**Business Report on**

***Machine Learning***

***Submitted to***



**Great Learning Olympus**

***Prepared by***

**Mrinmoy Majumdar**

**June-D Batch**

**Post Graduate Program in Data Science & Business Analytics**

**From**

****

UT Austin

**January, 2022**

TABLE OF CONTENT

**Problem 1…………………………………………………………….4**

**1.1** Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it………………………………………4

**1.2**Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for outliers………………………………………………………….........6

**1.3**Encode the data (having string values) for Modelling. Is Scaling necessary here or not? Data Split: Split the data into train and test (70:30)……………...14

###### **1.4**Apply Logistic Regression and LDA (linear discriminant analysis)………14

###### 1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results………...15

1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging) and Boosting…………………………………………………………………...16

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………………………………………20

1.8 Based on these predictions, what are the insights? ......................................31

**Problem 2……………………………………………………………32**

2.1 Find the number of characters, words, and sentences for the mentioned documents...........................................................................................................32

2.2 Remove all the stop words from all three speeches......................................32

2.3 Which word occurs the most number of times in his inaugural address for each president? Mention the top three words………………………………….33

2.4 Plot the word cloud of each of the speeches of the variable (after removing the stopwords)....................................................................................................33

**LIST OF FIGURES**

|  |  |
| --- | --- |
| **Images** | **Page No** |
| **Info and null value checking** | **5** |
| **Skewness** | **6** |
| **Histogram and box plot** | **7-8** |
| **Bar plot** | **8-9** |
| **Strip plot** | **9-11** |
| **Correlation plot** | **11** |
| **Pair plot** | **12** |
| **Coefficient of determination** | **15** |
| **MCE plot** | **16** |
| **Feature importance** | **17** |
| **Decision tree** | **18** |
| **ROC curve** | **20** |
| **Classification report** | **21** |
| **Confusion matrix** | **21** |
| **Word cloud** | **33-34** |

**LIST OF TABLES**

|  |  |
| --- | --- |
| Table 1 | 4 |
| Table 2 | 4 |
| Table 3 | 5 |
| Table 4 | 6 |
| Table 5 | 6 |
| Table 6 | 14 |
| Table 7 | 14 |
| Table 8 | 17 |
| Table 9 | 31 |
| Table 10 | 32 |

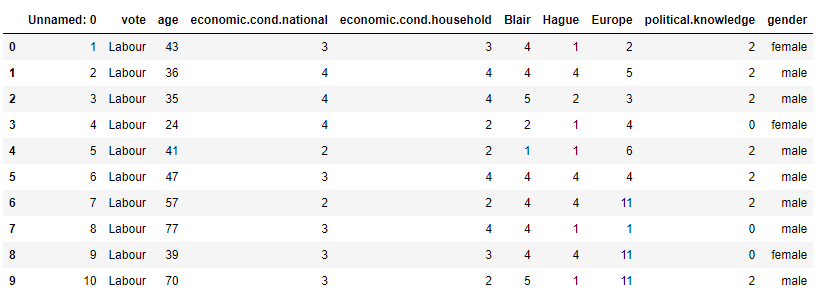
**Problem 1**

**Problem Statement:**

You are hired by one of the leading news channels CNBE who wants to analyse 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.

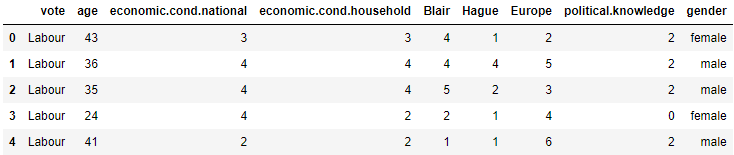
**1.1 Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it.**

Displaying election data:



**Table 1: Top 10 rows of election data Frame**

Unnamed: 0 column is not required as part of model building, hence we will drop the column from the dataset.

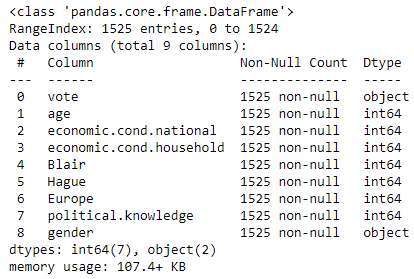


**Table 2: Data Frame without Unnamed: 0 variable**

## Basic EDA:

* Checking shape and information of data Frame

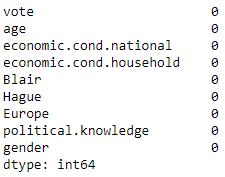
(1525, 9) – The data set contains 1525 observations of data and 9 variables. Previously there were 10 variables which included dropped column Unnamed: 0.



**Image 1: Information on election dataset**

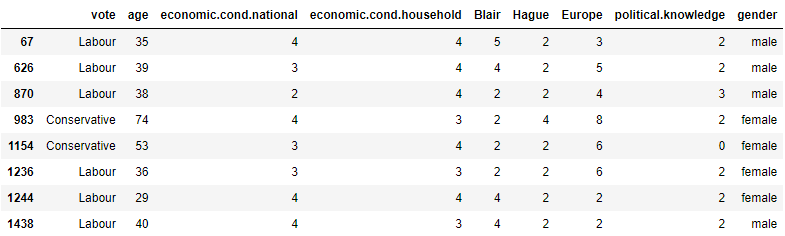
### Majority of the data has 1525 instances with 9 attributes – 7 variables of integer type, and 2 variables of object type.

* Check the presence of missing values



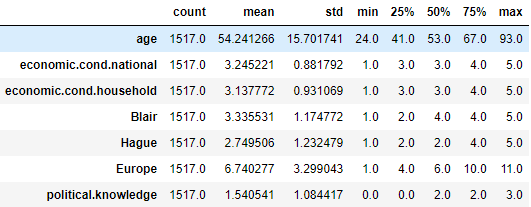
**Image 2: Checking null values in data**

### There are no null values in the dataset. There are 8 duplicate values in the dataset which was treated by dropping duplicates from the dataset. The shape of the dataset post dropping duplicates became (1517, 9).

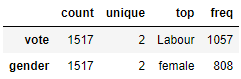


**Table 3: Duplicate records in dataset**

* Checking summary of data Frame

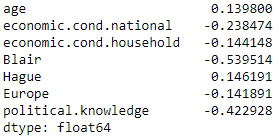


**Table 4: Description of numeric records**



**Table 5: Description of categorical records**

Looking at the 5 point summary, we can probably conclude that data is normally distributed as the mean and median of the columns are almost identical. The claims can be further solidified with the help of univariate, bivariate and multivariate analysis of feature columns along with its associated skewness.

