Deepak Singh Manola

19th March 2023

Great Learning – PGP – DSBA Machine Learning Project

Contents

* 1. [Read the dataset. Do the descriptive statistics and do the null value condition](#_bookmark0) [check? Write an inference on it 3](#_bookmark0)
  2. [Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for](#_bookmark7) [Outliers. 6](#_bookmark7)
  3. [Encode the data (having string values) for Modelling. Is Scaling necessary here or](#_bookmark27) [not? Data Split: Split the data into train and test (70:30) 16](#_bookmark27)
  4. [Apply Logistic Regression and LDA (linear discriminant analysis) 17](#_bookmark29)
  5. [Apply KNN Model and Naïve Bayes Model. Interpret the results 18](#_bookmark30)
  6. [Model Tuning, Bagging (Random Forest should be applied for Bagging), and](#_bookmark31) [Boosting 19](#_bookmark31)  
     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………………………………………………………………..

[1.8 Based on these predictions, what are the insights? (5 marks) 32](#_bookmark70)

* 1. [Find the number of characters, words, and sentences for the mentioned](#_bookmark71) [documents 33](#_bookmark71)
  2. [Remove all the stop words from all three speeches 34](#_bookmark72)
  3. [Which word occurs the greatest number of times in his inaugural address for each](#_bookmark73) [president? Mention the top three words. (after removing the stop words) 34](#_bookmark73)
  4. [Plot the word cloud of each of the speeches of the variable. (after removing the](#_bookmark74) [stop words) – 3 Marks [ refer to the End-to-End Case Study done in the Mentored](#_bookmark74) [Learning Session ] 35](#_bookmark74)

**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.**

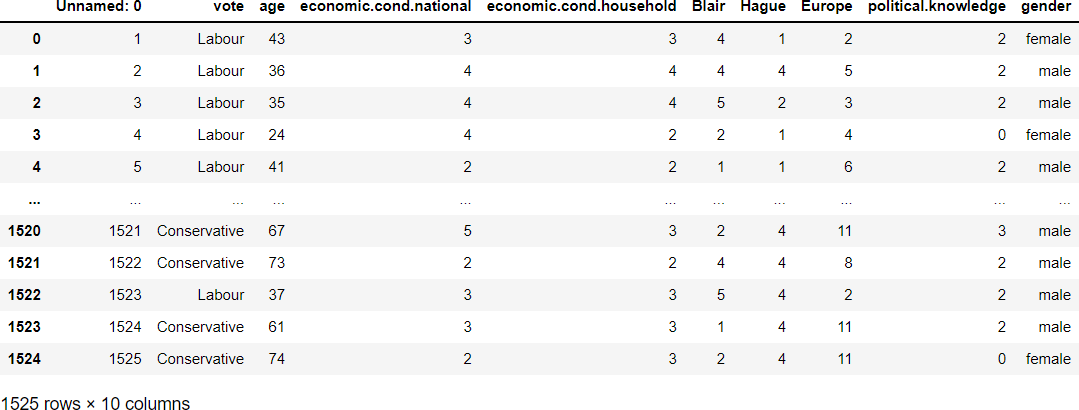
# Read the dataset. Do the descriptive statistics and do the null value condition check? Write an inference on it.

Data Dictionary

We can start exploring the dataset further to understand the information that it holds. The data dictionary for the given dataset is as follows

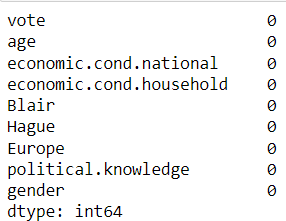
* + 1. vote: Party choice: Conservative or Labor (the vote that an individual undertook)
    2. age: in years (the age of the individual
    3. economic.cond.national: Assessment of current national economic conditions, 1 to 5.
    4. economic.cond.household: Assessment of current household economic 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.

We can load the dataset into the Python Program to further study the information that is available.



*Figure 1 Dataframe of Election Data*

We can understand from further exploring the Data that there are no missing values within the given dataframe.



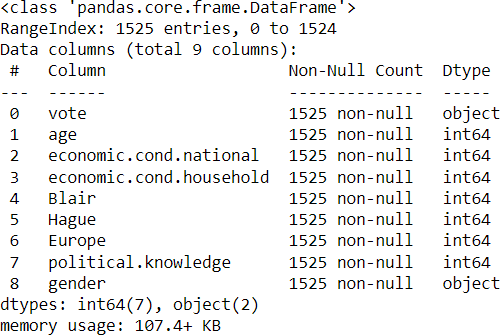
*Figure 2 Exploring Missing values*

The shape of the Dataset is



1525 Rows and 9 columns. The Unnamed column consist of a serial number that can be removed before processing the data.

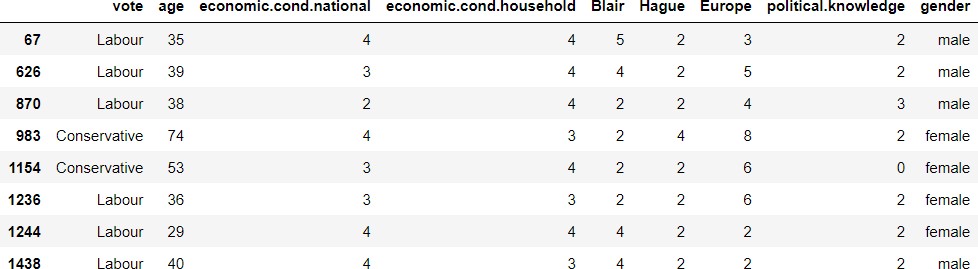
The Dataset consist of Integer Datatype and Object Datatype. The Dataset has been divided into continuous and categorical as well.



*Figure 3 Information of the Dataset*

* Categorical Variables
  + Vote
  + Economic Condition national
  + Economic Condition household
  + Blair
  + Hague
  + Europe
  + Political Knowledge
  + Gender
* Continuous Variable
  + Age

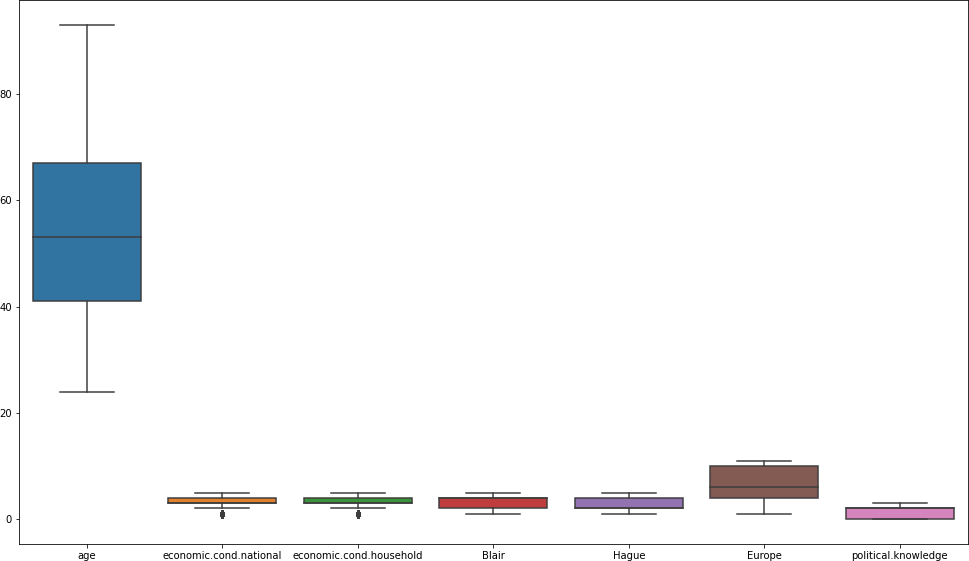
We can check for duplicate values within the Data frame



*Figure 4 Duplicate Values within the Dataset*

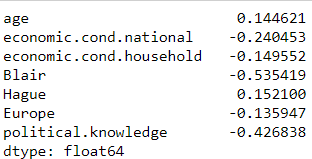
We can see that the compiler has produced 8 duplicate entries, however upon examining them closely we can understand that they are unique attributes within the different variables.

We can look for outliers within the dataset.



*Figure 5 Boxplot of the Dataset*

We can see from the given Boxplot that only Economic Condition National & Economic Condition Household has outliers within the dataset.

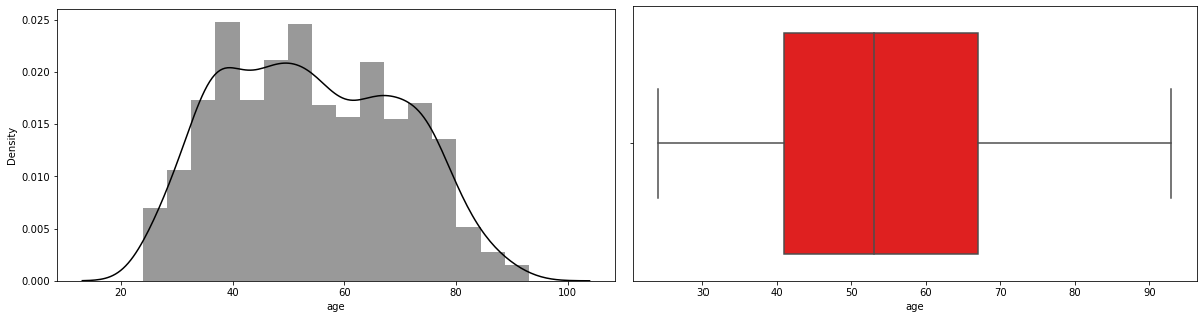


*Figure 6 Skewness of Dataframe*

There is a small level of skewness within the Election Dataset, we can further explore on the same while conducting the Univariate analysis.

