**Project: Machine Learning**

Pavan Kalyan B | PGP DSBA Oct\_C’21 | May 2022

**Table of contents**

* 1. Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it.
  2. Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.
  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).
  4. Apply Logistic Regression and LDA (linear discriminant analysis).
  5. Apply KNN Model and Naïve Bayes Model. Interpret the results.
  6. Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting.
  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.
  8. Based on these predictions, what are the insights?

**Natural Language Processing:**

2.1 Find the number of characters, words, and sentences for the mentioned documents.

2.2 Remove all the stopwords from all three speeches.

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)

2.4 Plot the word cloud of each of the speeches of the variable. (after removing the stopwords)

Problem 1:

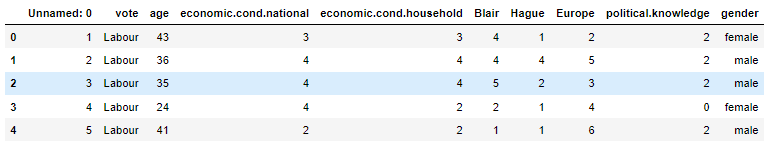
You are hired by one of the leading news channels CNBE who wants to analyze recent elections. This survey was conducted on 1525 voters with 9 variables. You have to build a model, to predict which party a voter will vote for on the basis of the given information, to create an exit poll that will help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

Dataset for Problem: [Election\_Data.xlsx](https://olympus.mygreatlearning.com/courses/63579/files/5239407/download?verifier=7wENWkaJuSzrrIvzR4Pt4rj9Zo27T0DtfZdBIztS&wrap=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.**

From the below image we can observe that there are several columns, we need understand each and every column.

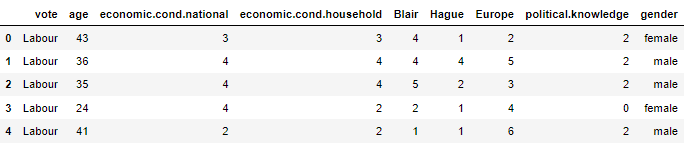
**Before removing Unnamed column:**



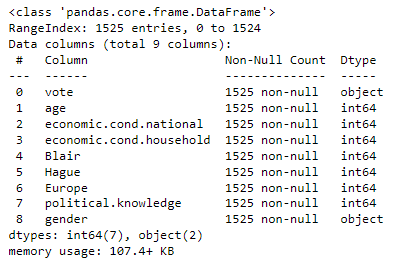
|  |
| --- |
| 1. vote: Party choice: Conservative or Labour |
| 2. age: in years |
| 3. economic.cond.national: Assessment of current national economic conditions, 1 to 5. |
| 4. economic.cond.household: Assessment of current household economic conditions, 1 to 5. |
| 5. Blair: Assessment of the Labour 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. |

From the above table we can observed that we don’t need the unnamed column as it doesn’t make any sense to the data. So we will drop it.

**After removing Unnamed column:**

****

This is the actual data without unnamed column with which we will proceed the future analysis.



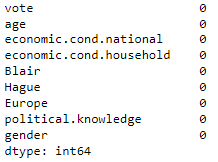
From the above figure we can observe that:

* vote and gender are object and other all integer columns
* There is no null data present inside
* We totally have 9 columns and 1525 rows
* The memory used is 107.4+ kb for this data

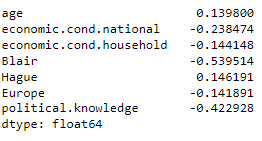


From the above table we can observe that:

* Age column is continuous and other are categorical
* For some algorithms only we will be doing scaling and rest we will use the original data
* The minimum value for age is 1 and maximum is 93 which is valid
* The minimum value for ‘economic.cond.national’, ‘economic.cond.household’, ‘Blair’, ‘Hague’ has 1 and maximum value as 5
* The std is higher in age only
* The minimum value for Europe is 1 and maximum value as 11, where as political.knowledge column has 0 as minimum and 3 as maximum



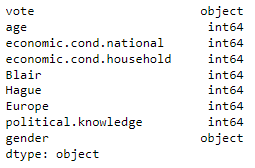
From the above table we don’t have any null values as shown.



From the above table we can observe that:

* Only for age column we have positive skewness
* Blair column which is moderately negative skewness
* Rest all columns are negatively skewed
  1. **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. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.**