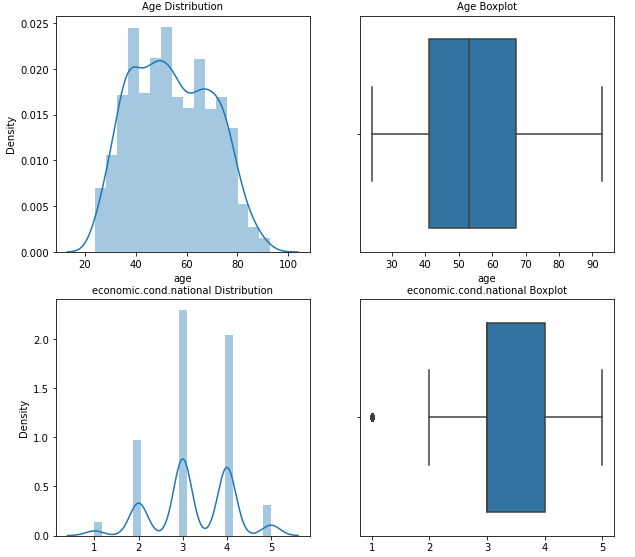


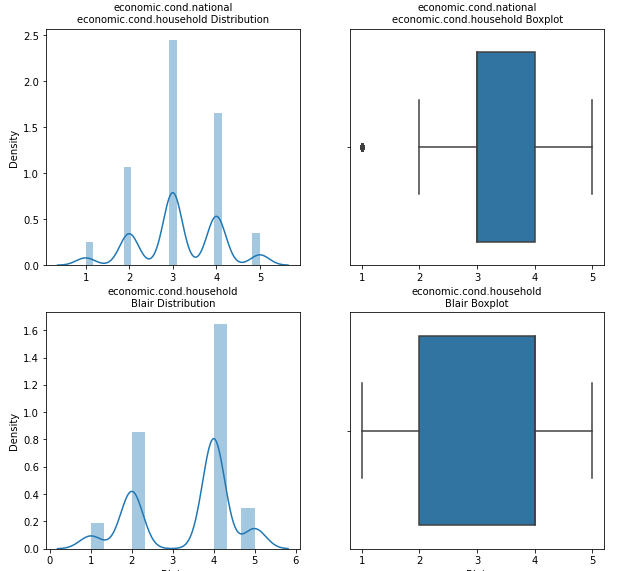
**Image 3: Skewness of dataset**

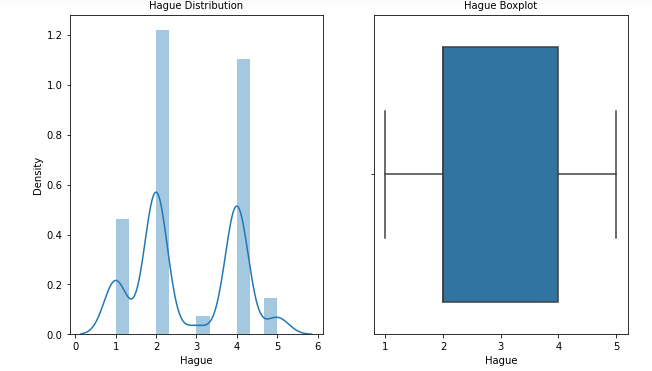
Age and Hague have positive skewness whereas the remaining columns have negative skewness.

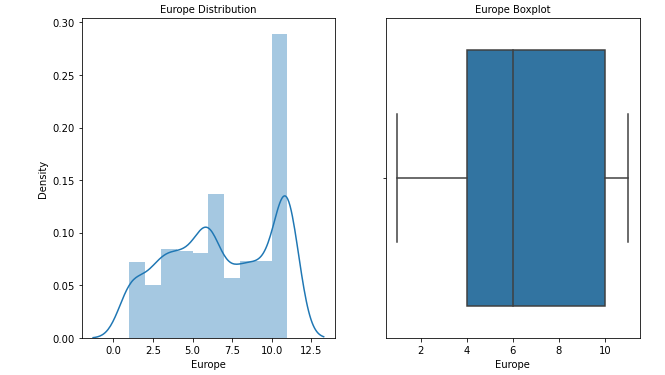
**1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.**

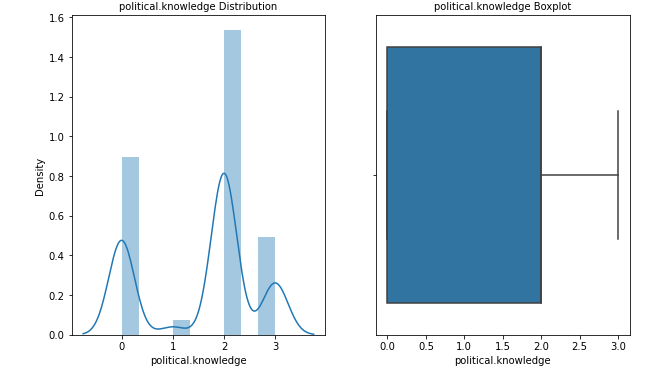
**Univariate Analysis**







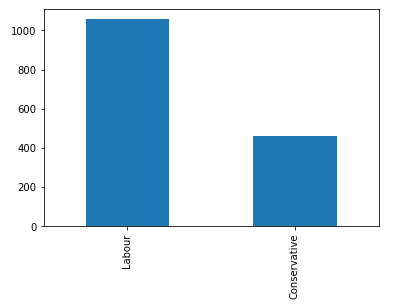




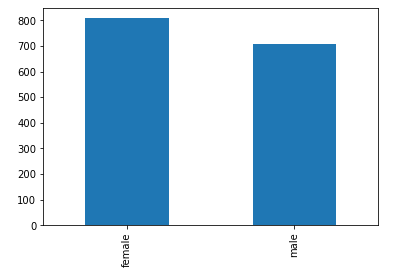
**Image 4: Histogram and boxplot of numerical columns**

Insights

There are outliers in the dataset as evident from the boxplot which need to be treated.



**Image 5: Bar plot showing vote column**

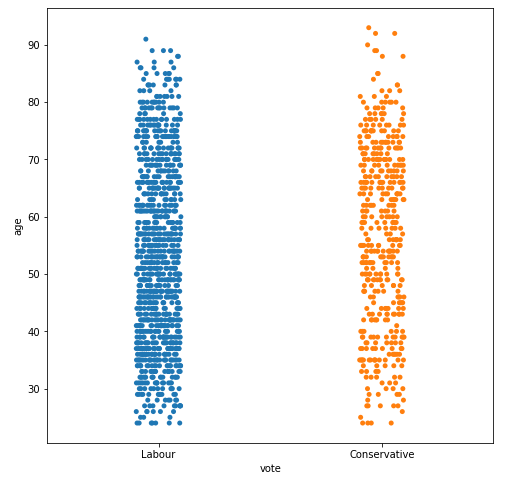


**Image 6: Gender variable bar plot**

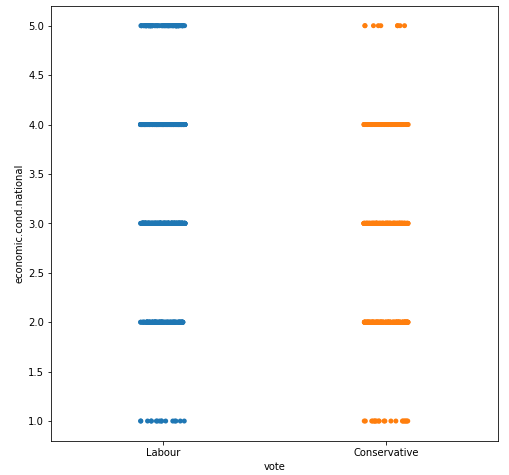
Observations**-**

* Labour has given maximum number of votes in the election.
* Female voters are more compared to male voters.

