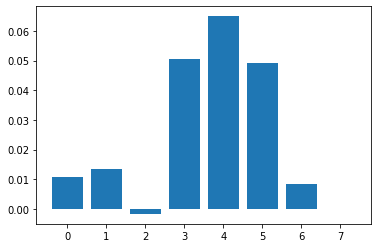


Fig 1.31: feature importance of best Bagging model

Inference:

Tuned model is more effectively fitted.

----------------------------------------------------------------------------------------------------------------------------

**(vii) Gradient Boosting model with Hyper –Tune Parameter**

Hyper Tune Parameters for Gradient Boosting model:

‘loss':['log\_loss','deviance','exponential'],

'n\_estimators':[10,50,100],

'min\_samples\_split':[2,10,100],

'min\_samples\_leaf':[1,10,100,1000],

'max\_depth':[3,4,5,6],

'tol':[0.0001,0.00001]

Grid search parameter :

GridSearchCV(cv=3, estimator=GradientBoostingClassifier(), n\_jobs=2,  
 param\_grid={'loss': ['log\_loss', 'deviance', 'exponential'],  
 'max\_depth': [3, 4, 5, 6],  
 'min\_samples\_leaf': [1, 10, 100, 1000],  
 'min\_samples\_split': [2, 10, 100],  
 'n\_estimators': [10, 50, 100],  
 'tol': [0.0001, 1e-05]},  
 scoring='f1')

Best Parameter :

'loss': 'exponential',

'max\_depth': 3,

'min\_samples\_leaf': 10,

'min\_samples\_split': 100,

'n\_estimators': 50, '

tol': 0.0001

Best score :

0.8825360297603358

Accuracy:

|  |  |
| --- | --- |
| Best ADA Boosting model | ADA Boosting model |
| Training Data: 0.8284910965323337  Test Data: 0.851528384279476 | train: 0.8856607310215557 test: 0.8537117903930131 |

Table 1.31: accuracy of best g boosting

Features Importance:

|  |  |  |
| --- | --- | --- |
|  | columns | Score |
| **0** | **age** | 0.00984 |
| **1** | **economic.cond.national** | 0.01010 |
| **2** | **economic.cond.household** | -0.00197 |
| **3** | **Blair** | 0.05456 |
| **4** | **Hague** | 0.06584 |
| **5** | **Europe** | 0.04577 |
| **6** | **political.knowledge** | 0.00905 |
| **7** | **gender\_male** | 0.00000 |

Table 1.32 : feature importance of best g boosting model

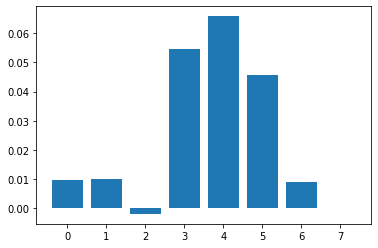


Fig 1.32: feature importance of best g Boosting model

Inference:

There is not much difference in the accuracy of tuned model and normal model.

----------------------------------------------------------------------------------------------------------------------------

## **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, classification report (4 pts) Final Model - Compare and comment on all models on the basis of the performance metrics in a structured tabular manner. Describe on which model is best/optimized, After comparison which model suits the best for the problem in hand on the basis of different measures. Comment on the final model.(3 pts)**

Performance Metrix:

1.Logistic Regeression:

Accuracy :

|  |  |  |
| --- | --- | --- |
| data | Best LR model | LR model |
| Train  test | 0.8331  0.8340 | 0.833  0.838 |

Table1.19 accuracy LR

Confusion Matrix :

Default model------------------------------------------------------------------------------------------------------------------------------------------

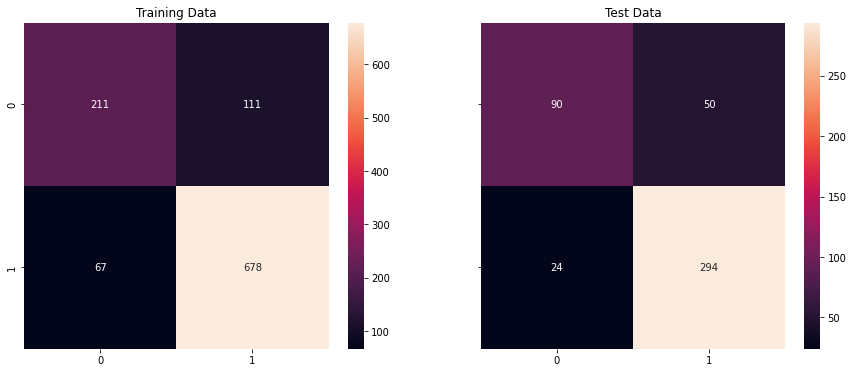
----

Fig 1.10 : confusion matrix of lrmodel

Tuned model--------------------------------------------------------------------------------------------------------------------------------------------

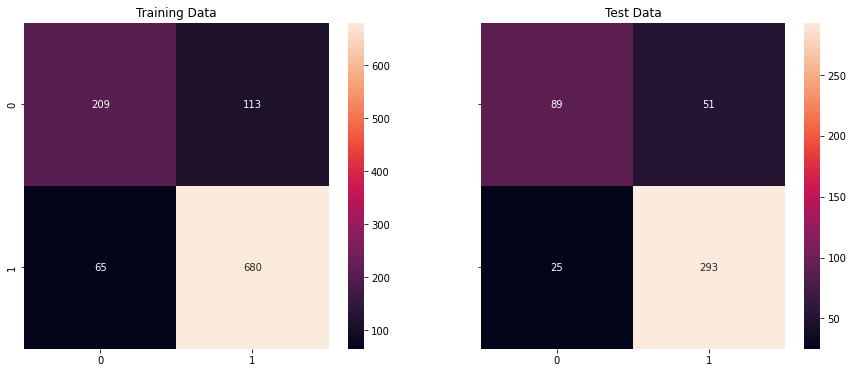
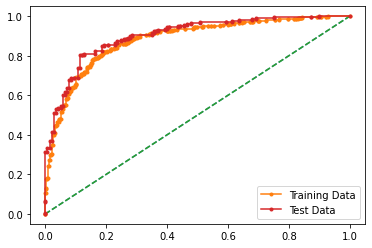