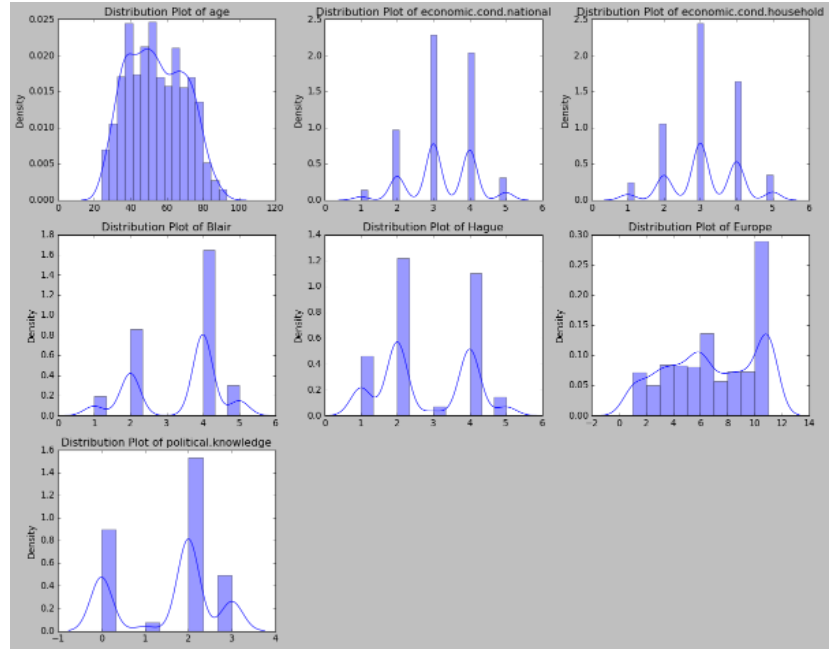
**Data type:**

****

From the above table we can observe that vote and gender attributes are object data type and other are integer data type.

**Univariate Analysis:**

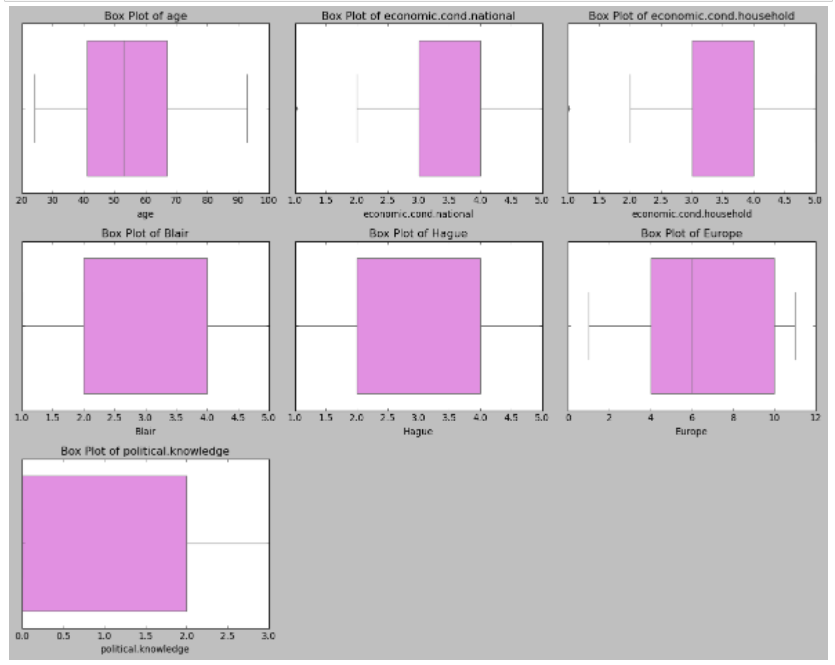
**Distribution Plots:**

****

From the above distribution plots, we can observe that:

* Age column only is normally distributed and symmetric
* Rest all other columns are not normally distributed as they are categorical in nature but numerical

**Box plots:**

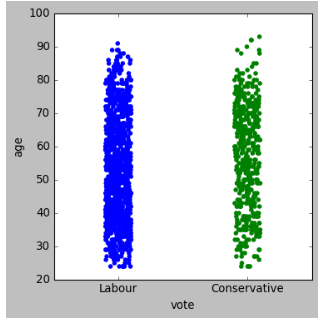


From the above figure we can observe that:

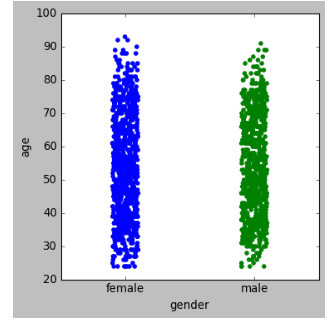
* Age is continuous and it has mean and median, where as Europe also has the same
* economic.cond.national and economic.cond.houshold columns has outliers in the data