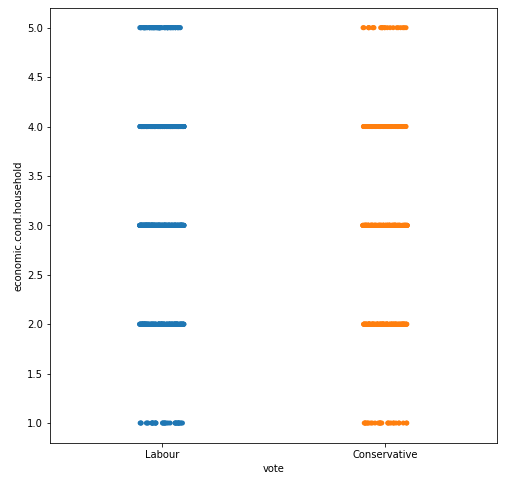
**Bivariate Analysis**



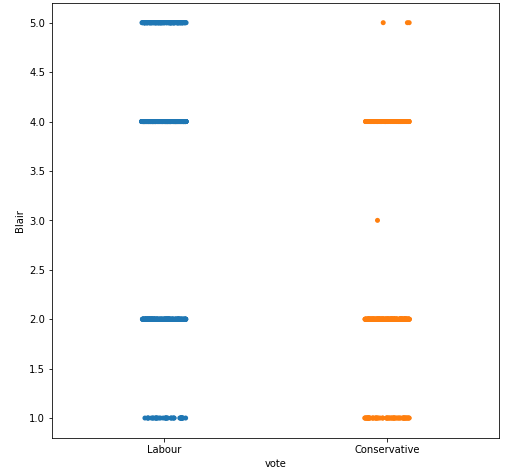
**Image 7: vote vs age strip plot**



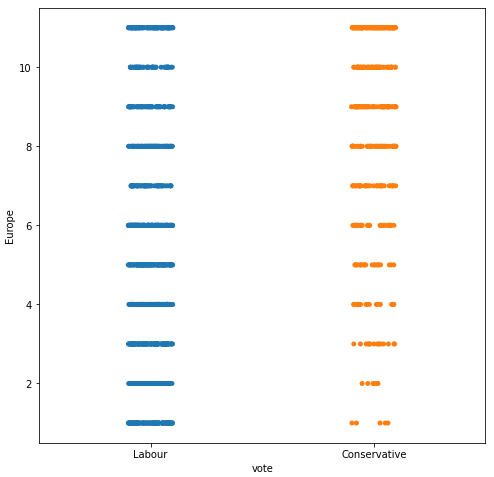
**Image 8: Strip plot for vote vs economic.cond.national**



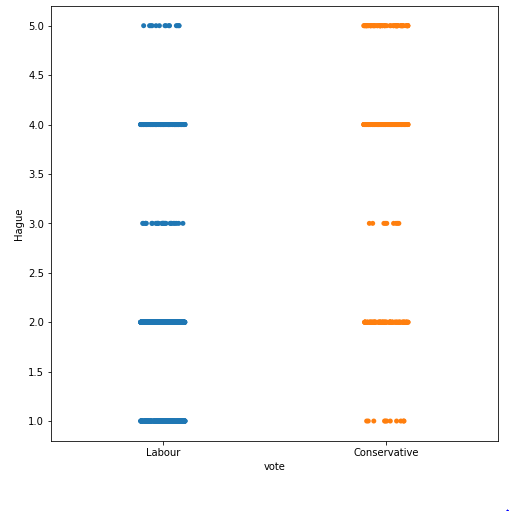
**Image 9: Strip plot for vote vs economic.cond.household**



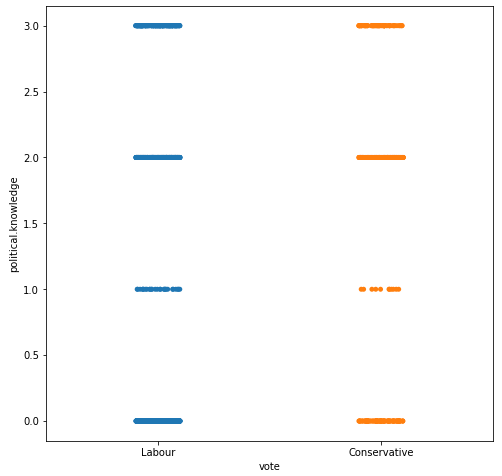
**Image 10: Strip plot for vote vs Blair**



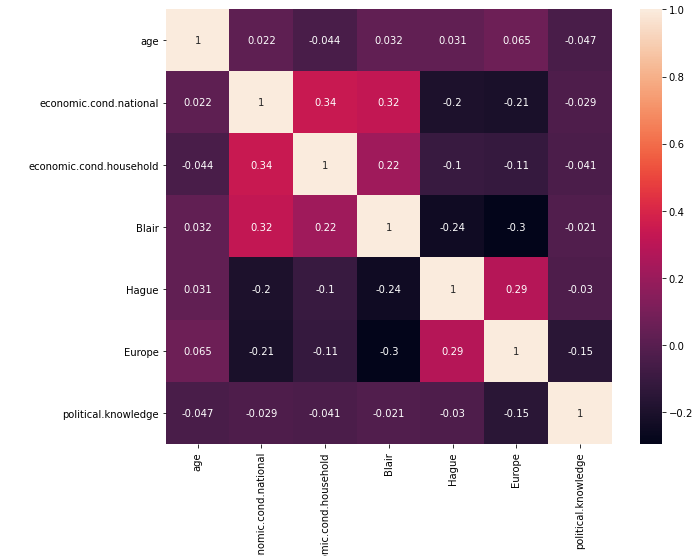
**Image 11: Strip plot for vote vs Europe**



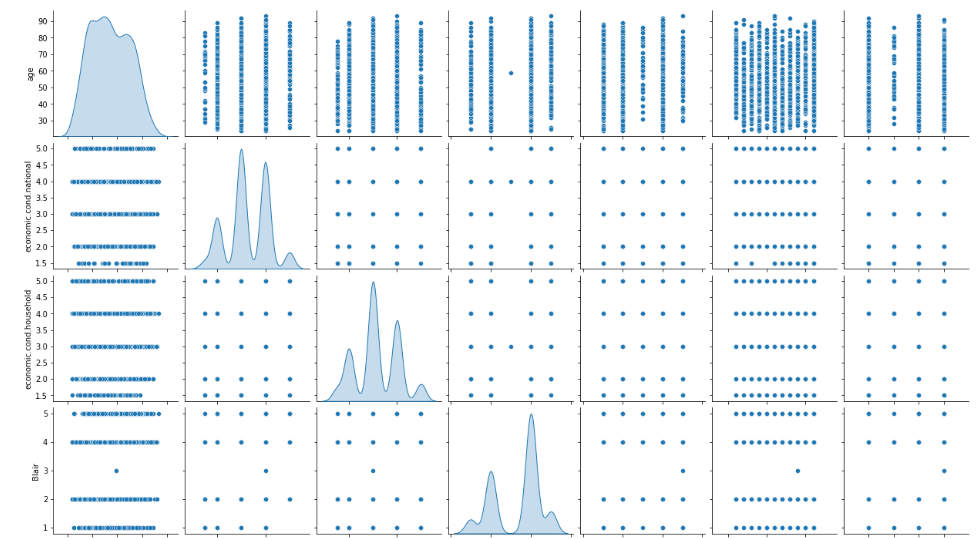
**Image 12: Strip plot for vote vs Hague**