Fig 1.33: confusion matrix of blrmodel

ROC-AUC score and plot :

|  |  |  |
| --- | --- | --- |
| data | Default model | Tuned model |
| train | 0.882 | 0.882 |
| test | 0.902 | 0.904 |

Default model--------------------------------------------------------------------------------------------------------------------------------------------



Tuned model----------------------------------------------------------------------------------------------------------------------------------------------

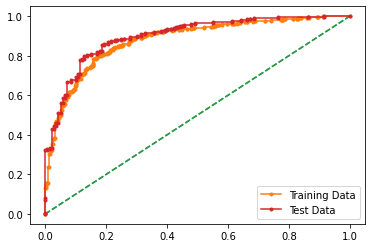


Fig 1.34 : roc and auc curve of best and default model

Classification report:

Default model--------------------------------------------------------------------------------------------------------------------------------------------

Classification Report of the training data:  
  
 precision recall f1-score support  
  
 0 0.76 0.66 0.70 322  
 1 0.86 0.91 0.88 745  
  
 accuracy 0.83 1067  
 macro avg 0.81 0.78 0.79 1067  
weighted avg 0.83 0.83 0.83 1067  
   
  
----------------------------------------------------------   
  
Classification Report of the test data:  
  
 precision recall f1-score support  
  
 0 0.79 0.64 0.71 140  
 1 0.85 0.92 0.89 318  
  
 accuracy 0.84 458  
 macro avg 0.82 0.78 0.80 458  
weighted avg 0.83 0.84 0.83 458

Tuned------------------------------------------------------------------------------------------------------------------------------------------------------

Classification Report of the training data:  
  
 precision recall f1-score support  
  
 0 0.76 0.65 0.70 322  
 1 0.86 0.91 0.88 745  
  
 accuracy 0.83 1067  
 macro avg 0.81 0.78 0.79 1067  
weighted avg 0.83 0.83 0.83 1067  
   
  
----------------------------------------------------------   
  
Classification Report of the test data:  
  
 precision recall f1-score support  
  
 0 0.78 0.64 0.70 140  
 1 0.85 0.92 0.89 318  
  
 accuracy 0.83 458  
 macro avg 0.82 0.78 0.79 458  
weighted avg 0.83 0.83 0.83 458  
   
--------------------------------------------------------------------------------------------

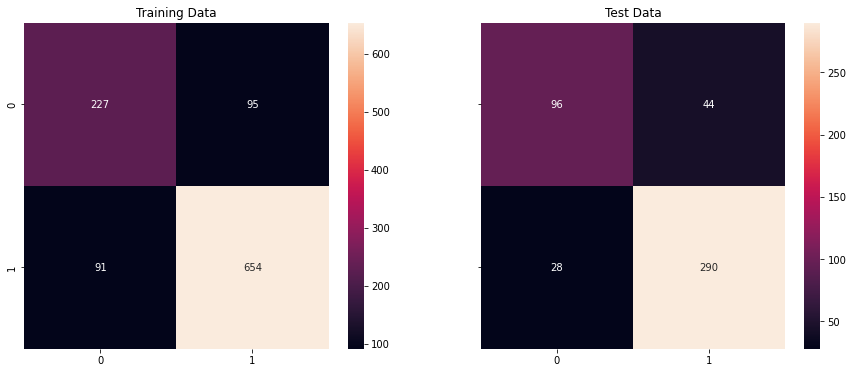
2.LDA:

Accuracy :

|  |  |  |
| --- | --- | --- |
| data | Best LDA model | LDA model |
| Train  test | 0.83  0.82 | 0.83  0.83 |

Confusion Matrix:

Default model------------------------------------------------------------------------------------------------------------------------------------------

----

Tuned model--------------------------------------------------------------------------------------------------------------------------------------------

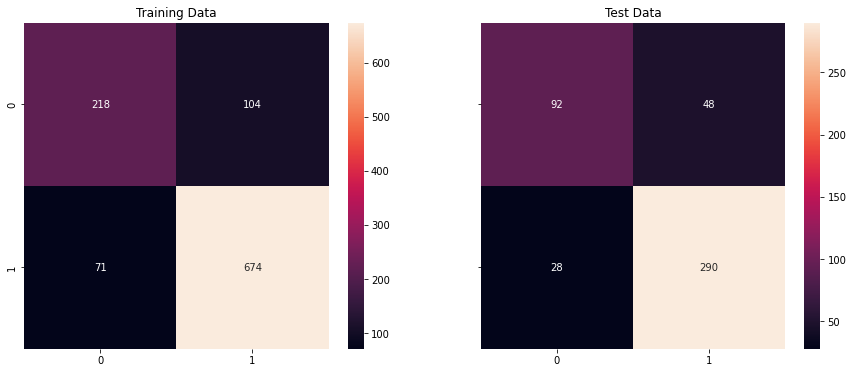
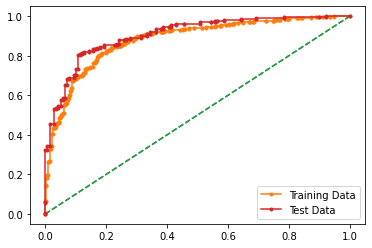