**Bivariate Analysis:**

**Strip plot:**

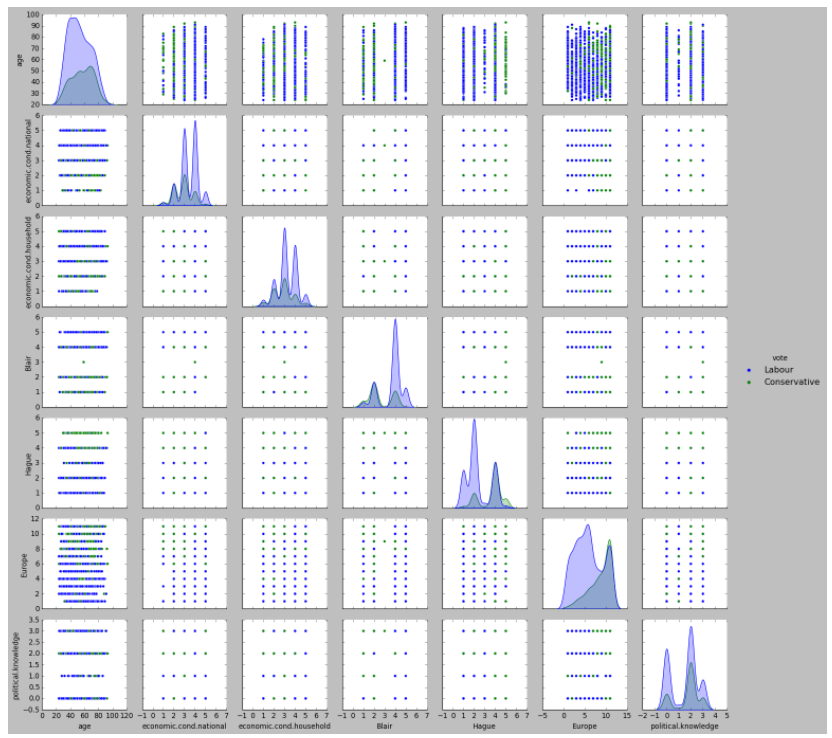
****

From the above figure, the plot is plotted between age and vote, we can see more number of votes ie., 70% from the labour where as 30% from the conservative.

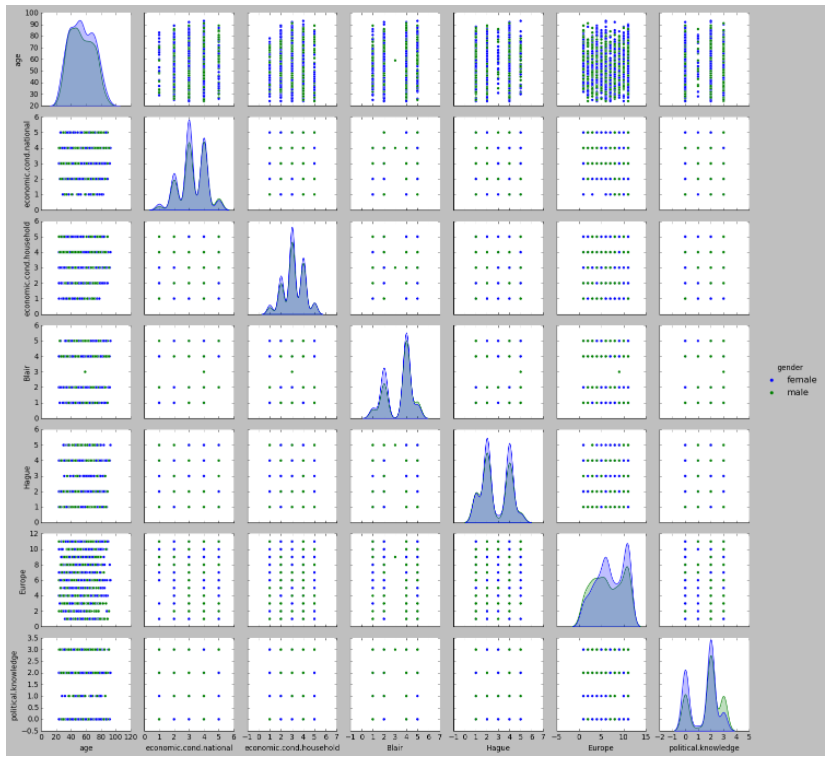


From the above figure, the plot is plotted between age and vote, we can see more number of votes ie., 55% from the female where as 45% from the male.

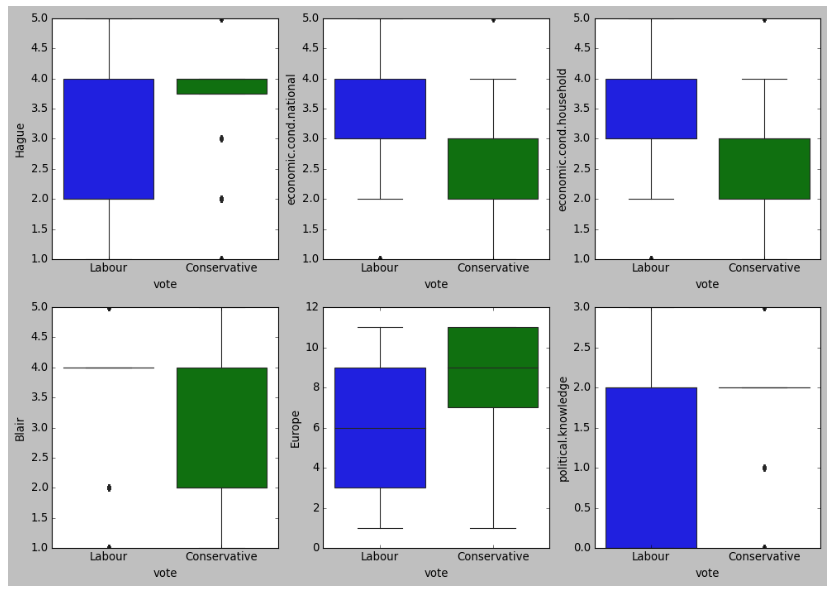
**Pair plot for bivariate taking hue as vote column:**