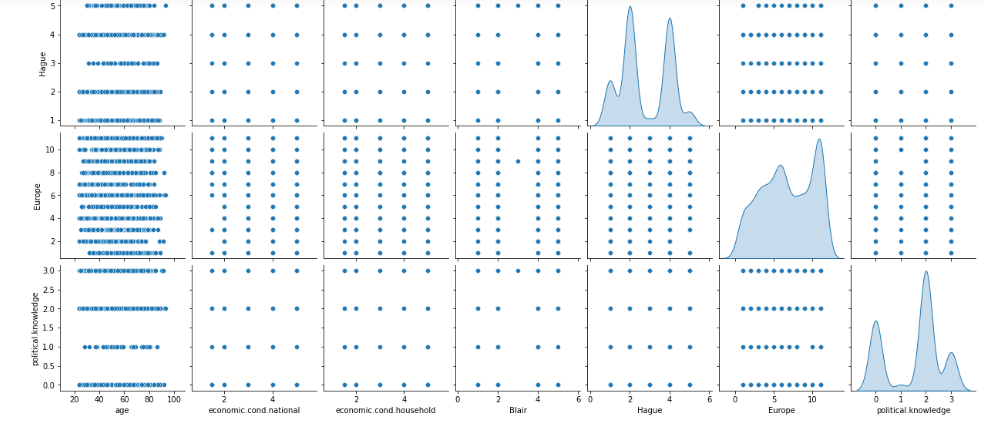


**Image 13: Strip plot for vote vs political.knowledge**

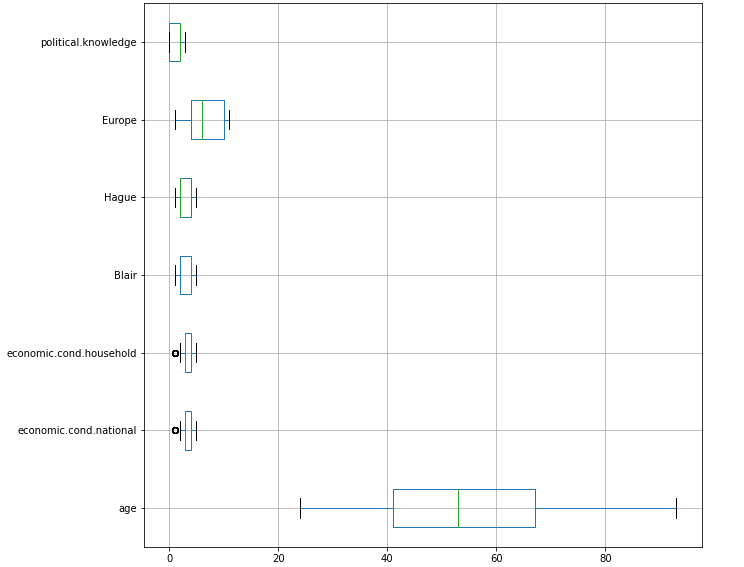


**Image 14: Correlation plot**

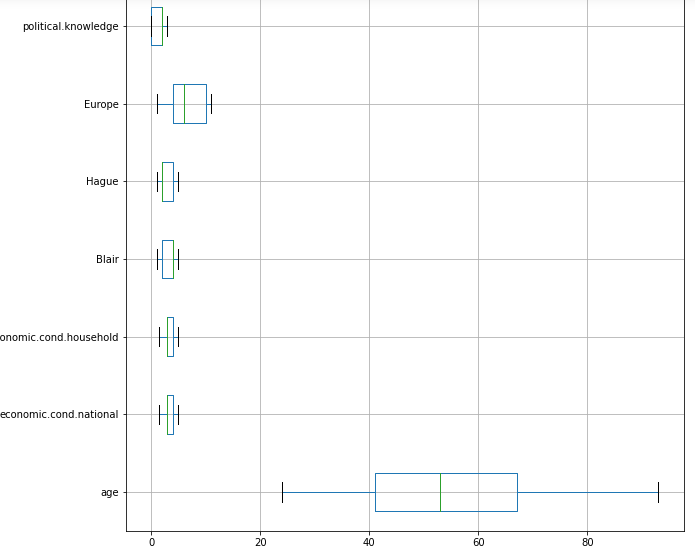




**Image 15: Pair plot of numerical features**



**Image 16: Presence of outliers**



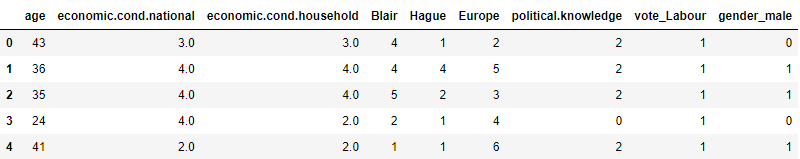
**Image 17: Outlier removal**

Observations**-**

* There is a strong correlation between economic.cond.national and economic.cond.household, Europe and Hague and Blair and economic.cond.national.
* No linear relationship can be established between different variables in the dataset.
* Outlier presence in the data was detected and removed.
* Voters of different ages have voted mostly for labour party.
* Voters of labour party are satisfied with national and household economic policies.
* Both parties have identical political knowledge as indicated from voter’s feedback.
* Blair, Hague and Europe rating parameters signify that labour party has received more votes.

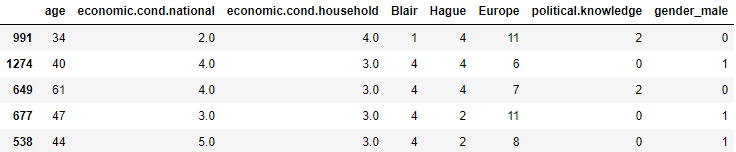
**1.3 Encode the data (having string values) for Modelling. Is Scaling necessary here or not? Data Split: Split the data into train and test (70:30)**

**Drop First is used to ensure that multiple columns created based on the levels of categorical variable are not included else it will result in to multicollinearity. This is done to ensure that we do not land in to dummy trap. A separate data frame is created** consisting of only of the features that is independent attributes. The data is arranged into independent and dependent variables. The dataset is split into training and test set in 70:30 ratio with random state=1.



**Table 6: Dataset post dummy encoding**

Scaling is not necessary for logistic regression, linear discriminant analysis and Naive Bayes model but is required for distance based algorithm like KNN. Scale doesn’t matter. Performing a features scaling in these algorithms may not have much effect.



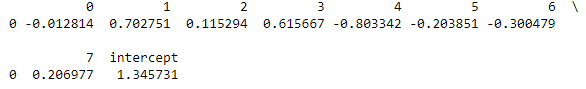
**Table 7: Head of training dataset post train-test split**

**1.4** **Apply Logistic Regression and LDA (linear discriminant analysis).**

**Logistic regression is a linear model for classification rather than regression. It is also known as logit regression. In this model, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.**

**LDA can be derived from simple probabilistic models which model the class conditional distribution of the data.**

**The logistic regression model is fitted on the training data and predicted on the testing data. Hyper parameter solver is considered as “liblinear” and we get a positive intercept of 1.345 and the model coefficients are shown as below –**



**Image 18: Coefficient of determination and intercept for Logistic regression model**

Accuracy of train set is 84% whereas for test set is 83% which are close to each other which represents a perfectly fit logistic regression model.

A basic linear discriminant analysis model is applied on the training sample and predicted on the testing sample for which we get identical model scores of 83% on both the samples. Hence, we can conclude that it is neither an under fit nor an over fit model.