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

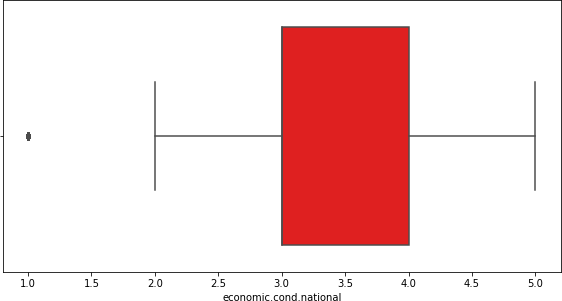
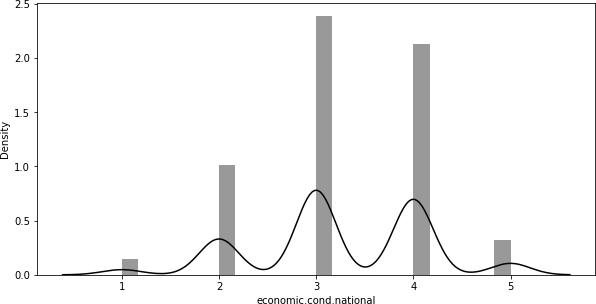
**Description of the Age Column**



*Figure 7 Distribution and Boxplot of Age*

* Mean value of Column - 54.18229
* Standard Deviation - 15.71120
* The minimum Age seen in the Dataset 24
* The maximum Age seen in the Dataset 93
* Interquartile Range Q1 – 41
* Interquartile Range Q3 – 67
* The distribution seems to be Multimodal in Nature
* There are no outliers within the column that needs to be treated

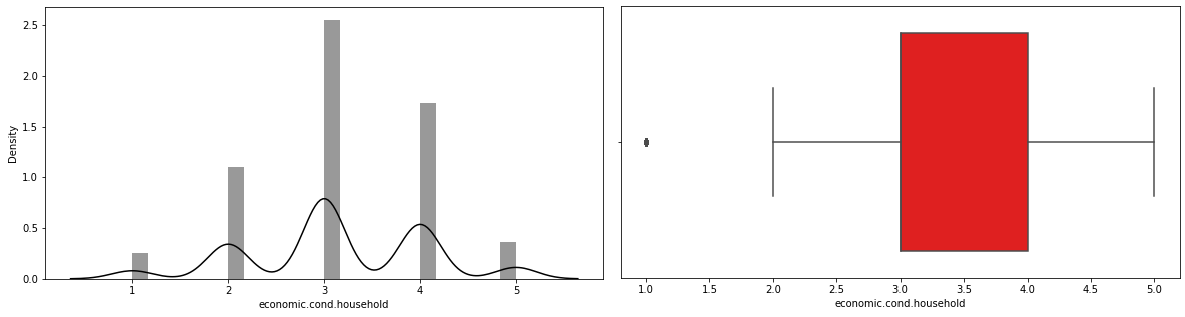
**Description of economic.cond.national**



*Figure 8 Distribution and Boxplot of Economic Conditional National*

* Mean value of Column - 3.2459
* Standard Deviation - 0.88096
* The minimum Economic Condition seen in the Dataset 1
* The maximum Economic Condition seen Nationally in the Dataset 5
* The column is categorical in nature, we can see that majority of the dataset consists of individuals at level 3 and level 4
* There are few outliers beyond Q1 that may be treated.

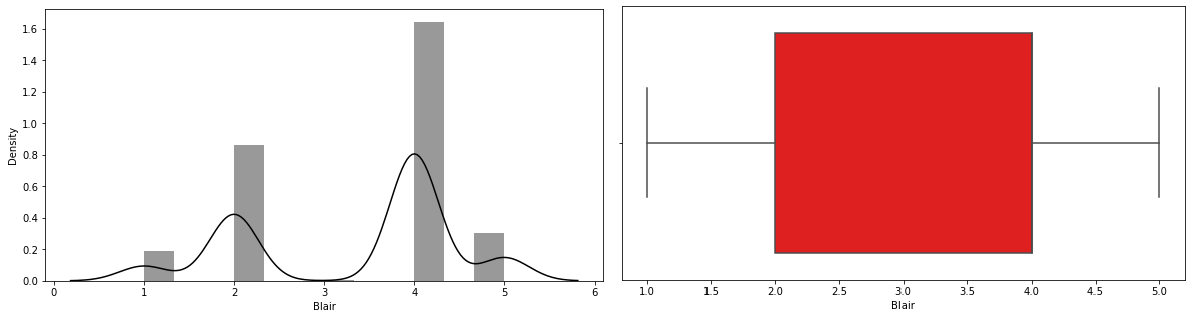
**Description of economic.cond.household**



*Figure 9 Distribution and Boxplot of Economic Conditional Household*

* Mean value of Column - 3.1403
* Standard Deviation - 0.92995
* The minimum Economic Condition household seen in the Dataset 1
* The maximum Economic Condition household seen Nationally in the Dataset 5
* The column is categorical in nature, we can see that majority of the dataset consists of individuals at level 3
* There are few outliers beyond Q1 that may be treated.

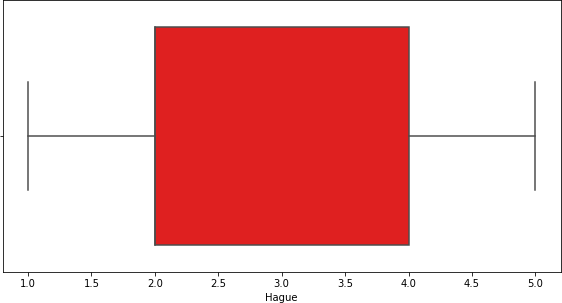
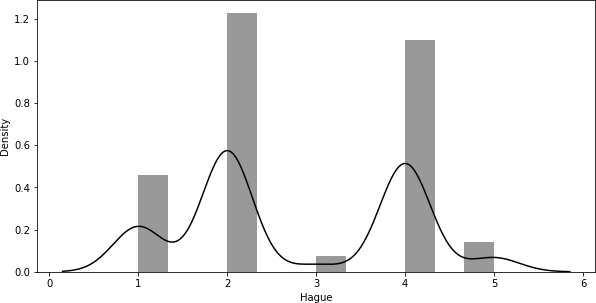
**Description of Blair**



*Figure 10 Distribution and Boxplot of Blair*

* Mean value of Column - 3.3344
* Standard Deviation - 1.17482
* The minimum value seen for Blair seen in the Dataset 1
* The maximum value seen for Blair seen Nationally in the Dataset 5
* The column is categorical in nature, we can see the dataset has been distributed with value of 1-2 and then 4 and 5.
* The distribution shows a value of 4 as the highest entry for Blair
* There is only 1 entry who has given Blair a value of 3.
* There are no outliers in the column.

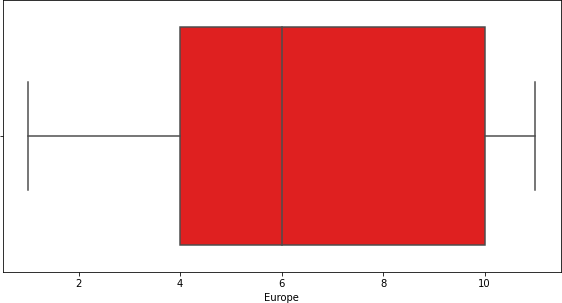
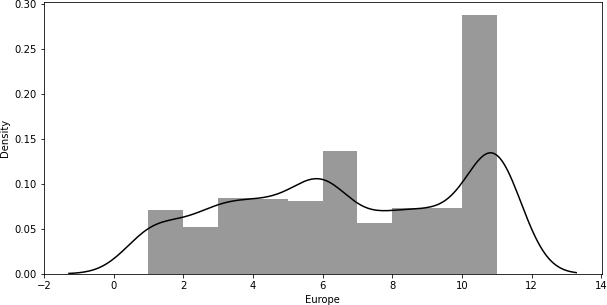
**Description of Hague**



*Figure 11 Distribution and Boxplot of Hague*

* Mean value of Column - 2.74688
* Standard Deviation - 1.2307
* The minimum value seen for Hague seen in the Dataset 1
* The maximum value seen for Hague seen Nationally in the Dataset 5
* The column is categorical in nature, we can see that majority of entries are for 2 and 4 in the distribution
* There are no outliers that may need to be treated.