**Pair plot for bivariate taking hue as gender column:**

****

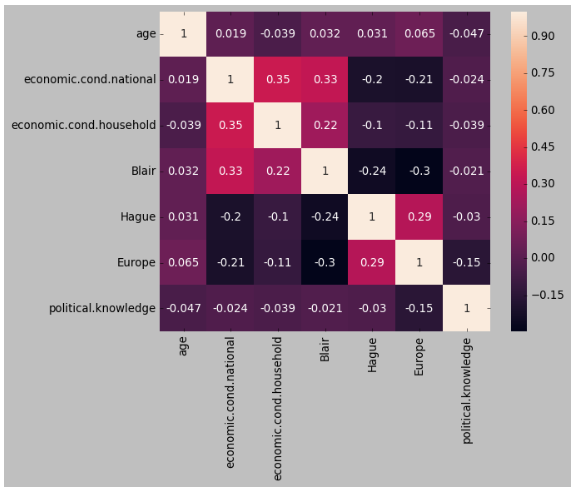
**Boxplot for bivariate:**

****

From the above figure we can observe that:

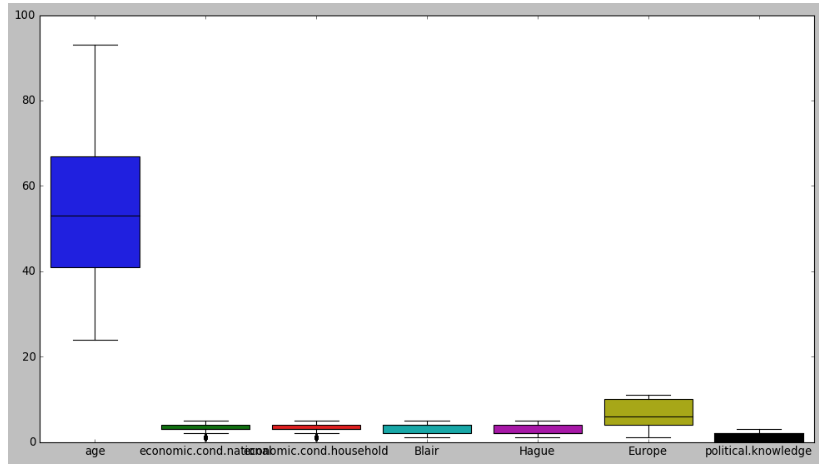
* vote and age column has some pattern which continuous and categorical plotted
* rest all not properly pattern generated as they are categorical vs categorical plots

**Heatmap:**

****

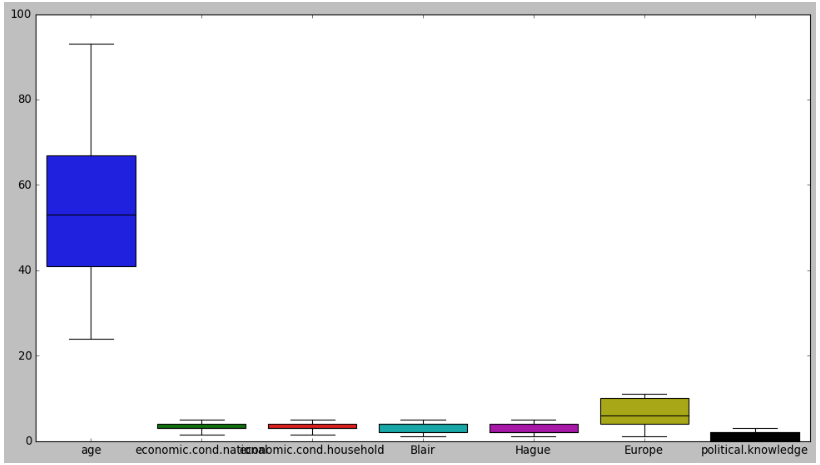
From the above figure we can observe that:

* Age and Europe has the moderate positive correlation
* Age with Blair, Hague and economic.cond.national has weak correlation
* Political.knowledge and age has negative correaltion
* Economic.cond.houshold with economic.cond.national, Blair has positive weak correlation

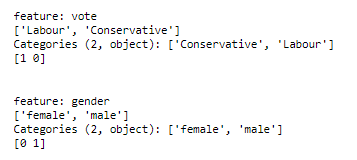
**Before outlier treatment:**

* For the columns economic.cond.national and economic.cond.household has outliers

**After outlier treatment:**

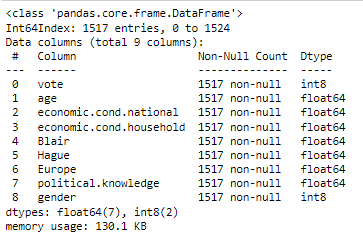
****

**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.**

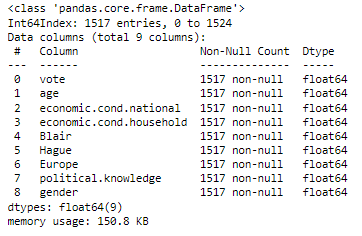


As we have converted the categorical into numerical,

* In vote column, 1 as labour, 0 as conservative
* In gender column, 0 as female and 1 as male