**1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results.**

**Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable.** **In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters.** Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality. GaussianNB implements the Gaussian Naive Bayes algorithm for classification.

**Neighbours-based classification is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbours of each point: a query point is assigned the data class which has the most representatives within the nearest neighbours of the point.**

Now GaussianNB classifier is built. The classifier is trained using training data. We can use fit () method for training it. After building a classifier, our model is ready to make predictions. We can use predict () method with test set features as its parameters.

Using Gaussian algorithm from naïve Bayes, we obtain model accuracy for training data as 83% and for testing data as 82% that signifies good model performance.

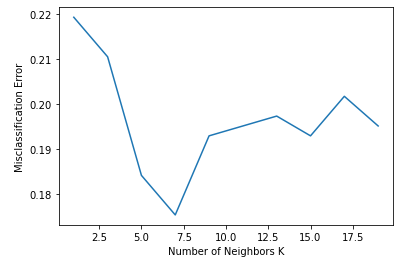
We build the KNN model with different values of number of neighbours 1, 3, 5…19. Starting off with default value of k as 5 and minkowski distance metric, we get train and test accuracies of 85% and 81% respectively.

We need to find the optimal number of neighbours using misclassification error with formula as shown below -

Misclassification error (MCE) = 1 - Test accuracy score. We calculated MCE for each model with neighbours = 1, 3, 5...19 and selected the model with lowest MCE.

[0.2192982456140351, 0.21052631578947367, 0.1842105263157895, 0.17543859649122806, 0.19298245614035092, 0.19517543859649122, 0.19736842105263153, 0.19298245614035092, 0.20175438596491224, 0.19517543859649122]---> MCE for different values of k.

We select k value of 7 for which MCE is minimum as reflected in above list. **As the difference between train (84%) and test (82%) accuracies is 2.36 % which is less than 10% (Industry standard), it is a valid model.**



**Image 19: MCE vs k plot**

**1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging) and Boosting.**

We are making some adjustments to the parameters in the Logistic Regression Class to get a better accuracy. Below are the hyper parameters used in grid search CV for logistic regression model –

{'penalty': ['l2','none'],'solver': ['sag','lbfgs'],'tol': [0.0001, 0.00001]}

Argument=solver{‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’}, default=’lbfgs’ Algorithm to use in the optimization problem.

For small datasets, ‘liblinear’ is a good choice, whereas ‘sag’ and ‘saga’ are faster for large ones.

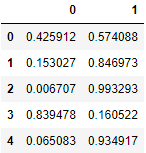
For multiclass problems, only ‘newton-cg’, ‘sag’, ‘saga’ and ‘lbfgs’ handle multinomial loss; ‘liblinear’ is limited to one-versus-rest schemes.

‘newton-cg’, ‘lbfgs’, ‘sag’ and ‘saga’ handle L2 or no penalty

‘liblinear’ and ‘saga’ also handle L1 penalty

‘saga’ also supports ‘elasticnet’ penalty

‘liblinear’ does not support setting penalty='none'



**Table 8: Predicted classes and probabilities**

Model evaluation gives 83% accuracy for both test and train data. But as compared to basic logistic regression model there is a slight dip in training sample accuracy while the test sample results remains the same. We can select other parameters to perform GridSearchCV and try optimize the desired parameter.

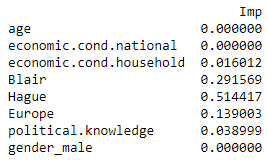
Now we perform model tuning for KNN by changing the distance measuring metric from Minkowski to Euclidean. The learning model is instantiated with different values of k like 3, 5 and 9, it is fitted into training samples and the response is predicted on testing sample. The accuracies for K values 3, 5 and 9 are 78%, 81% and 80% respectively which are same as the model with default values.

**Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The deeper the tree, the more complex the decision rules and the fitter the model.**

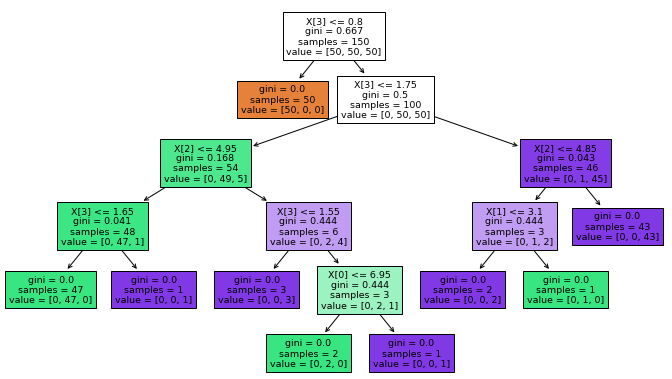
We will build our model using the DecisionTreeClassifier function. Using default 'Gini' criteria to split. Other option include 'entropy'.

We obtain accuracy of 100% for training data and 79% for testing data, the difference clearly shows that model needs to be regularized.

To reduce overfitting we regularize the model by setting maximum depth to 3, we get an accuracy of 80% and 78% for train and test sets respectively that improves the performance to a greater extent.



**Image 20: Feature importance of regularized DT model**



**Image 21: Decision tree visualization**

In random forests, each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. Furthermore, when splitting each node during the construction of a tree, the best split is found either from all input features or a random subset of size max\_features.

Considering number of estimators as 100, accuracy of training data is 100% but for testing data is 83%.

**A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction.**

**Bagging model is built on base estimator as random forest model and number of estimators as 50, accuracy of training sample is obtained as 96% and 83% for testing sample.**

The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction.

The number of weak learners is controlled by the parameter n\_estimators. The learning\_rate parameter controls the contribution of the weak learners in the final combination. By default, weak learners are decision stumps. Different weak learners can be specified through the base\_estimator parameter. The main parameters to tune to obtain good results are n\_estimators and the complexity of the base estimators (e.g., its depth max\_depth or minimum required number of samples to consider a split min\_samples\_split).

Model performs well with number of estimators as 100 as accuracy of train and test data is 85% and 81% respectively.

**Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.**

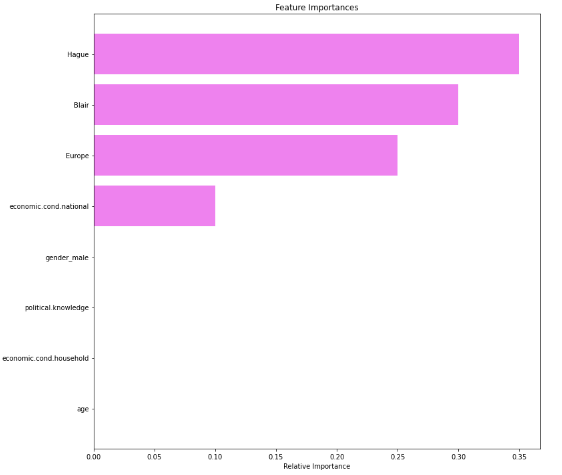
Training and testing set accuracies for above model is 89% and 84% respectively which indicates good performance.

We apply tuning to ada boosting and gradient boosting models to check on improvement in performance.

Trying different maximum depth values with base estimator as decision tree in grid search hyperparameters, we get the below best combination of parameters –

AdaBoostClassifier (base\_estimator=DecisionTreeClassifier (max\_depth=1), learning\_rate=0.1, n\_estimators=20, random\_state=1)

Accuracy of tuned Ada boosting model for training and testing sets are 79% and 77% respectively which is a decrement in performance in comparison to base model.