Fig 1.35: confusion matrix of blda model

ROC-AUC score and plot :

|  |  |  |
| --- | --- | --- |
| data | Default model | Tuned model |
| train | 0.881 | 0.882 |
| test | 0.906 | 0.904 |

Default model--------------------------------------------------------------------------------------------------------------------------------------------



Tuned model----------------------------------------------------------------------------------------------------------------------------------------------

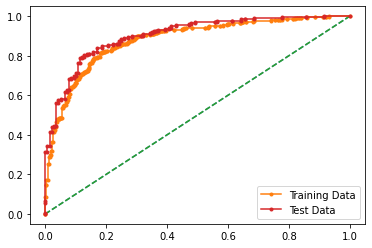


Fig 1.36 : roc and auc curve of best and default model

Classification report:

Default model--------------------------------------------------------------------------------------------------------------------------------------------

Classification Report of the training data:  
  
 precision recall f1-score support  
  
 0 0.75 0.69 0.72 322  
 1 0.87 0.90 0.88 745  
  
 accuracy 0.84 1067  
 macro avg 0.81 0.79 0.80 1067  
weighted avg 0.83 0.84 0.83 1067  
   
  
-------------------------------------------------------   
  
Classification Report of the test data:  
  
 precision recall f1-score support  
  
 0 0.75 0.66 0.70 140  
 1 0.86 0.90 0.88 318  
  
 accuracy 0.83 458  
 macro avg 0.80 0.78 0.79 458  
weighted avg 0.83 0.83 0.83 458  
Tuned------------------------------------------------------------------------------------------------------------------------------------------------------

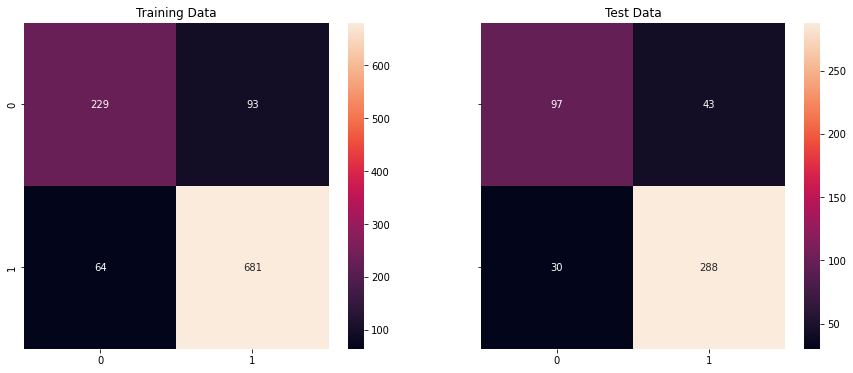
Classification Report of the training data:  
  
 precision recall f1-score support  
  
 0 0.75 0.68 0.71 322  
 1 0.87 0.90 0.89 745  
  
 accuracy 0.84 1067  
 macro avg 0.81 0.79 0.80 1067  
weighted avg 0.83 0.84 0.83 1067  
   
  
-------------------------------------------------------   
  
Classification Report of the test data:  
  
 precision recall f1-score support  
  
 0 0.77 0.66 0.71 140  
 1 0.86 0.91 0.88 318  
  
 accuracy 0.83 458  
 macro avg 0.81 0.78 0.80 458  
weighted avg 0.83 0.83 0.83 458  
 weighted avg 0.83 0.83 0.83 458  
   
--------------------------------------------------------------------------------------------  
3.KNN:

Accuracy :

|  |  |  |
| --- | --- | --- |
| data | KNN model | Best KNN model |
| Train  test | 0.852  0.840 | 0.853  0.849 |

Confusion Matrix:

Default model------------------------------------------------------------------------------------------------------------------------------------------

----

Tuned model--------------------------------------------------------------------------------------------------------------------------------------------

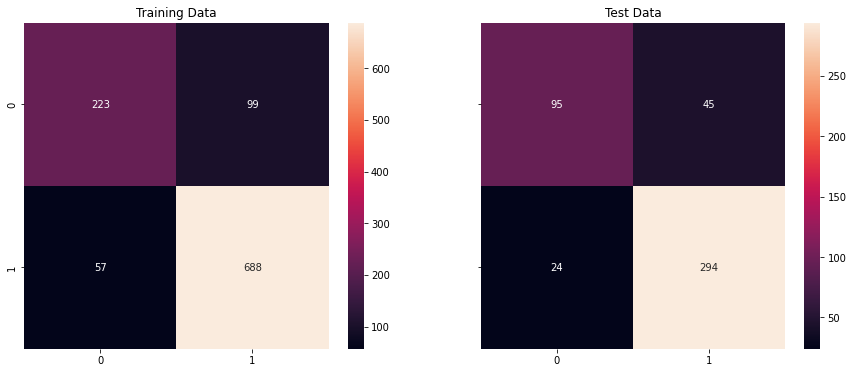
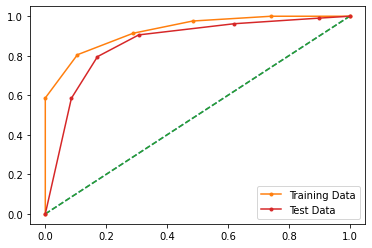