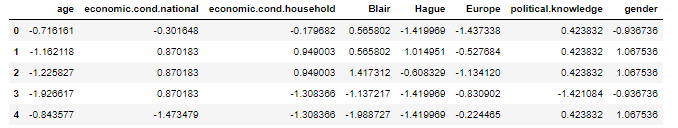


As we converted the object columns using categorical.codes(), the object data type is changed to int8



From the above table we converted the int8 to float64 using the astype function

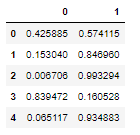
**After scaling:**

****

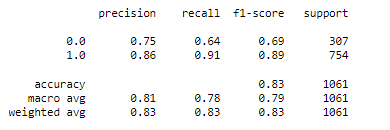
Scaling is need for some models like KNN, Bagging and Boosting. Rest all models we donot need any scaling things

**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)**

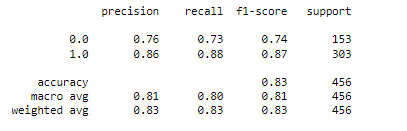
**Logistic Regression using grid search predict proba data:**

****

**Classification metrics on training data:**

****

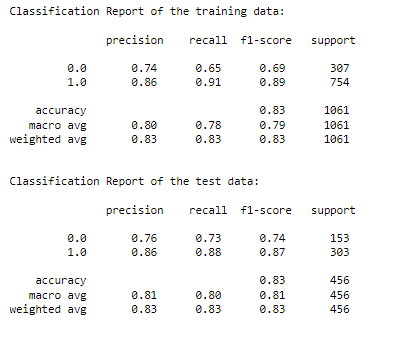
**Classification metrics on testing data:**

****

Based on the metrics, For the logistic model as it is stable which we cannot say it is overfitting or underfitting here as the both metrics are same for training and testing

**Linear Discriminant Analysis:**

**Classification metrics on training and testing:**

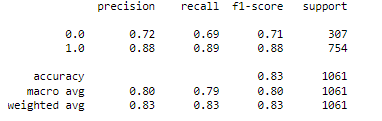
****

Based on the metrics, For LDA model as it is stable which we cannot say it is overfitting or underfitting here as the both metrics are same for training and testing

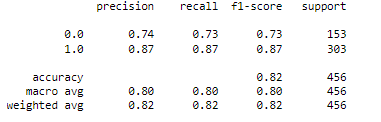
**1.5) Apply KNN Model and Naïve Bayes Model (2pts). Interpret the inferences of each model (2 pts). Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Calculate Train and Test Accuracies for each model. Comment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

**Naive Bayes model metrics:**

**Classification metrics on training:**

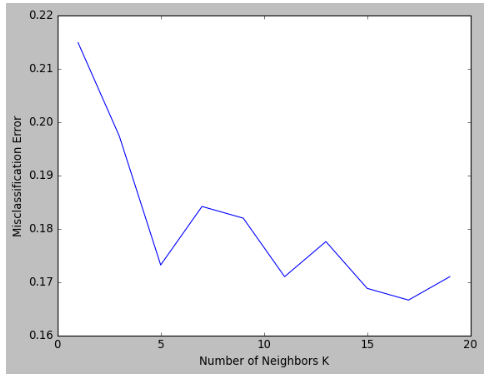
****

**Classification metrics on testing:**

****

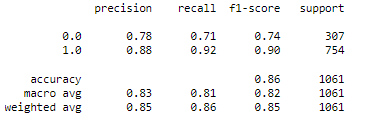
Based on the metrics, For Naïve Bayes model training accuracy is high and less in testing while performing but there is no huge difference to say it overfitting model

**KNN Model:**

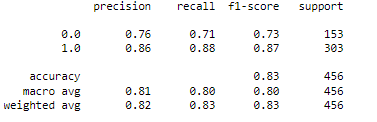
****

Using the misclassification plot, We take nneighbors best as 5, the value k=5 is the best one so we build model with k=5 and perform the classification

**Classification metrics on training:**



**Classification metrics on testing:**

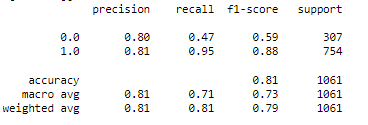


Based on the metrics, For KNN model training accuracy is high and less in testing. We call it as overfitting model but not highly.