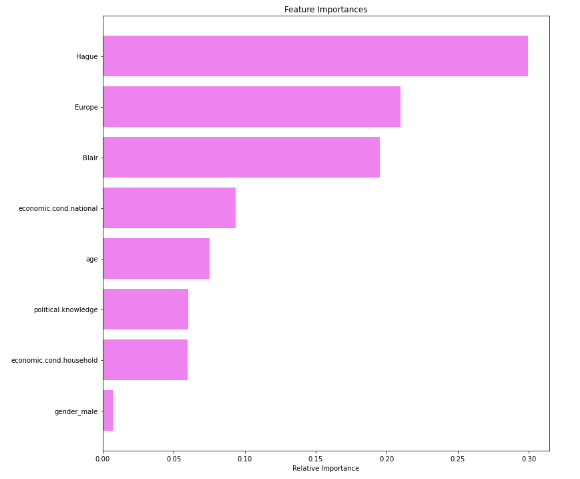
**Image 22: Feature importance of tuned Ada boost**

Grid parameters selected for tuned gradient boosting model are as follows –

{"n\_estimators": [100,150,200,250],"subsample": [0.8, 0.9, 1],"max\_features": [0.7, 0.8, 0.9, 1]}

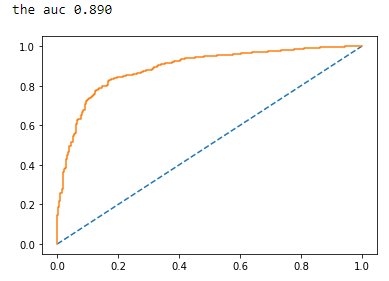
Best estimator of tuned model – GradientBoostingClassifier (init=AdaBoostClassifier (random\_state=1), max\_features=1, random\_state=1, subsample=1)

Accuracy of training and testing samples are 87% and 83% respectively which is also a let down from the previous instance.

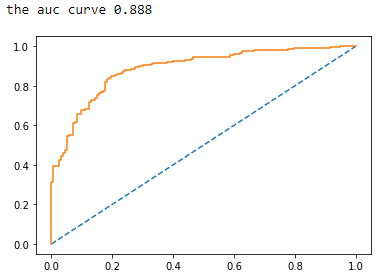


**Image 23: Feature importance of tuned gradient boost**

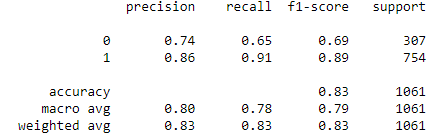
**1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized.**



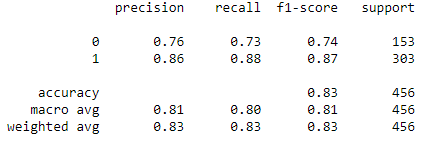
**Image 24: ROC curve for LDA train**



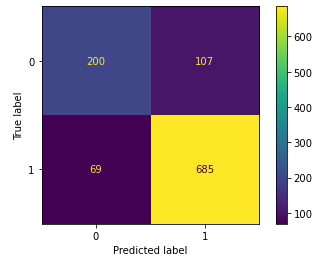
**Image 25: ROC curve for LDA test**



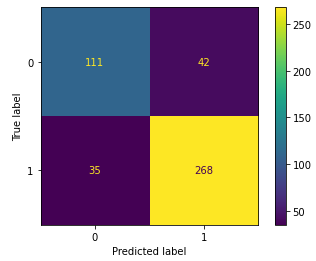
**Image 26: Classification report for LDA train**



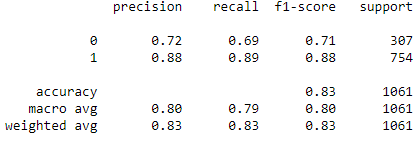
**Image 27: Classification report for LDA test**



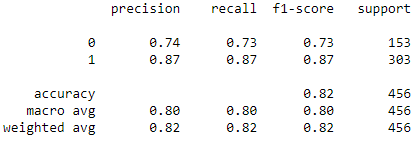
**Image 28: Confusion matrix for LDA train**



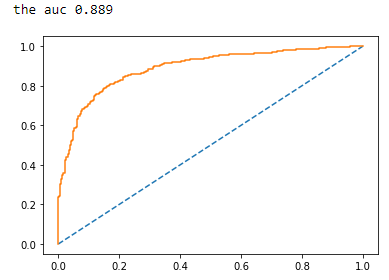
**Image 29: Confusion matrix for LDA test**



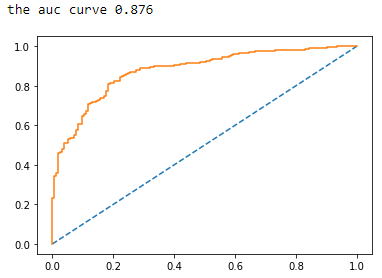
**Image 30: Naïve Bayes training sample classification report**



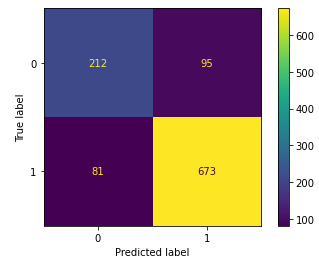
**Image 31: Naïve Bayes testing sample classification report**



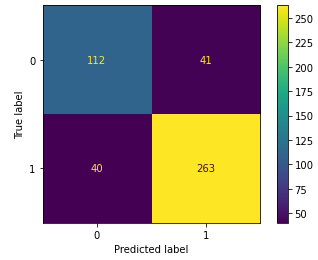
**Image 32: Naïve Bayes ROC curve for train data**



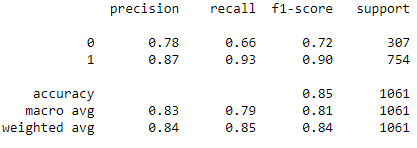
**Image 33: Naïve Bayes ROC curve for test data**



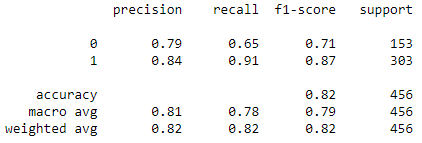
**Image 34: Naïve Bayes confusion matrix for train data**



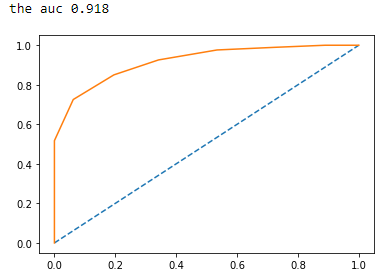
**Image 35: Naïve Bayes confusion matrix for test data**



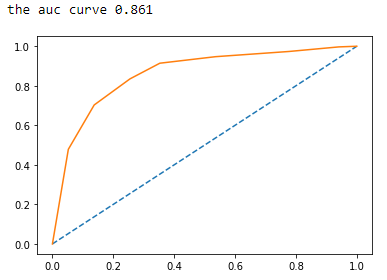
**Image 36: Classification report for optimal KNN training data**



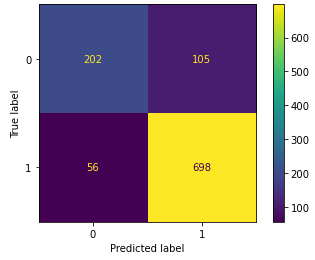
**Image 37: Classification report for optimal KNN testing data**