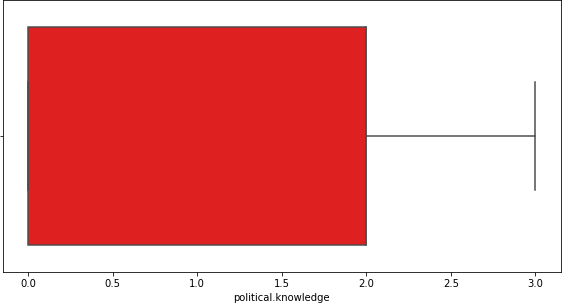
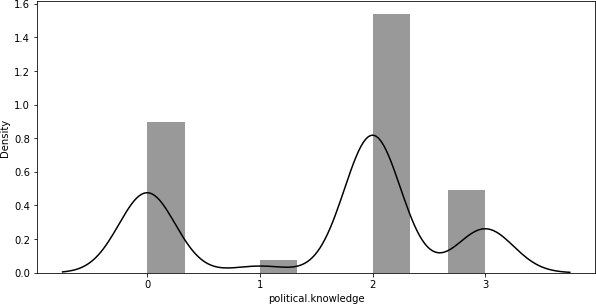
**Description of Europe**



*Figure 12 Distribution and Boxplot of Europe*

* Mean value of Column - 6.7285
* Standard Deviation - 3.297
* The minimum value within Europe column seen in the Dataset 1
* The maximum value within Europe column seen Nationally in the Dataset 11
* The column is categorical in nature, the column looks to understand the individuals who have Eurosceptic sentiment
* There are a large number of entries within Data frame for Eurosceptic sentiment values at 11.
* The second highest entry can be seen at 6 which represents individuals who show mediocre interest to the same when undertaking voting decisions.

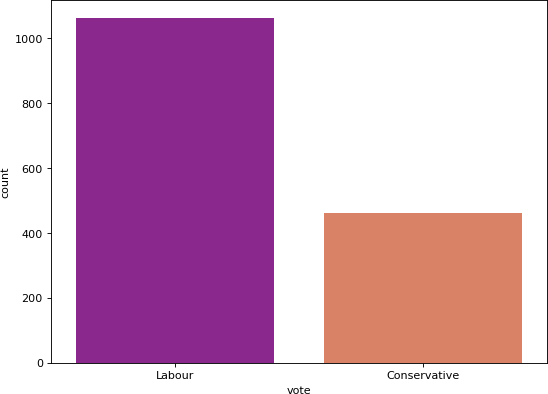
**Description of Political Knowledge**



*Figure 13 Distribution and Boxplot of Political Knowledge*

* Mean value of Column - 1.5422
* Standard Deviation - 1.08331
* The minimum Economic Condition seen in the Dataset 0
* The maximum Economic Condition seen Nationally in the Dataset 3
* The column is categorical in nature, we can see that majority of the dataset consists of individuals at level 2 followed by Level 0.

**Distribution of Vote**



*Figure 14 Countplot of Vote*

The dataset has a larger number of data within the Election Dataset belongs to votes undertaken for the Labour party.

This is important to understand that the dataset that we need to process would not be balanced in nature.

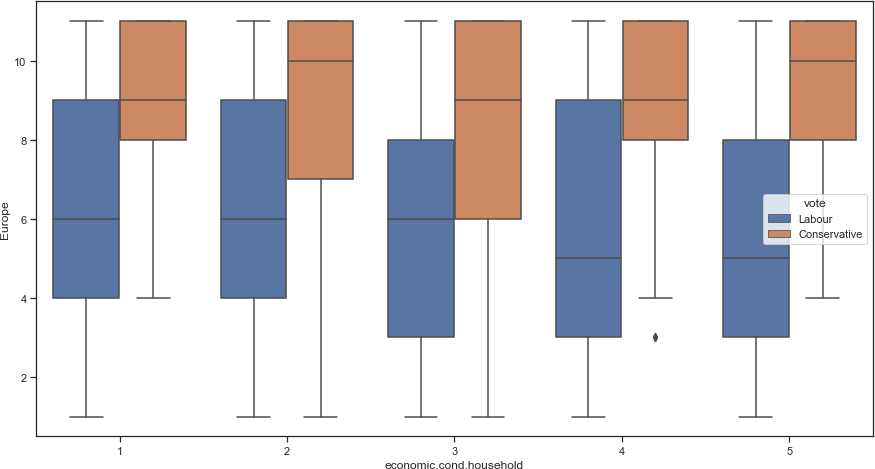
**Distribution of Gender**



*Figure 15 Countplot of Gender*

We can understand from the countplot that the Dataset has an equal number of entries between both Male and Female candidates who have cast their votes.

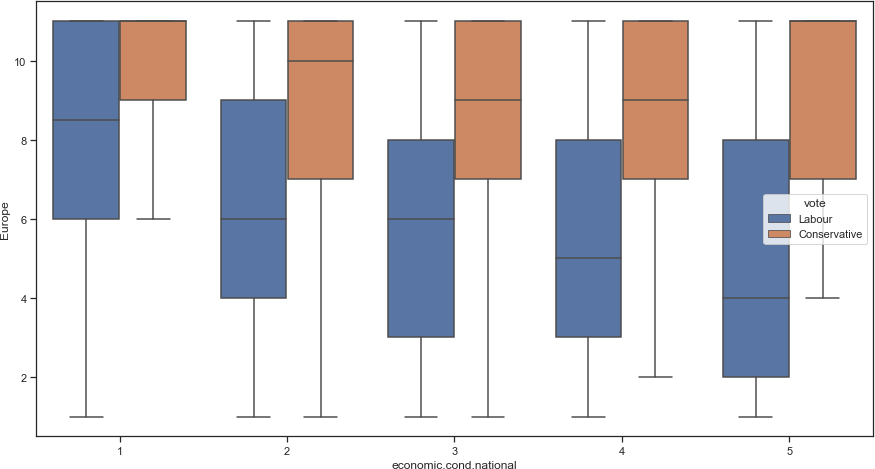
We can further try to understand the Dataset by Conducting the Bivariate Analysis.



*Figure 16 Economic Household VS Europe*

We can understand from the boxplot

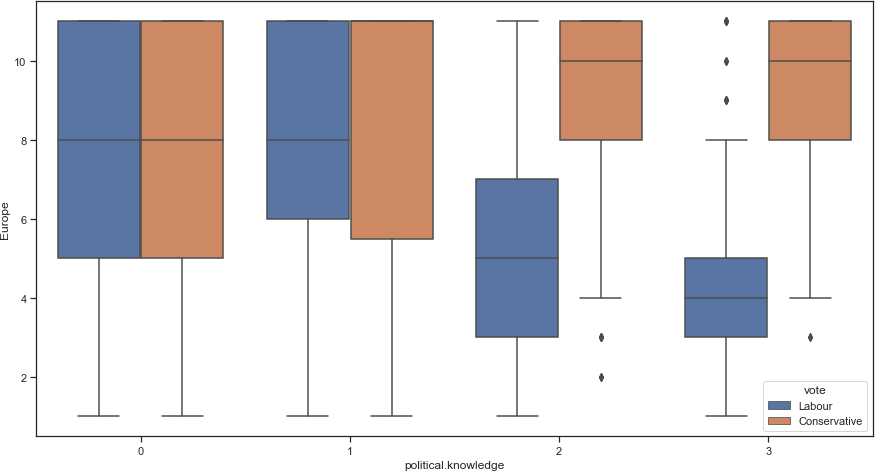
* Voters who chose Labour Party have lower Eurosceptic Sentiment
* Conservative voters independent of their economic household condition have higher levels of Eurosceptic sentiment.



*Figure 17 Economic Conditional National VS Europe*

We can understand from the given boxplot that

* + As the economic conditions improved on the national scale the Euroseptic sentiment had reduced.
  + However, individuals who vote conservative showed very low changes as the economic conditions are measured on a national level.



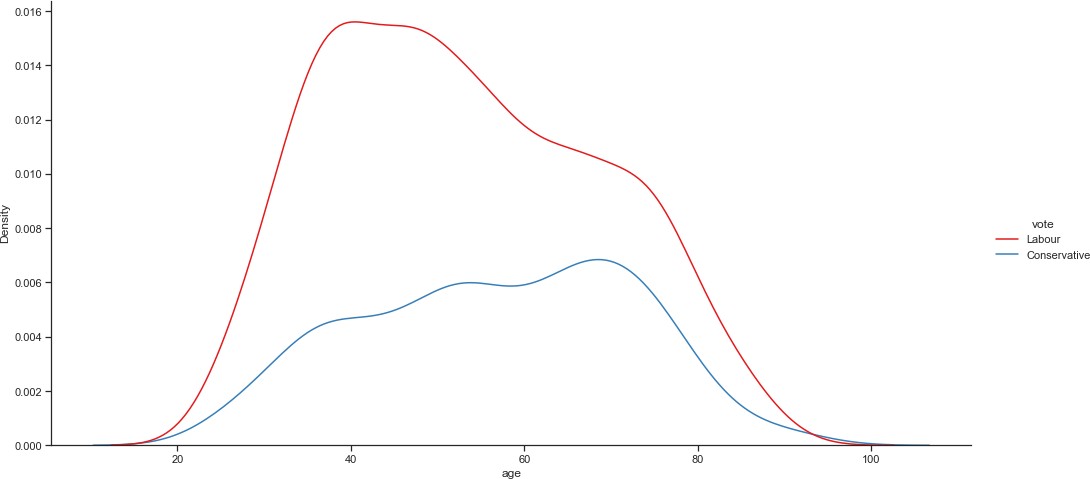
*Figure 18 Political Knowledge VS Europe*

We can understand from the boxplot

Individuals who have scored low political knowledge have shown higher Eurosceptic sentiment

As the Political Knowledge improved voters of labour party showed lower value of Eurosceptic sentiment.