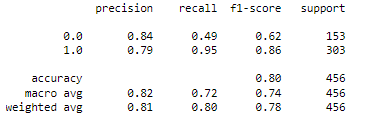
**1.6) Model Tuning (4 pts) , Bagging ( 1.5 pts) and Boosting (1.5 pts). Apply grid search on each model (include all models) and make models on best\_params. Define a logic behind choosing particular values for different hyper-parameters for grid search. Compare and comment on performances of all. Comment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

**Bagging with random forest classifier:**

**Classification metrics on training:**

****

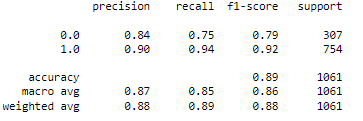
**Classification metrics on testing:**

****

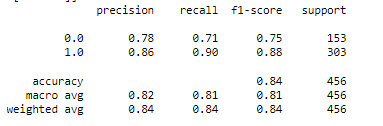
Based on the metrics, For Bagging model training accuracy is high and less in testing. We call it as overfitting model but not highly.

**Gradient Boosting using grid search:**

**Classification metrics on training:**

****

**Classification metrics on testing:**

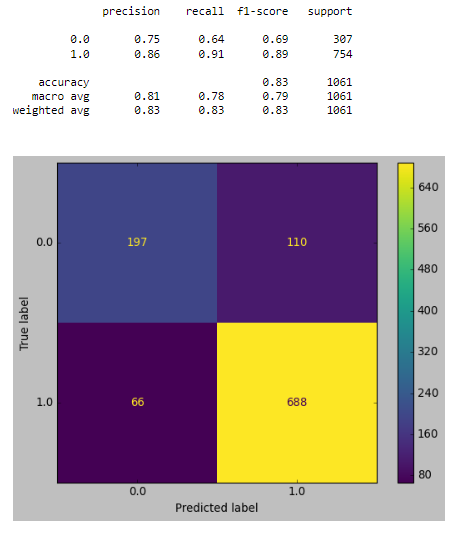
****

Based on the metrics, For Boosting model training accuracy is high and less in testing. We call it as overfitting model.

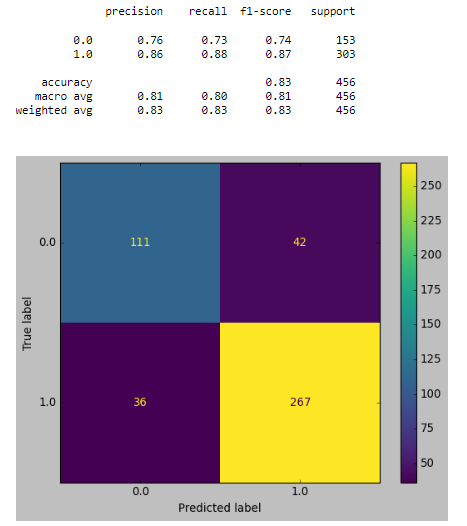
**1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model, classification report (4 pts) Final Model - Compare and comment on all models on the basis of the performance metrics in a structured 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)**

**Logistic Regression:**

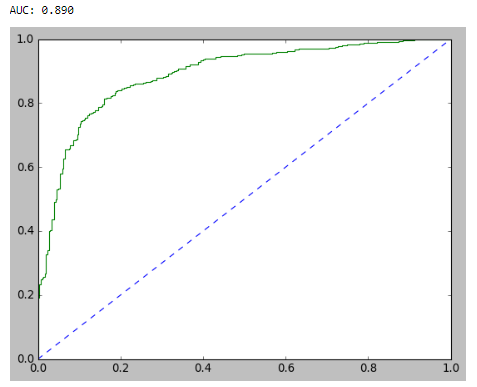
**Confusion matrix and classification matrix on training:**

****

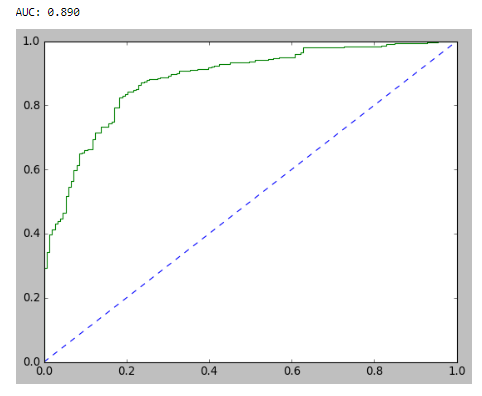
**Confusion matrix and classification matrix on testing:**