Fig 1.37: confusion matrix of knn model

ROC-AUC score and plot :

|  |  |  |
| --- | --- | --- |
| data | Default model | Tuned model |
| train | 0.928 | 0.909 |
| test | 0.865 | 0.894 |

Default model--------------------------------------------------------------------------------------------------------------------------------------------



Tuned model----------------------------------------------------------------------------------------------------------------------------------------------

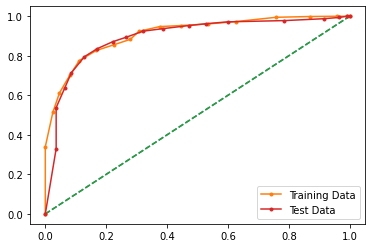


Fig 1.38 : roc and auc curve of best and default model

Classification report:

Default model--------------------------------------------------------------------------------------------------------------------------------------------

train  
 precision recall f1-score support  
  
 0 0.78 0.71 0.74 322  
 1 0.88 0.91 0.90 745  
  
 accuracy 0.85 1067  
 macro avg 0.83 0.81 0.82 1067  
weighted avg 0.85 0.85 0.85 1067  
  
test  
 precision recall f1-score support  
  
 0 0.76 0.69 0.73 140  
 1 0.87 0.91 0.89 318  
  
 accuracy 0.84 458  
 macro avg 0.82 0.80 0.81 458  
weighted avg 0.84 0.84 0.84 458

Tuned------------------------------------------------------------------------------------------------------------------------------------------------------

train  
 precision recall f1-score support  
  
 0 0.79 0.67 0.73 322  
 1 0.87 0.92 0.89 745  
  
 accuracy 0.85 1067  
 macro avg 0.83 0.80 0.81 1067  
weighted avg 0.84 0.85 0.84 1067  
  
test  
 precision recall f1-score support  
  
 0 0.80 0.68 0.73 140  
 1 0.87 0.92 0.89 318  
  
 accuracy 0.85 458  
 macro avg 0.83 0.80 0.81 458  
weighted avg 0.85 0.85 0.85 458

--------------------------------------------------------------------------------------------  
4.naïve bayes:

Accuracy :

|  |  |  |
| --- | --- | --- |
| data | model | Best model |
| Train  test | 0.825  0.842 | 0.825  0.842 |

Confusion Matrix:

Default model------------------------------------------------------------------------------------------------------------------------------------------

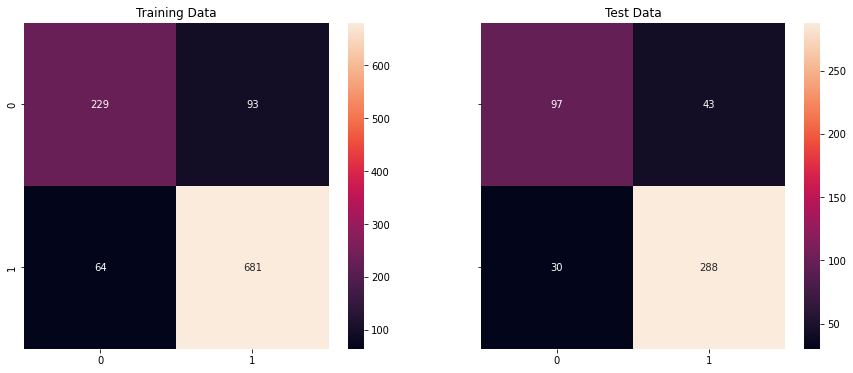
----

Fig 1.10 : confusion matrix of lrmodel

Tuned model--------------------------------------------------------------------------------------------------------------------------------------------

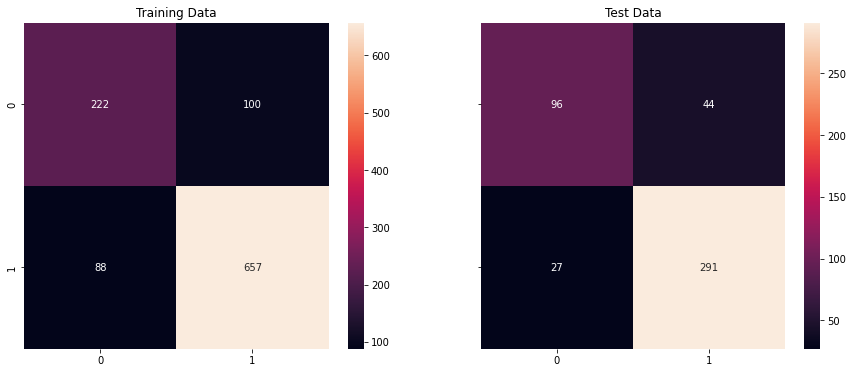
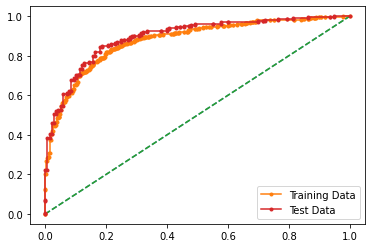


Fig 1.39: confusion matrix of best model

ROC-AUC score and plot :

|  |  |  |
| --- | --- | --- |
| data | Default model | Tuned model |
| train | 0.878 | 0.878 |
| test | 0.895 | 0.895 |

Default model--------------------------------------------------------------------------------------------------------------------------------------------



Tuned model----------------------------------------------------------------------------------------------------------------------------------------------

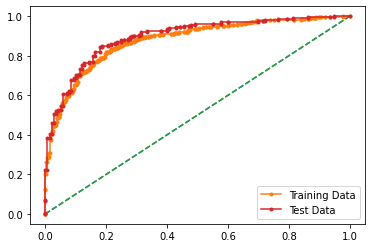


Fig 1.40: roc and auc curve of best and default model

Classification report:

Default model--------------------------------------------------------------------------------------------------------------------------------------------

train  
 precision recall f1-score support  
  
 0 0.71 0.70 0.71 322  
 1 0.87 0.88 0.88 745  
  
 accuracy 0.83 1067  
 macro avg 0.79 0.79 0.79 1067  
weighted avg 0.83 0.83 0.83 1067  
  
test  
 precision recall f1-score support  
  
 0 0.77 0.69 0.73 140  
 1 0.87 0.91 0.89 318  
  
 accuracy 0.84 458  
 macro avg 0.82 0.80 0.81 458  
weighted avg 0.84 0.84 0.84 458

Tuned------------------------------------------------------------------------------------------------------------------------------------------------------

train  
 precision recall f1-score support  
  
 0 0.71 0.70 0.71 322  
 1 0.87 0.88 0.88 745  
  
 accuracy 0.83 1067  
 macro avg 0.79 0.79 0.79 1067  
weighted avg 0.83 0.83 0.83 1067  
  
test  
 precision recall f1-score support  
  
 0 0.77 0.69 0.73 140  
 1 0.87 0.91 0.89 318  
  
 accuracy 0.84 458  
 macro avg 0.82 0.80 0.81 458  
weighted avg 0.84 0.84 0.84 458

--------------------------------------------------------------------------------------------

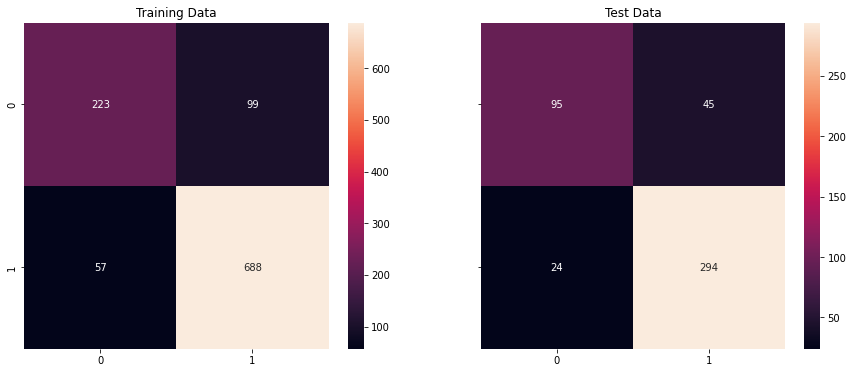
5.Gradient Boosting:

Accuracy :

|  |  |  |
| --- | --- | --- |
| data | model | Best model |
| Train  test | 0.88  0.85 | 0.82  0.85 |

Confusion Matrix:

Default model------------------------------------------------------------------------------------------------------------------------------------------



Tuned model--------------------------------------------------------------------------------------------------------------------------------------------

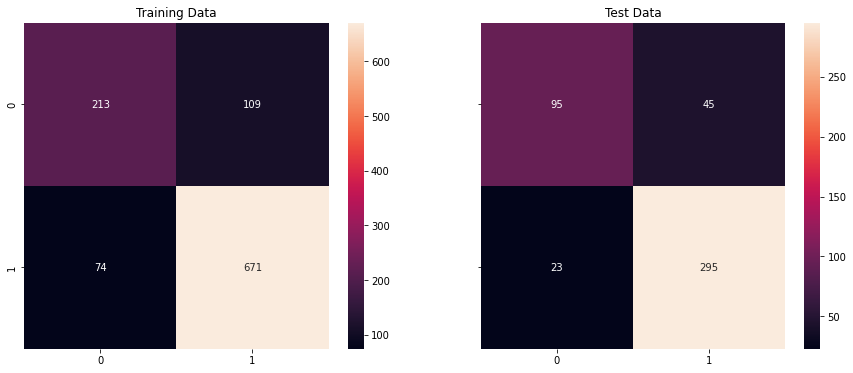
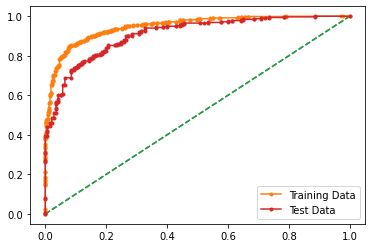


Fig 1.41: confusion matrix of best model

ROC-AUC score and plot :

|  |  |  |
| --- | --- | --- |
| data | Default model | Tuned model |
| train | 0.949 | 0.895 |
| test | 0.908 | 0.907 |

Default model--------------------------------------------------------------------------------------------------------------------------------------------



Tuned model----------------------------------------------------------------------------------------------------------------------------------------------

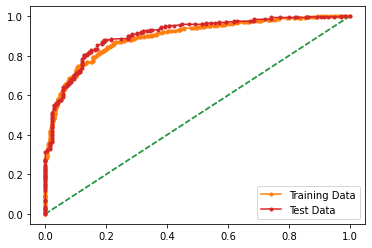


Fig 1.42 : roc and auc curve of best and default model

Classification report:

Default model--------------------------------------------------------------------------------------------------------------------------------------------

train  
 precision recall f1-score support  
  
 0 0.83 0.78 0.80 322  
 1 0.91 0.93 0.92 745  
  
 accuracy 0.89 1067  
 macro avg 0.87 0.85 0.86 1067  
weighted avg 0.88 0.89 0.88 1067  
  
test  
 precision recall f1-score support  
  
 0 0.82 0.67 0.74 140  
 1 0.87 0.94 0.90 318  
  
 accuracy 0.86 458  
 macro avg 0.85 0.80 0.82 458  
weighted avg 0.85 0.86 0.85 458  
Tuned------------------------------------------------------------------------------------------------------------------------------------------------------