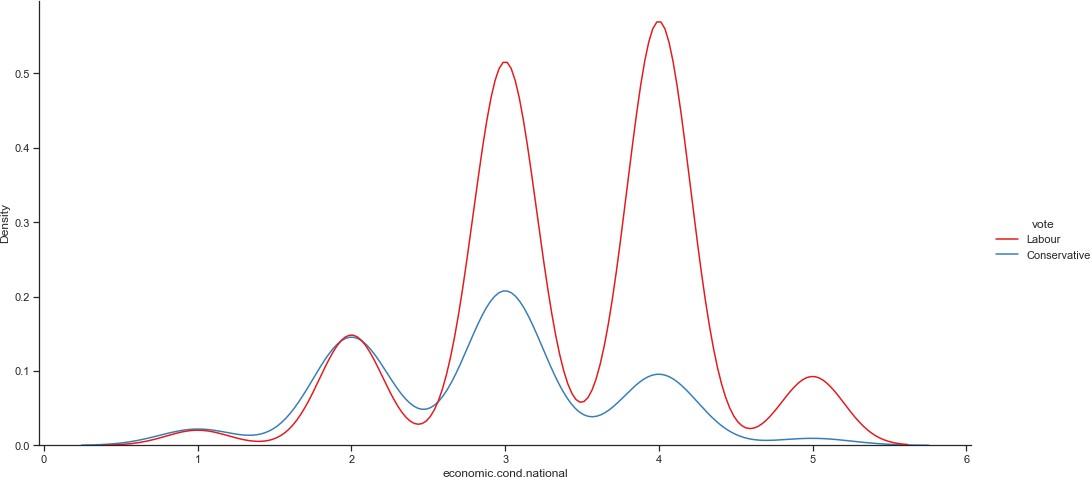
Conservative voters did not improve on their Eurosceptic sentiment even with improved Political Knowledge



*Figure 19 Age VS Vote*

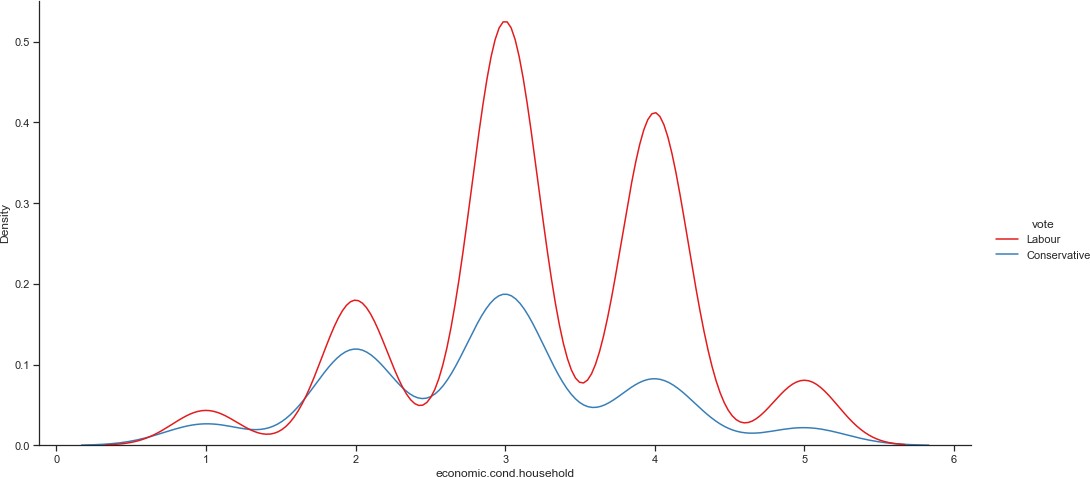
From the plot we can see that majority of the voters who chose the Labour Party were younger in age than those who voted for the Conservative Party

The mode of those voting for the labour party is before the age of 40 and those voting for the Conservative party is beyond 60 years of age.



*Figure 20 Economic Condition National VS Vote*

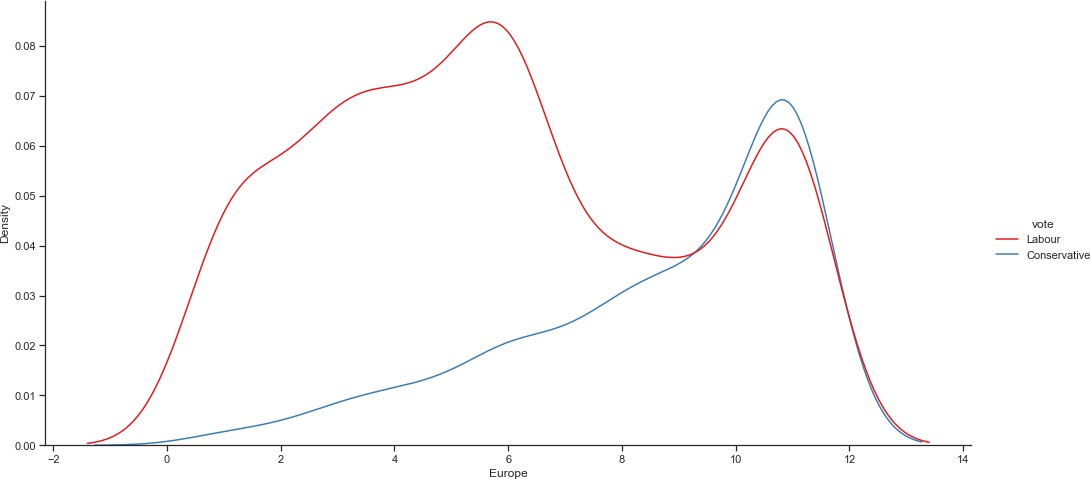
The individuals who voted for the labour party have an economic scale of 3 and 4. The individuals who voted for conservative party were seen within 3



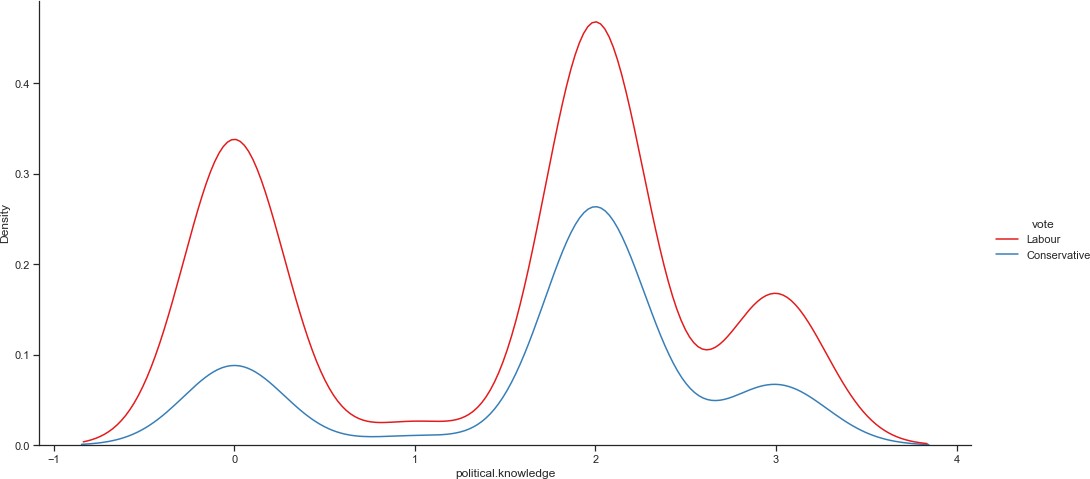
*Figure 21 Economic Household VS Vote*

The voters for the labour party as seen similar with the economic national scale fall within categories of 3 and 4