****

**ROC and AUC for training:**

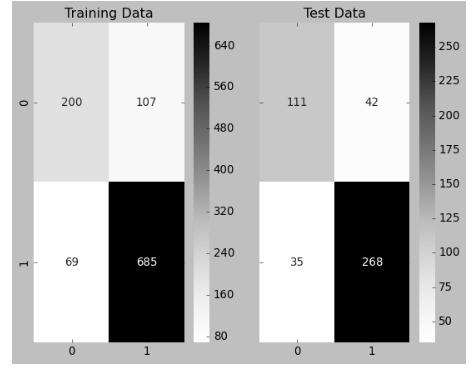
****

**ROC and AUC for testing:**

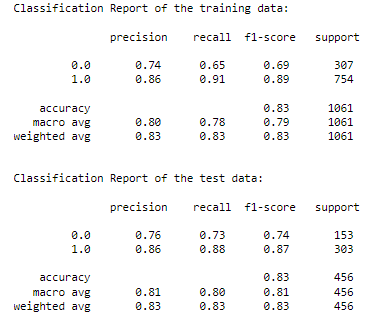
****

**Linear Discriminant Analysis:**

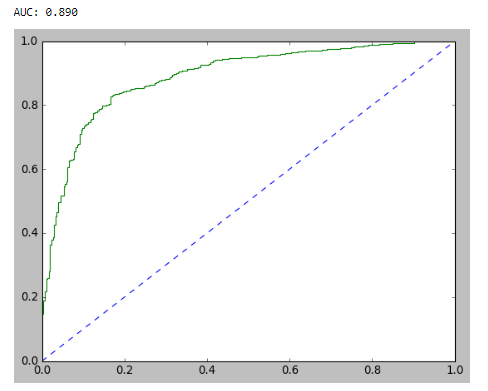
**Confusion matrix on training and testing:**

****

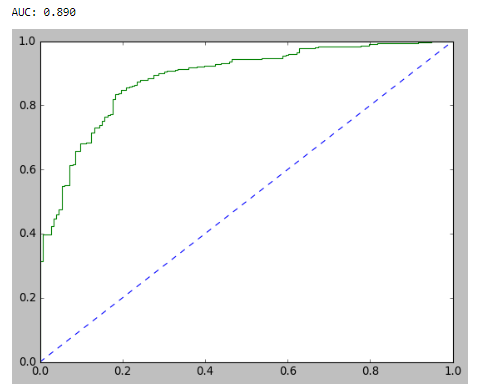
**Classification metrics on training and testing:**

****

**ROC and AUC for training:**

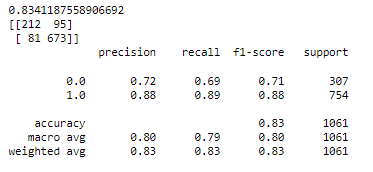
****

**ROC and AUC for testing:**

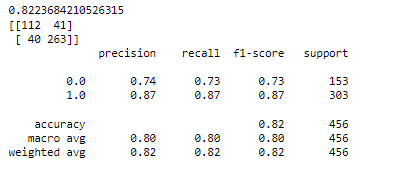
****

**Naive Bayes Model:**

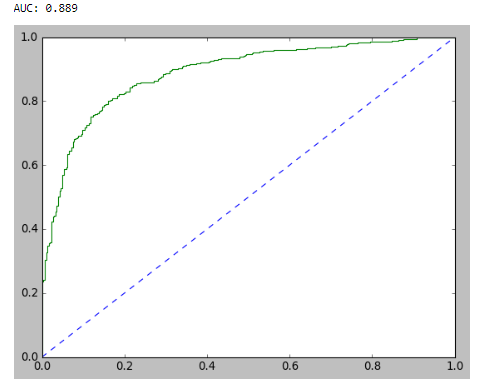
**Confusion matrix and Classification metrics on training:**

****

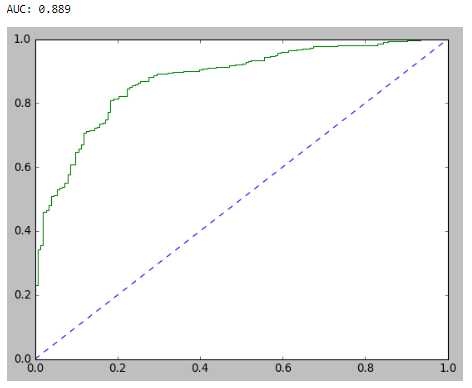
**Confusion matrix and Classification metrics on testing:**

****

**ROC and AUC on training:**

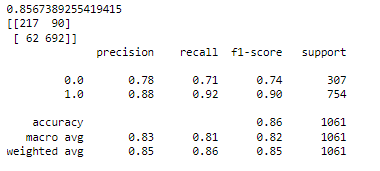
****

**ROC and AUC on testing:**

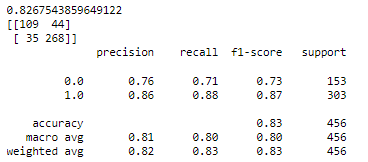
****

**KNN Model:**

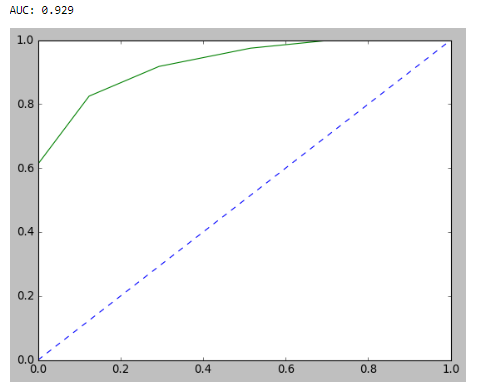
**Confusion matrix and Classification metrics on training:**