Classification Report of the training data:  
  
 precision recall f1-score support  
  
 0 0.74 0.66 0.70 322  
 1 0.86 0.90 0.88 745  
  
 accuracy 0.83 1067  
 macro avg 0.80 0.78 0.79 1067  
weighted avg 0.82 0.83 0.83 1067  
   
  
----------------------------------------------------------   
  
Classification Report of the test data:  
  
 precision recall f1-score support  
  
 0 0.81 0.68 0.74 140  
 1 0.87 0.93 0.90 318  
  
 accuracy 0.85 458  
 macro avg 0.84 0.80 0.82 458  
weighted avg 0.85 0.85 0.85 458

--------------------------------------------------------------------------------------------

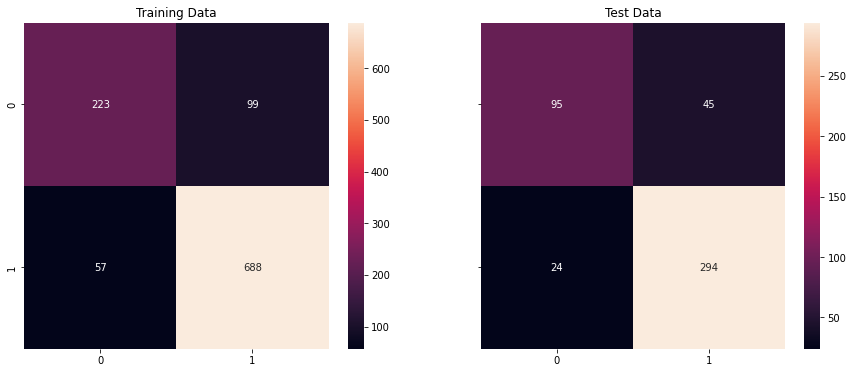
6.Bagging:

Accuracy :

|  |  |  |
| --- | --- | --- |
| data | model | Best model |
| Train  test | 0.84  0.82 | 0.84  0.81 |

Confusion Matrix:

Default model------------------------------------------------------------------------------------------------------------------------------------------

----

Tuned model--------------------------------------------------------------------------------------------------------------------------------------------

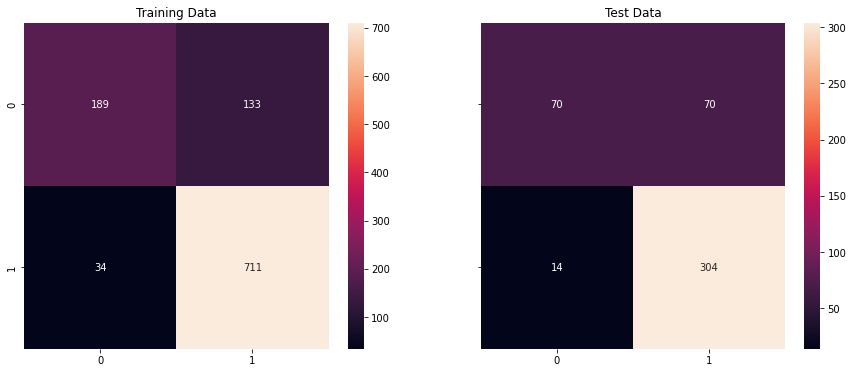
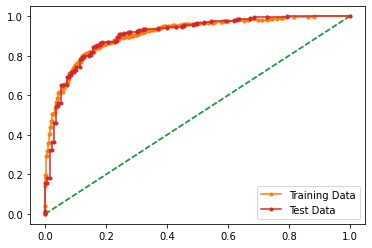


Fig 1.43: confusion matrix of bagging model

ROC-AUC score and plot :

|  |  |  |
| --- | --- | --- |
| data | Default model | Tuned model |
| train | 0.910 | 0.925 |
| test | 0.909 | 0.897 |

Default model--------------------------------------------------------------------------------------------------------------------------------------------



Tuned model----------------------------------------------------------------------------------------------------------------------------------------------

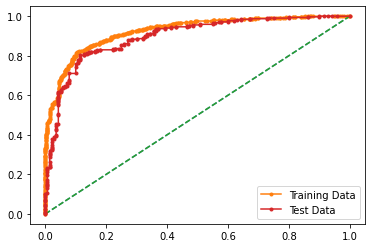


Fig 1.44 : roc and auc curve of best and default model

Classification report:

Default model--------------------------------------------------------------------------------------------------------------------------------------------

train  
 precision recall f1-score support  
  
 0 0.84 0.60 0.70 322  
 1 0.85 0.95 0.90 745  
  
 accuracy 0.85 1067  
 macro avg 0.84 0.78 0.80 1067  
weighted avg 0.84 0.85 0.84 1067  
  
test  
 precision recall f1-score support  
  
 0 0.84 0.54 0.66 140  
 1 0.83 0.96 0.89 318  
  
 accuracy 0.83 458  
 macro avg 0.84 0.75 0.77 458  
weighted avg 0.83 0.83 0.82 458

Tuned------------------------------------------------------------------------------------------------------------------------------------------------------

Classification Report of the training data:  
  
 precision recall f1-score support  
  
 0 0.85 0.59 0.69 322  
 1 0.84 0.95 0.89 745  
  
 accuracy 0.84 1067  
 macro avg 0.84 0.77 0.79 1067  
weighted avg 0.84 0.84 0.83 1067  
   