The voters for the conservative party fall within the category 3



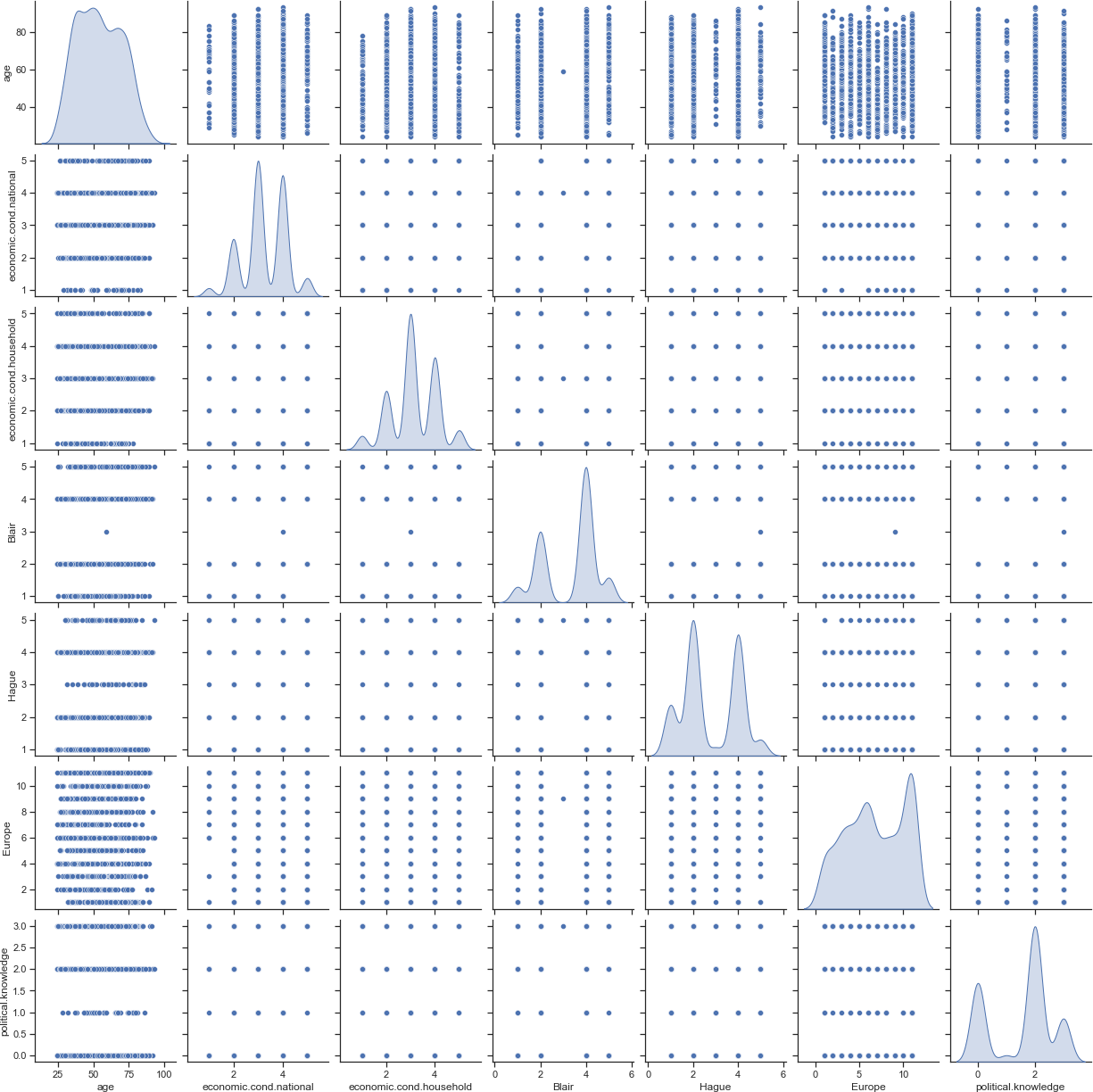
*Figure 22 Europe VS Vote*

The Eurosceptic sentiment within the Labour Vote seems bimodal in nature. The conservative voters have very high Eurosceptic Sentiment

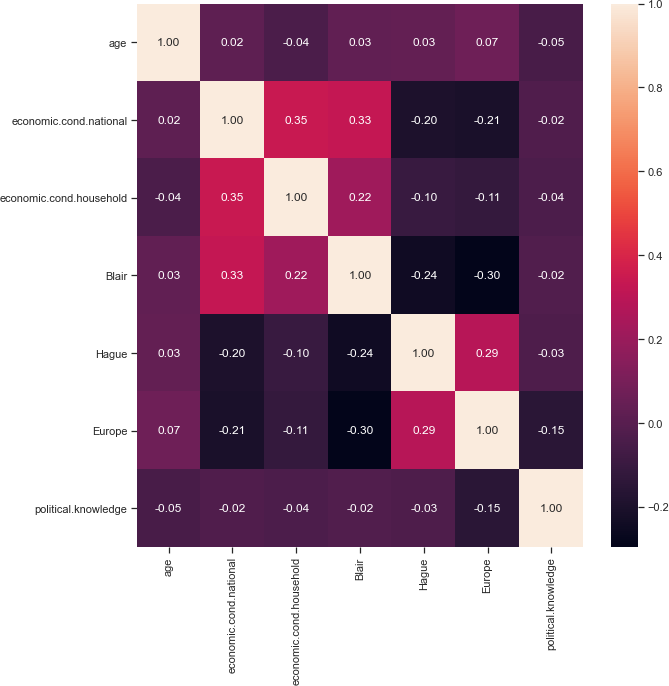
*Figure 23 Political Knowledge VS Vote*

A very close and similar pattern is seen between both Labour and Conservative Party in terms of Political Knowledge.

We can further draw the Pairplot and Heatmap to study the correlation between the various columns in the given Data set.



*Figure 24 Pairplot of the Election Dataset*

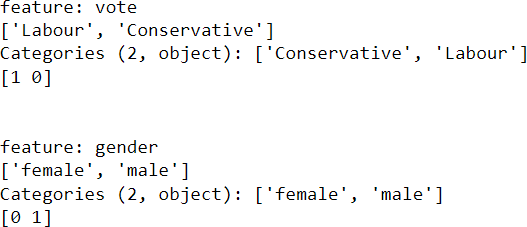


*Figure 25 Heat Map on the Dataset*

We can understand that there is very low correlation between the various columns in the given Dataset.

# 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).

In order to effectively build machine learning models, we would need to encode the data.

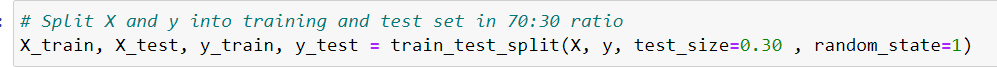


*Figure 26 Encoding the Categorical Variables*

We can understand from the given dataset that there are outliers that would need to be treated, further we can understand that the Age category is of distinctly higher values in comparison to the rest of the columns.

We can scale the data to further improve the modelling that is to be done using the StandardScalar() within Python.

We can further split the Train-Test Data



# Apply Logistic Regression and LDA (linear discriminant analysis).

In order to understand the best Model, we would need to consider multiple models that can be tuned. We can apply the Tuning methods along with the basic methods to further understand how the model will perform under different circumstances.

The python code for the same can be referenced within the ipynb file. The following results have been examined.

**Logistic Regression**

**LDA Model**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **LDA - Class 0** | **Basic Model** | **Gridsearch CV** | **SMOTE Model** | **Train Data** |  | **LDA - Class 0** | **Basic Model** | **Gridsearch CV** | **SMOTE Model** | **Test Data** |
| **Precision** | 0.76 | 0.76 | 0.83 | **Precision** | 0.69 | 0.69 | 0.61 |
| **Recall** | 0.71 | 0.72 | 0.82 | **Recall** | 0.66 | 0.68 | 0.78 |
| **F1 Score** | 0.73 | 0.74 | 0.82 | **F1 Score** | 0.67 | 0.69 | 0.69 |
| **Model Accuracy** | 0.84 | 0.84 | 0.82 | **Model Accuracy** | 0.82 | 0.82 | 0.8 |
|  |  |  |  |  |  |  |  |  |  |  |
| **LDA - Class 1** | **Basic Model** | **Gridsearch CV** | **SMOTE Model** | **Train Data** |  | **LDA - Class 1** | **Basic Model** | **Gridsearch CV** | **SMOTE Model** | **Test Data** |
| **Precision** | 0.87 | 0.88 | 0.82 | **Precision** | 0.87 | 0.88 | 0.9 |
| **Recall** | 0.9 | 0.9 | 0.83 | **Recall** | 0.88 | 0.88 | 0.8 |
| **F1 Score** | 0.89 | 0.89 | 0.82 | **F1 Score** | 0.87 | 0.88 | 0.85 |
| **Model Accuracy** | 0.84 | 0.84 | 0.82 | **Model Accuracy** | 0.82 | 0.82 | 0.8 |

**Inference**

Both the Logistic and LDA models have performed well on both the training and testing data.

* From the given results we can see that the Basic model and the GridsearchCV tuned model are very close to each other in results.
* The SMOTE model is slightly overfitting in terms of precision, recall and F1 score for both class 0 and class 1 in Logistic and LDA models.
* A wide range of parameters have been employed to best select the values to be selected within GridsearchCV. The values however have brought the results closely similar to the basic model as well.
* The LDA model works at 82% and 80% accuracy for train and test data respectively.
* The LR model works at 83% and 80% accuracy for train and test data respectively.