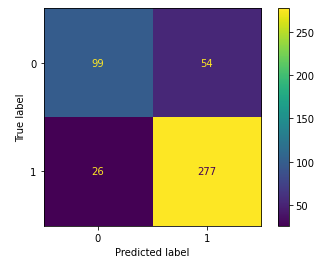
**Image 38: ROC curve for optimal KNN trained sample**



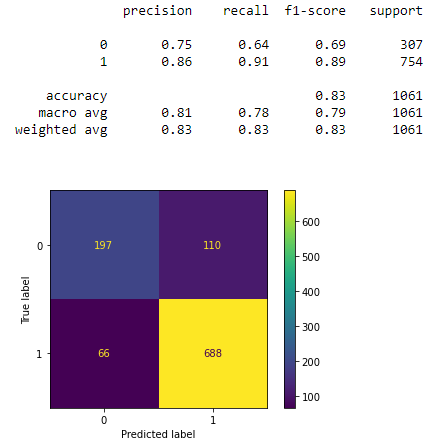
**Image 39: ROC curve for optimal KNN tested sample**



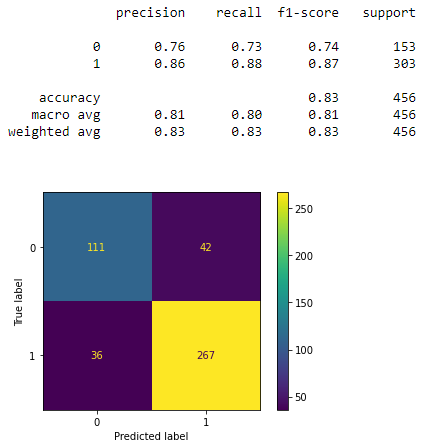
**Image 40: Confusion matrix for optimal KNN train**



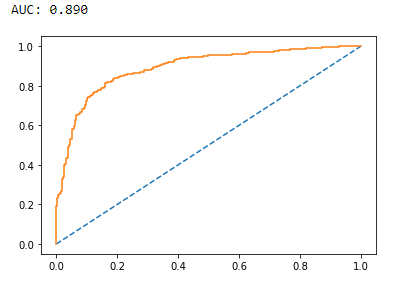
**Image 41: Confusion matrix for optimal KNN test**



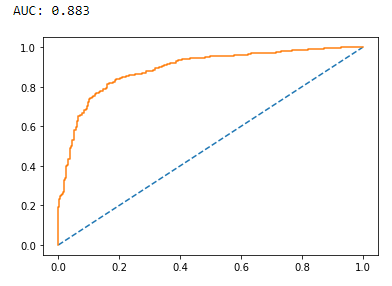
**Image 42: Confusion matrix and classification report for logistic regression train**



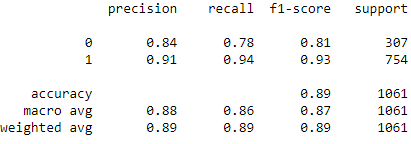
**Image 43: Confusion matrix and classification report for logistic regression test**

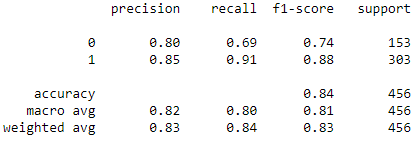


**Image 44: ROC curve for logistic regression train**

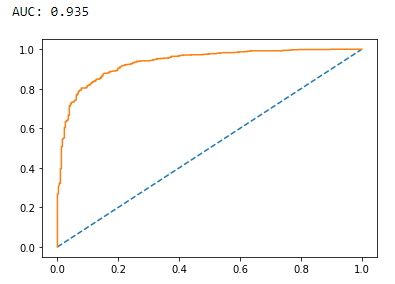


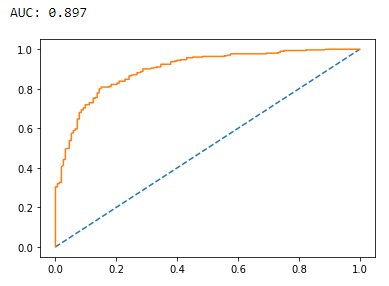
**Image 45: ROC curve for logistic regression test**



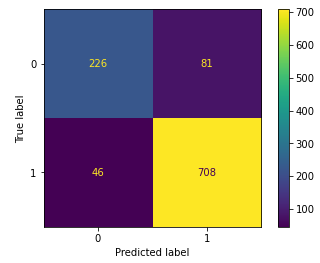


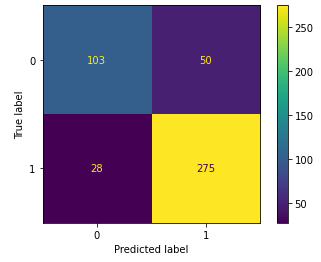
**Image 46: Gradient Boosting classification reports for train and test**



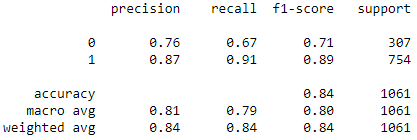


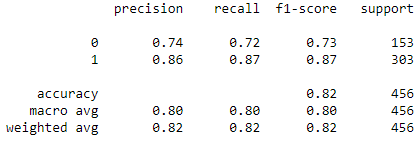
**Image 47: Gradient Boosting ROC curve for train and test**



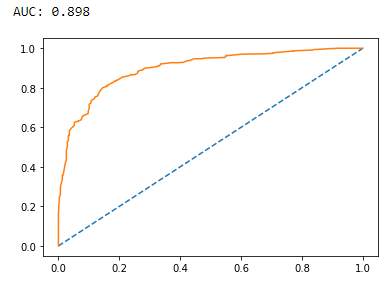


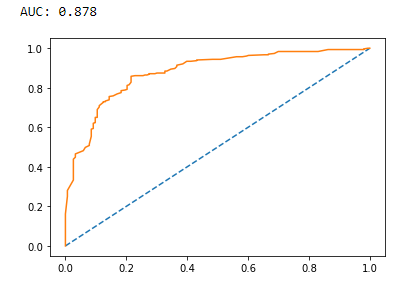
**Image 48: Gradient boosting confusion matrix for train and test**



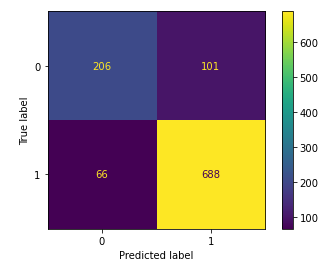


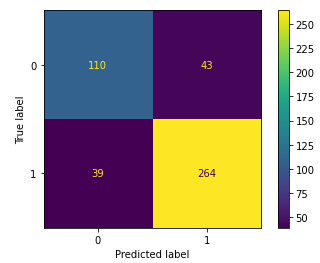
**Image 49: Adaptive boosting classification report for train and test**



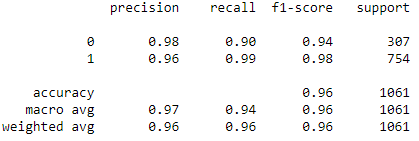


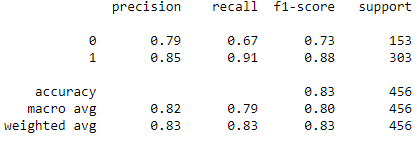
**Image 50: Adaptive boosting ROC curve for train and test**