  
----------------------------------------------------------   
  
Classification Report of the test data:  
  
 precision recall f1-score support  
  
 0 0.83 0.50 0.62 140  
 1 0.81 0.96 0.88 318  
  
 accuracy 0.82 458  
 macro avg 0.82 0.73 0.75 458  
weighted avg 0.82 0.82 0.80 458

--------------------------------------------------------------------------------------------

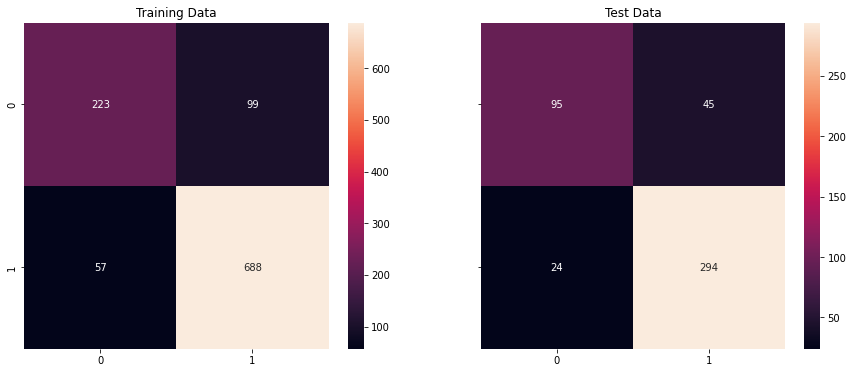
7.ADA Boosting:

Accuracy :

|  |  |  |
| --- | --- | --- |
| data | model | Best model |
| Train  test | 0.83  0.84 | 0.82  0.85 |

Confusion Matrix:

Default model------------------------------------------------------------------------------------------------------------------------------------------

----

Tuned model--------------------------------------------------------------------------------------------------------------------------------------------

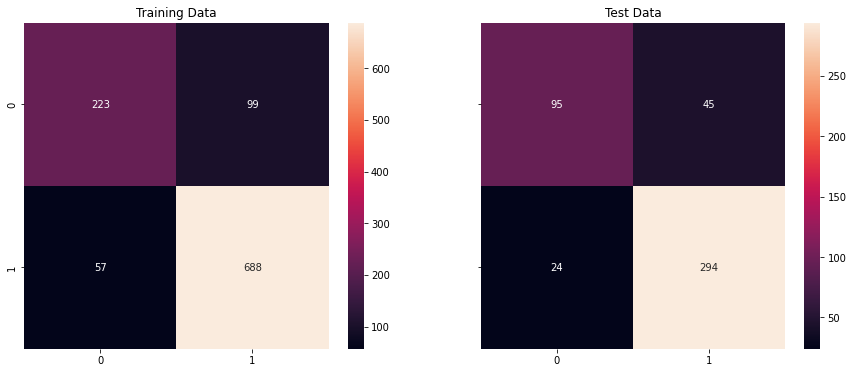
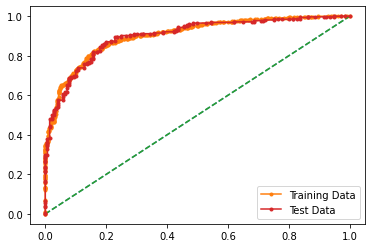


Fig 1.45: confusion matrix of best rmodel

ROC-AUC score and plot :

|  |  |  |
| --- | --- | --- |
| data | Default model | Tuned model |
| train | 0.903 | 0.895 |
| test | 0.901 | 0.907 |

Default model--------------------------------------------------------------------------------------------------------------------------------------------



Tuned model----------------------------------------------------------------------------------------------------------------------------------------------

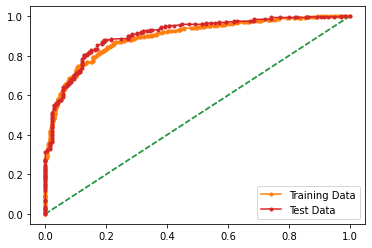


Fig 1.46 : roc and auc curve of best and default model

Classification report:

Default model--------------------------------------------------------------------------------------------------------------------------------------------

train  
 precision recall f1-score support  
  
 0 0.75 0.69 0.72 322  
 1 0.87 0.90 0.89 745  
  
 accuracy 0.84 1067  
 macro avg 0.81 0.80 0.80 1067  
weighted avg 0.83 0.84 0.84 1067  
  
test  
 precision recall f1-score support  
  
 0 0.77 0.69 0.72 140  
 1 0.87 0.91 0.89 318  
  
 accuracy 0.84 458  
 macro avg 0.82 0.80 0.81 458  
weighted avg 0.84 0.84 0.84 458

Tuned------------------------------------------------------------------------------------------------------------------------------------------------------

Classification Report of the training data:  
  
 precision recall f1-score support  
  
 0 0.74 0.66 0.70 322  
 1 0.86 0.90 0.88 745  
  
 accuracy 0.83 1067  
 macro avg 0.80 0.78 0.79 1067  
weighted avg 0.82 0.83 0.83 1067  
   
  
----------------------------------------------------------   
  
Classification Report of the test data:  
  
 precision recall f1-score support  
  
 0 0.81 0.68 0.74 140  
 1 0.87 0.93 0.90 318  
  
 accuracy 0.85 458  
 macro avg 0.84 0.80 0.82 458  
weighted avg 0.85 0.85 0.85 458

--------------------------------------------------------------------------------------------

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| model | Accuracy | | | |
| default | | Best model | |
| Train | test | train | test |
| 1 logical regression | 83 | 84 | 83 | 83 |
| 2.LDA | 83 | 82 | 83 | 83 |
| 3. KNN | 85 | 84 | 85 | 84 |
| 4.Naive bayes | 82 | 84 | 82 | 84 |
| 5.Bagging | 84 | 82 | 84 | 81 |
| 6. ADA boosting | 83 | 84 | 82 | 85 |
| 7.Gradient boosting | 88 | 85 | 82 | 85 |

Table 1.33 : Table of all models accuracy

Comment :

* Bagging and boosting models perform better without tuning.
* Roc and auc also perform better without tuning for bagging and boosting.