****

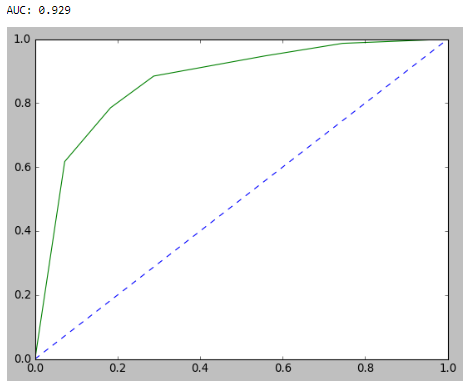
**Confusion matrix and Classification metrics on testing:**

****

**ROC and AUC on Training:**

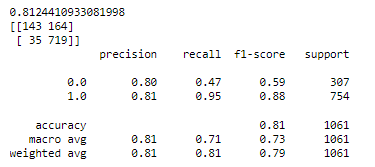
****

**ROC and AUC on testing:**

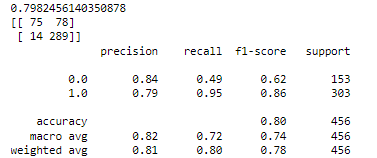
****

**Bagging model with random forest:**

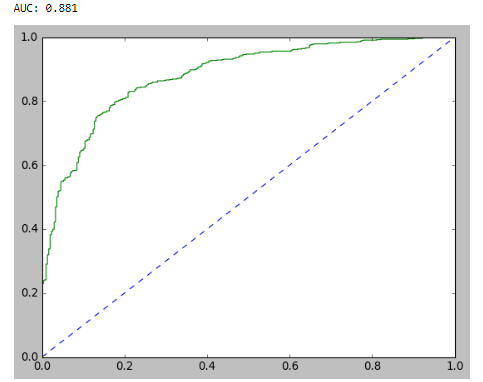
**Confusion matrix and Classification metrics on training:**

****

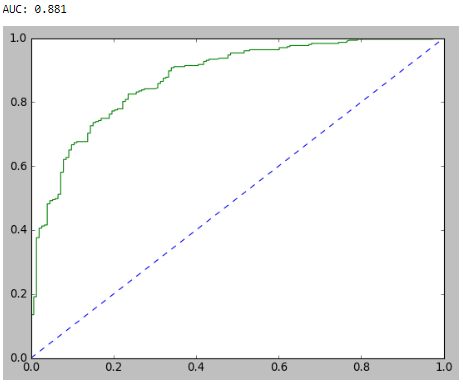
**Confusion matrix and Classification metrics on testing:**

****

**ROC and AUC on training:**

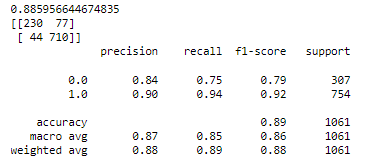
****

**ROC and AUC on testing:**

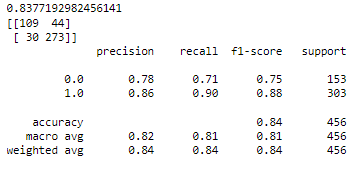
****

**Gradient Boosting Model:**

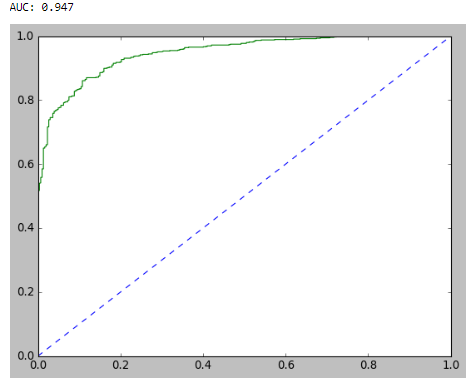
**Confusion matrix and Classification metrics on training:**

****

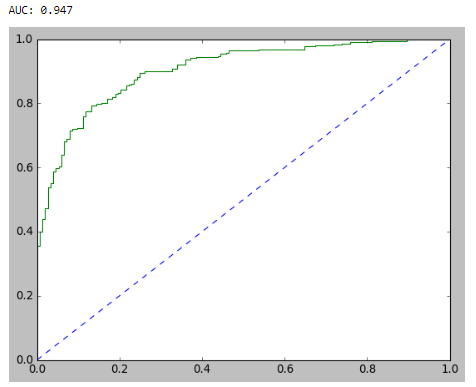
**Confusion matrix and Classification metrics on testing:**

****

**AUC and ROC on training:**

****

**AUC and ROC on testing:**

****

* Almost all the models performed well with accuracy between 80-90%
* Comparing all the models, we can observe that gradient boosting classifier works smart and faster than the other models.
* AUC of train and test is high respectively
* Precision for gradient boosting is 90% and 86%
* Recall for gradient boosting is 90% and 94%

**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.**

Based on the analysis we have to perform gradient boosting classifier model for the future prediction,

* Using this gradient boosting classifier, we can predict which party voter might vote with more sample voters.
* Exit polls can be created with this model to predict which party will win
* We need to collect more amount of data inorder to predict perfectly in future predictions
* As we donot have much amount of data our model made some misclassification
* To avoid this we will collect huge data to classify among the voters.

**======================================================================================**