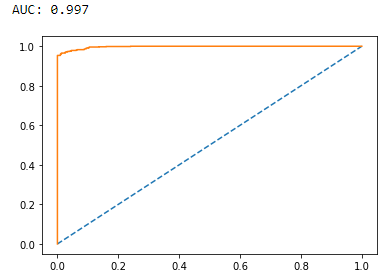


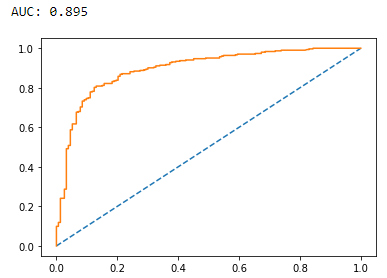
**Image 51: Adaptive boosting confusion matrix for train and test**



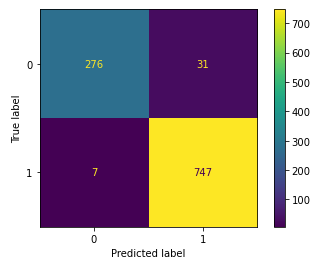


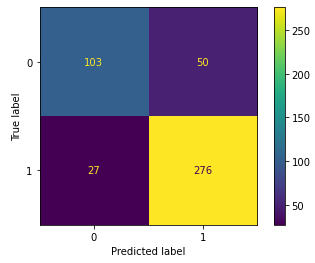
**Image 52: Bagging classification report for train and test**



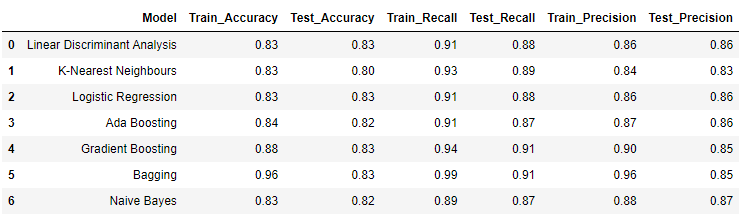


**Image 53: Bagging ROC curve for train and test**





**Image 54: Bagging confusion matrix for train and test**



**Table 9: Model comparison**

Linear discriminant analysis or logistic regression can be selected as the final model for the problem at hand as both perform equally well on training and testing sets based on the parameters shown in above table. Bagging shows excellent results on training data but poorly on testing data. KNN, adaptive boosting and gradient boosting displays decent performance on both training and testing samples.

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

Business insights and recommendations

* Labour party has the highest chance of winning the election.
* Tony Blair from labour party will receive more votes than Hague from conservative party as voters are impressed with their national and household economic policies.
* Europe is also an important factor in deciding the fate of the candidates in the election.
* In order to win the election, conservative party should either improve the image of their candidate or field an entire new face by taking feedback from their supporters.
* Labour party should focus on senior citizens who seem to be more inclined towards the conservative party and provide schemes suiting their demands.
* Female voters are higher in comparison to male voters. Both parties should try to woo male voters to increase votes.

**Problem 2**

**Problem Statement:**

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

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

|  |  |  |  |
| --- | --- | --- | --- |
| President | Number of characters | Number of words | Number of sentences |
| Roosevelt | 7571 | 1536 | 68 |
| Kennedy | 7618 | 1546 | 52 |
| Nixon | 9991 | 2028 | 69 |

**Table 10: Character, word and sentence count of speeches**

**2.2 Remove all the stop words from all three speeches.**

The words in each of the text documents containing speeches of former U.S presidents were converted to lowercase, punctuations and stop words were removed.

Word count before removal of stop words from Roosevelt’s speech=1536

Word count after removal of stop words from Roosevelt’s speech=632

Sample sentence post stop word removal from Roosevelt’s speech= ['On ', 'national', 'day', 'inauguration', 'since', '1789', 'people', 'renewed', 'sense', 'dedication', 'United', 'States', 'In', 'Washington', "'s", 'day', 'task', 'people', 'create', 'weld']

Word count before removal of stop words from Kennedy’s speech=1546

Word count after removal of stop words from Kennedy’s speech=697

Sample sentence post stop word removal from Kennedy’s speech= ['Vice', 'President', 'Johnson', 'Mr.', 'Speaker', 'Mr.', 'Chief', 'Justice', 'President', 'Eisenhower', 'Vice', 'President', 'Nixon', 'President', 'Truman', 'reverend', 'clergy', 'fellow', 'citizens', 'observe']

Word count before removal of stop words from Nixon’s speech=2028

Word count after removal of stop words from Nixon’s speech=836

Sample sentence post stop word removal from Nixon’s speech= ['Mr.', 'Vice', 'President', 'Mr.', 'Speaker', 'Mr.', 'Chief', 'Justice', 'Senator', 'Cook', 'Mrs.', 'Eisenhower', 'fellow', 'citizens', 'great', 'good', 'country', 'share', 'together', 'When']

**2.3 Which word occurs the most number of times in his inaugural address for each president? Mention the top three words.**

A frequency distribution of lowercase words excluding stop words are created for each document and the most common 2000 words are extracted.

[('nation', 12),('know', 10),('spirit', 9),('life', 9),('democracy', 9),('us', 8),('people', 7),('america', 7),('years', 6),('freedom', 6)]----> Word occurrence count in Roosevelt’s speech.

['nation', 'know', 'spirit']---> Top 3 words in Roosevelt’s speech.

[('let', 16), ('us', 12), ('world', 8), ('sides', 8), ('new', 7), ('pledge', 7), ('citizens', 5), ('power', 5), ('shall', 5), ('free', 5)] ----> Word occurrence count in Kennedy’s speech.

['let', 'us', 'world'] ---> Top 3 words in Kennedy’s speech.

[('us', 26),('let', 22),('america', 21),('peace', 19),('world', 18),('new', 15),('nation', 11),('responsibility', 11),('government', 10),('great', 9)] ----> Word occurrence count in Nixon’s speech.

['us', 'let', 'america'] ---> Top 3 words in Nixon’s speech.

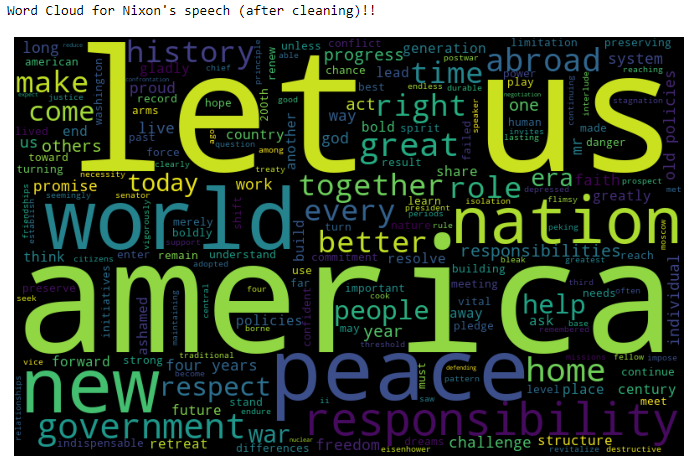
**2.4 Plot the word cloud of each of the speeches of the variable.**



**Image 55: Word cloud for Roosevelt’s speech**



**Image 56: Word cloud for Kennedy’s speech**



**Image 57: Word cloud for Nixon’s speech**