# Apply KNN Model and Naïve Bayes Model. Interpret the results.

**KNN Model**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **KNN - Class 0** | **Basic Model** | **Gridsearch CV** | **SMOTE Model** | **Train Data** |  | **KNN - Class 0** | **Basic Model** | **Gridsearch CV** | **SMOTE Model** | **Test Data** |
| **Precision** | 0.75 | 0.75 | 0.84 |  | **Precision** | 0.67 | 0.7 | 0.59 |
| **Recall** | 0.72 | 0.71 | 0.94 |  | **Recall** | 0.7 | 0.71 | 0.81 |
| **F1 Score** | 0.74 | 0.73 | 0.89 |  | **F1 Score** | 0.69 | 0.7 | 0.68 |
| **Model Accuracy** | 0.84 | 0.84 | 0.88 |  | **Model Accuracy** | 0.82 | 0.83 | 0.78 |
|  |  |  |  |  |  |  |  |  |  |  |
| **KNN - Class 1** | **Basic Model** | **Gridsearch CV** | **SMOTE Model** | **Train Data** |  | **KNN - Class 1** | **Basic Model** | **Gridsearch CV** | **SMOTE Model** | **Test Data** |
| **Precision** | 0.88 | 0.87 | 0.93 |  | **Precision** | 0.88 | 0.88 | 0.91 |
| **Recall** | 0.89 | 0.89 | 0.83 |  | **Recall** | 0.87 | 0.88 | 0.77 |
| **F1 Score** | 0.88 | 0.88 | 0.88 |  | **F1 Score** | 0.87 | 0.88 | 0.84 |
| **Model Accuracy** | 0.84 | 0.84 | 0.88 |  | **Model Accuracy** | 0.82 | 0.83 | 0.78 |

**Naïve Bayes Model**

**Inference**

* The results of the KNN and the Naïve Bayes model shows both these models are effectively perform upon both the Train and Test data.
* The basic model and the GridsearchCV model shows similarity in the output.
* The SMOTE model has improved both KNN and Naïve Bayes model.
* Between both the models the SMOTE model of KNN model can be recommended as the Train and Test have values that are high and close to each other. They hold higher Recall values.
* The KNN has been explored to understand that K-neighbours of 16 would improve the accuracy of the model.
* The KNN model works at 83% and 82% accuracy for train and test data respectively.
* The GNB model works at 83% and 82% accuracy for train and test data respectively.

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

We can further examine Boosting and Bagging methods by applying them unto the Election Dataset.

**Ada Boosting**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ADA - Class 0** | **Basic Model** | **SMOTE Model** | **Train Data** |  | **ADA - Class 0** | **Basic Model** | **SMOTE Model** | **Test Data** |
| **Precision** | 0.78 | 0.86 |  | **Precision** | 0.68 | 0.63 |
| **Recall** | 0.72 | 0.87 |  | **Recall** | 0.69 | 0.75 |
| **F1 Score** | 0.74 | 0.87 |  | **F1 Score** | 0.68 | 0.69 |
| **Model Accuracy** | 0.85 | 0.87 |  | **Model Accuracy** | 0.82 | 0.8 |
|  |  |  |  |  |  |  |  |  |
| **ADA - Class 1** | **Basic Model** | **SMOTE Model** | **Train Data** |  | **ADA - Class 1** | **Basic Model** | **SMOTE Model** | **Test Data** |
| **Precision** | 0.88 | 0.87 |  | **Precision** | 0.88 | 0.89 |
| **Recall** | 0.91 | 0.86 |  | **Recall** | 0.87 | 0.82 |
| **F1 Score** | 0.89 | 0.86 |  | **F1 Score** | 0.87 | 0.86 |
| **Model Accuracy** | 0.85 | 0.87 |  | **Model Accuracy** | 0.82 | 0.8 |

**Gradient Boosting**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Gradient - Class 0** | **Basic Model** | **SMOTE Model** | **Train Data** |  | **Gradient - Class 0** | **Basic Model** | **SMOTE Model** | **Test Data** |
| **Precision** | 0.84 | 0.9 | **Precision** | 0.69 | 0.63 |
| **Recall** | 0.79 | 0.91 | **Recall** | 0.74 | 0.75 |
| **F1 Score** | 0.81 | 0.9 | **F1 Score** | 0.71 | 0.69 |
| **Model Accuracy** | 0.89 | 0.9 | **Model Accuracy** | 0.83 | 0.81 |
|  |  |  |  |  |  |  |  |
| **Gradient - Class 1** | **Basic Model** | **SMOTE Model** | **Train Data** | **Gradient - Class 1** | **Basic Model** | **SMOTE Model** | **Test Data** |
| **Precision** | 0.91 | 0.91 | **Precision** | 0.89 | 0.89 |
| **Recall** | 0.93 | 0.9 | **Recall** | 0.87 | 0.83 |
| **F1 Score** | 0.92 | 0.9 | **F1 Score** | 0.88 | 0.86 |
| **Model Accuracy** | 0.89 | 0.9 | **Model Accuracy** | 0.83 | 0.81 |

* There Basic model of both Gradient and Ada Boosting performs well on both the train and test data
* The SMOTE model has improved the training data, but in comparison to the test data we can see that there is a level of overfitting in the model.
* The Adaboost Basic model has shown good application on both the train and test data.
* The Gradient boosting model shows high and accurate values for Class 1 data in the SMOTE model.

**Random Forest**

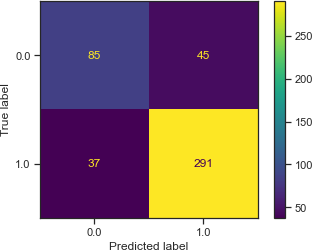
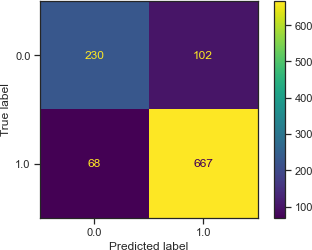
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RF - Class 0** | **Basic Model** | **SMOTE Model** | **Train Data** |  | **RF - Class 0** | **Basic Model** | **SMOTE Model** | **Test Data** |
| **Precision** | 1 | 1 | **Precision** | 0.67 | 0.66 |
| **Recall** | 1 | 1 | **Recall** | 0.7 | 0.73 |
| **F1 Score** | 1 | 1 | **F1 Score** | 0.69 | 0.7 |
| **Model Accuracy** | 1 | 1 | **Model Accuracy** | 0.82 | 0.82 |
|  |  |  |  |  |  |  |  |
| **RF - Class 1** | **Basic Model** | **SMOTE Model** | **Train Data** | **RF - Class 1** | **Basic Model** | **SMOTE Model** | **Test Data** |
| **Precision** | 1 | 1 | **Precision** | 0.88 | 0.89 |
| **Recall** | 1 | 1 | **Recall** | 0.87 | 0.85 |
| **F1 Score** | 1 | 1 | **F1 Score** | 0.87 | 0.87 |
| **Model Accuracy** | 1 | 1 | **Model Accuracy** | 0.82 | 0.82 |

**Bagging**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bagging - Class 0** | **Basic Model** | **SMOTE Model** | **Train Data** |  | **Bagging - Class 0** | **Basic Model** | **SMOTE Model** | **Test Data** |
| **Precision** | 1 | 1 | **Precision** | 0.64 | 0.64 |
| **Recall** | 1 | 1 | **Recall** | 0.65 | 0.67 |
| **F1 Score** | 1 | 1 | **F1 Score** | 0.64 | 0.65 |
| **Model Accuracy** | 1 | 1 | **Model Accuracy** | 0.8 | 0.8 |
|  |  |  |  |  |  |  |  |
| **Bagging - Class 1** | **Basic Model** | **SMOTE Model** | **Train Data** | **Bagging - Class 1** | **Basic Model** | **SMOTE Model** | **Test Data** |
| **Precision** | 1 | 1 | **Precision** | 0.86 | 0.87 |
| **Recall** | 1 | 1 | **Recall** | 0.86 | 0.85 |
| **F1 Score** | 1 | 1 | **F1 Score** | 0.86 | 0.86 |
| **Model Accuracy** | 1 | 1 | **Model Accuracy** | 0.8 | 0.8 |

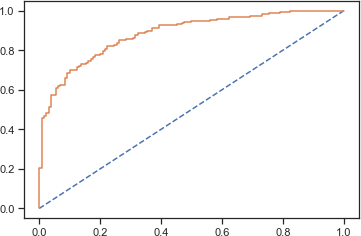
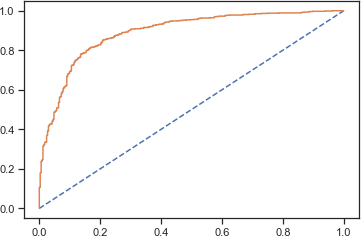
* For both the Random Forest and Bagging method, the model shows a large amount of overfitting
* The model cannot be used to study the given dataset as the results would be erroneous in nature.
  1. **Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized.**

**Logistic Regression**

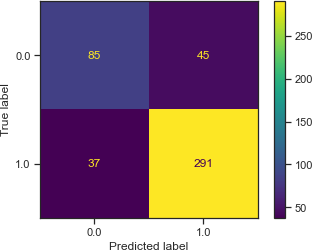
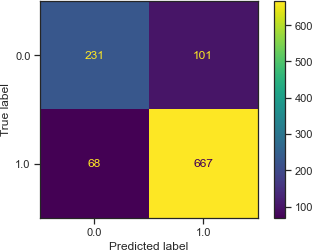