Final model :

If we have to choose a model among othermodel and boosting we will choose boosting (gardient model)with tuned as rediction as the accuracies achieved from the model is quite better .

## **1.8) Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective. There should be at least 3-4 Recommendations and insights in total. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks should only be allotted if the recommendations are correct and business specific.**

* Both economica.cond.national and economic.cond,national are averagely rated
* Labour is mostly voted by the voters
* There are more female voters so parties should do some women welfare like women empowerment, equality etc.
* Voters have medium political knowledge.
* Approx 30% of voters have no political knowledge
* Blair has higher number of voters than hague.
* Most models are not overfitted or not underfitted.

--------------------------------- PROBLEM 2 -------------------------------

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

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

Inaugural dataset has 59 number of speeches

### **2.1) Find the number of characters, words and sentences for the mentioned documents. (Hint: use .words(), .raw(), .sent() for extracting counts)**

Word, sentences, paragraph, characters counts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Speech | Sentences | Paragraphs | Words | Characters |
| Franklin D Roosevelt | 68 | 38 | 1536 | 6174 |
| John F Kennedy | 52 | 27 | 1546 | 6202 |
| Richard Nixon | 69 | 51 | 2028 | 8122 |

Table 2.1 : count table

### **2.2) Remove all the stopwords from the three speeches. Show the word count before and after the removal of stopwords. Show a sample sentence after the removal of stopwords.**

Word count after and befour removing stop words

|  |  |  |
| --- | --- | --- |
| Speech | Before removing stop words | after removing stop words |
| Franklin D Roosevelt | 1536 | 632 |
| John F Kennedy | 1546 | 697 |
| Richard Nixon | 2028 | 836 |

Table 2.2 : before, after word count

Sentences before and after removing stop words (example)

|  |  |
| --- | --- |
| Franklin D Roosevelt | |
| Before | On each national day of inauguration since 1789 , the people have renewed their sense of dedication to the United States . |
| after | national day inauguration since 1789 people renewed sense dedication united states |
| John F Kennedy | |
| Before | Vice President Johnson , Mr . Speaker , Mr . Chief Justice , President Eisenhower , Vice President Nixon , President Truman , reverend clergy , fellow citizens , we observe today not a victory of party , but a celebration of freedom -- symbolizing an end , as well as a beginning -- signifying renewal , as well as change . |
| after | vice president johnson mr speaker mr chief justice president eisenhower vice president nixon president truman reverend clergy fellow citizens observe today victory party celebration freedom symbolizing end well beginning signifying renewal well change |
| Richard Nixon | |
| Before | Mr . Vice President , Mr . Speaker , Mr . Chief Justice , Senator Cook , Mrs . Eisenhower , and my fellow citizens of this great and good country we share together : |
| after | mr vice president mr speaker mr chief justice senator cook mrs eisenhower fellow citizens great good country share together |

Table 2.3: sentences before and after removing stop word

### **2.3) Which word occurs the most number of times in his inaugural address for each president? Mention the top three words. (after removing the stopwords)**

most common words after and before removing stop words

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Speech | before | | | after | | |
| No. | word | count | No. | word | count |
| Roosevelt | 1  2  3 | The  Of  , | 104  81  77 | 1  2  3 | Nation  Know  Spirit | 12  10  9 |
| Kenney | 1  2  3 | ,  The  of | 85  83  65 | 1  2  3 | Let  Us  World | 16  12  8 |
| Nixon | 1  2  3 | ,  The  . | 96  80  68 | 1  2  3 | Us  Let  America | 26  22  21 |

Table 2.4 : most common words

### **2.4) Plot the word cloud of each of the three speeches. (after removing the stopwords)**

Word Cloud for 1941 Roosevelt Speech (after cleaning)!!



Fig 2.1: word cloud of Roosevelt speech

--------------------------------------------------------------------------------------------

Word Cloud for 1961 Kennedy Speech (after cleaning)!!

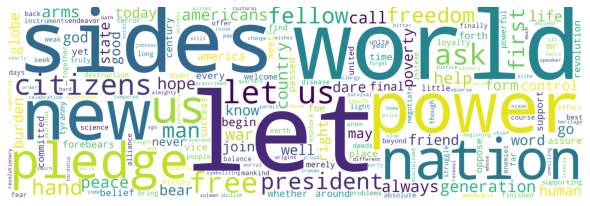


Fig 2.2 : word cloud of Kennedy speech

--------------------------------------------------------------------------------------------------------------------------------------------------------------

Word Cloud for 1973 Nixon Speech (after cleaning)!!

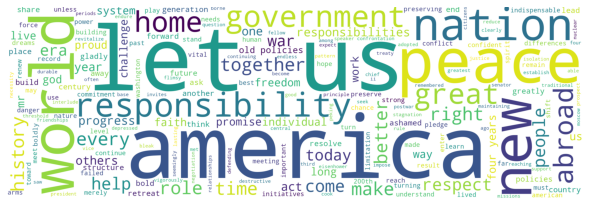


Fig 2.3 : word cloud of Nixon speech