*Figure 27 LR Train and Test Data Confusion Matrix*

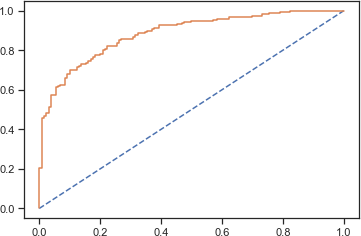
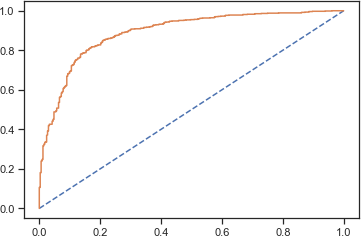
LR AUC Train Data – 0.89 LR AUC Test Data -0.89



Logistic Regression with GridsearchCV



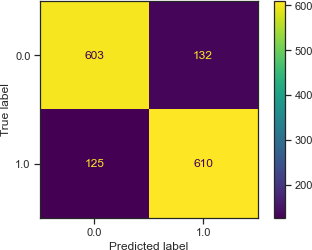
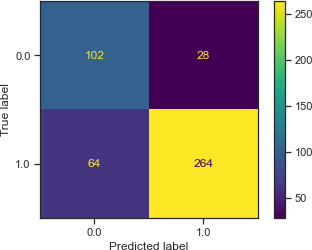
*Figure 28 LR GridsearchCV Train and Test Data Confusion Matrix*



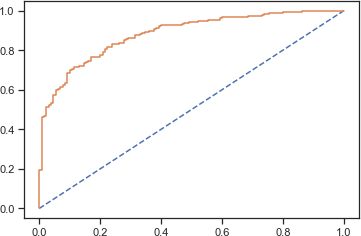
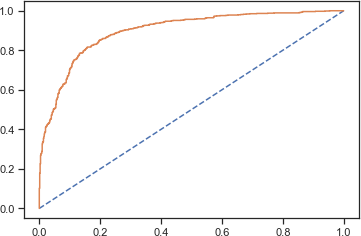
*Figure 30 LR GridsearchCV Train Data ACU – 0.89*

**Logistic Regression with SMOTE**

*Figure 29 LR GridsearchCV Test Data ACU – 0.89*

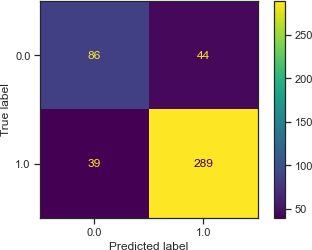
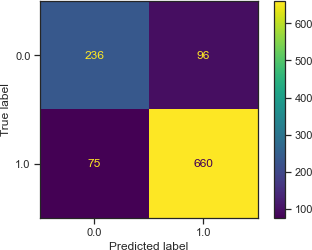
*Figure 31 LR SMOTE Train and Test Data Confusion Matrix*



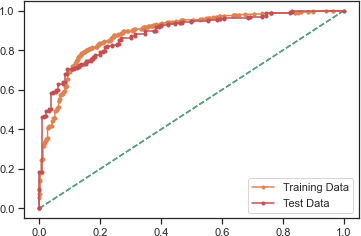
*Figure 33 LR SMOTE Train Data ACU – 0.897*

*Figure 32 LR SMOTE Test Data ACU – 0.890*

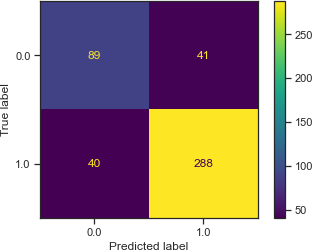
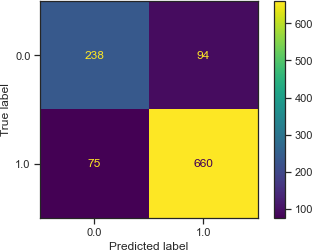
**LDA Model**



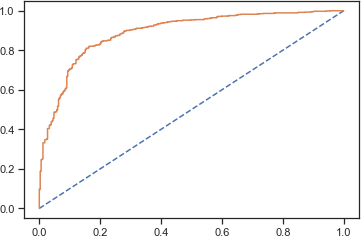
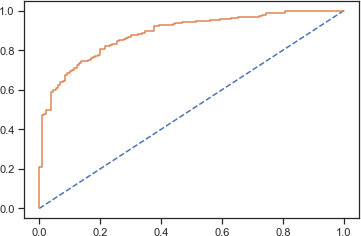
*Figure 34 LDA Train and Test Set*



*Figure 35 AUC for the Training Data: 0.889 / AUC for Test Data 0.884*



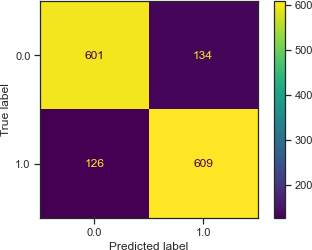
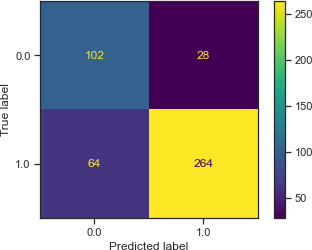
*Figure 36 LDA Gridsearch CV Train and Test Matrix*

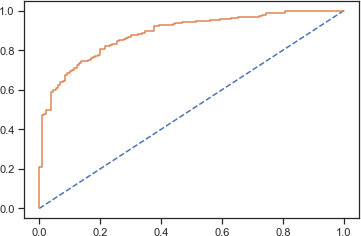
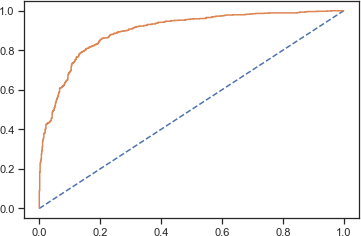
*Figure 38 AUC for LDA Gridsearch CV Train - 0.8904*

**LDA Model with SMOTE**

*Figure 37 AUC for LDA Gridsearch CV Test - 0.8904*

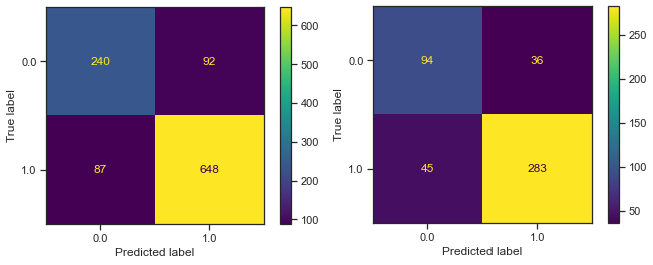
*Figure 39 LDA SMOTE Train and Test Matrix*



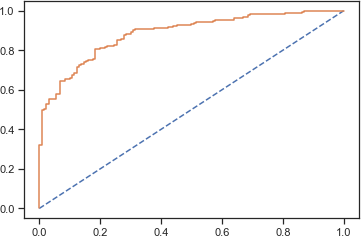
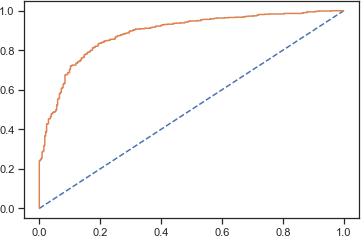
*Figure 41 AUC for LDA SMOTE CV Train - 0.897*

*Figure 40 AUC for LDA SMOTE Test - 0.897*

**Naïve Bayes**



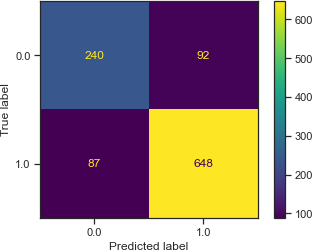
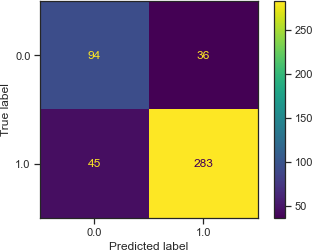
*Figure 42 Naive Bayes Train and Test Matrix*



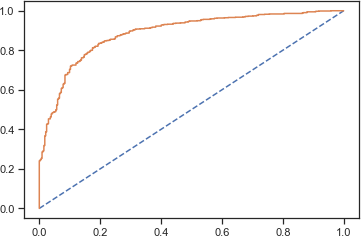
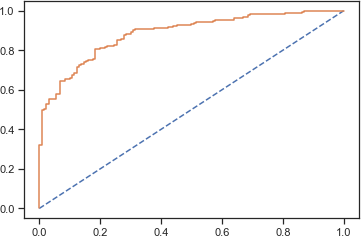
*Figure 44 AUC Naive Bayes Train Data 0.887*

**Naïve Bayes with GridsearchCV**

*Figure 43 AUC Naive Bayes Test Data 0.887*

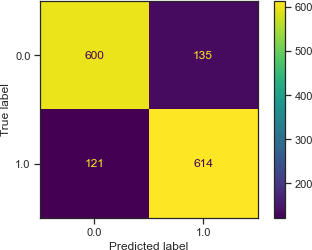
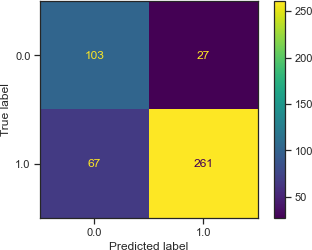
*Figure 45 Naive Bayes GridsearchCV Train and Test Matrix*

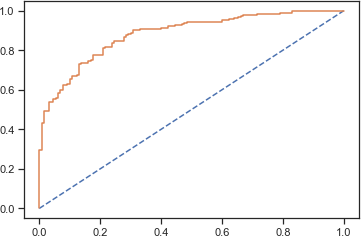
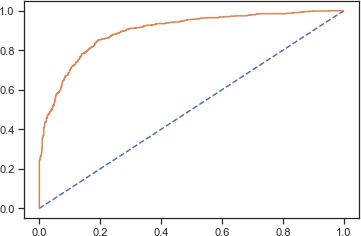
*Figure 47 AUC Naive Bayes GridsearchCV Train Data 0.887*

**Naïve Bayes with SMOTE**

*Figure 46 AUC Naive Bayes GridsearchCV Test Data 0.887*

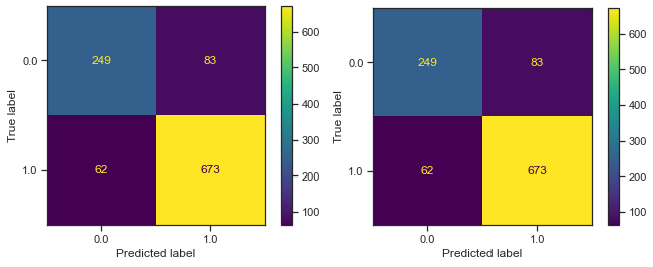
*Figure 48 Naive Bayes SMOTE Train and Test Matrix*



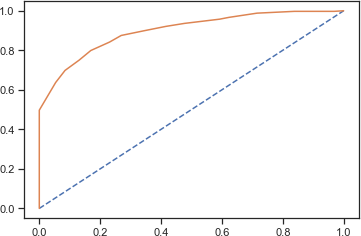
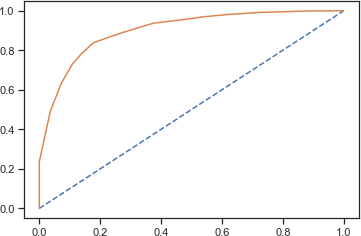
*Figure 49 AUC Naive Bayes SMOTE Test Data 0.887*

*Figure 50 AUC Naive Bayes SMOTE Train Data 0.897*

**KNN**

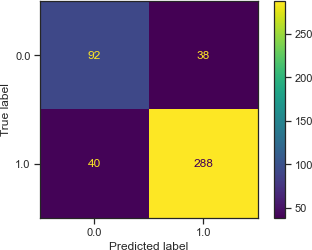
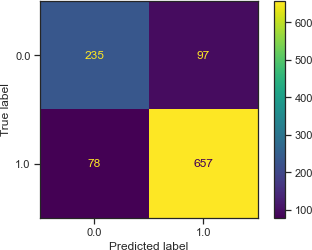


*Figure 51 KNN Train and Test Matrix*

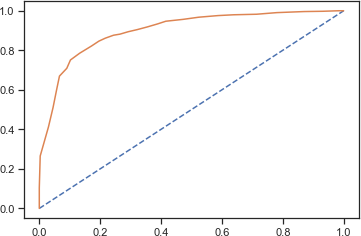
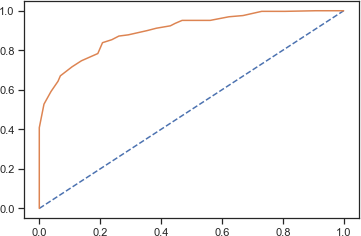


*Figure 52 AUC Train Data 0.901 Figure 53 AUC for KNN Test Data 0.901*

**KNN GridsearchCV**



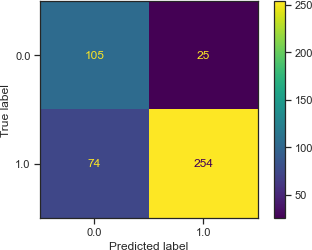
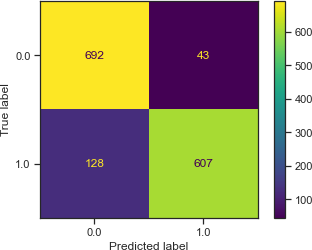
*Figure 54 KNN Gridsearch CV Train and Test Confusion Matrix*

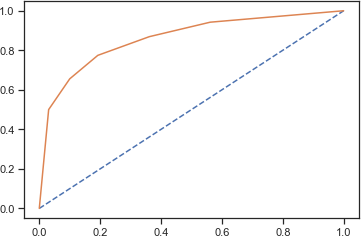
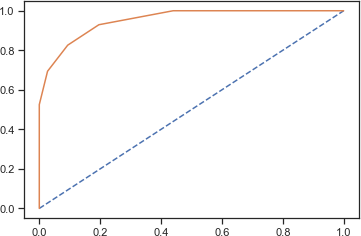
*Figure 55 AUC KNN GridsearchCV Test Data 0.900*

*Figure 56 AUC KNN Gridsearchcv Train Data - 0.900*

**KNN SMOTE**

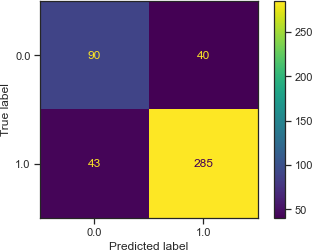
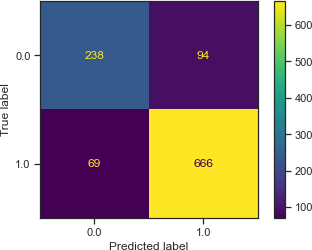


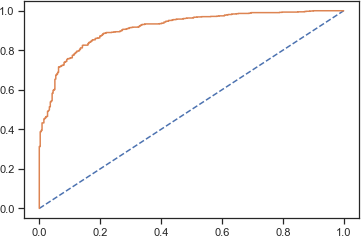
*Figure 57 Train and Test Confusion Matrix for KNN SMOTE*

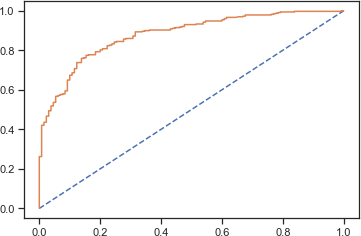


*Figure 58 AUC KNN SMOTE Training Data .966 Figure 59 AUC KNN Test data 0.952*

**ADA Boost**



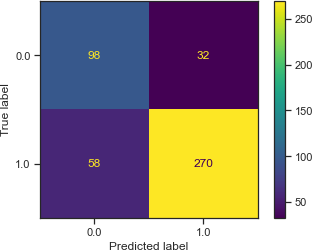
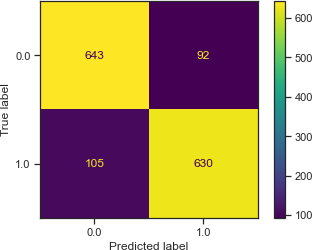
*Figure 60 Train and Test Confusion Matrix of AdaBoost*



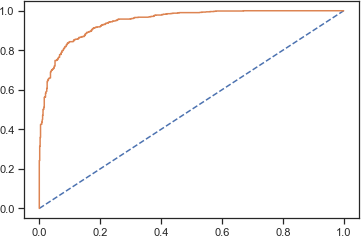
*Figure 61 AUC Ada Boost Test Data .913*

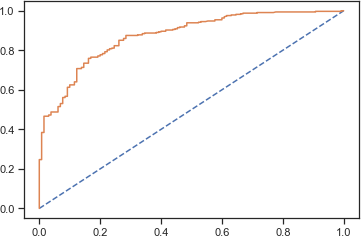
*Figure 62 AUC Ada Boost Training Data .913*

**ADA Boost SMOTE**



*Figure 63 60 Train and Test Confusion Matrix of AdaBoost SMOTE*

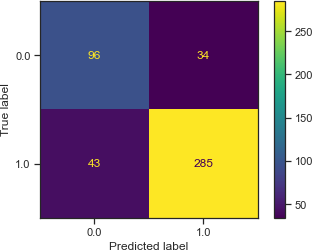
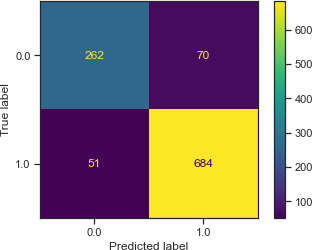




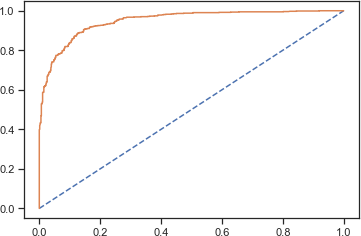
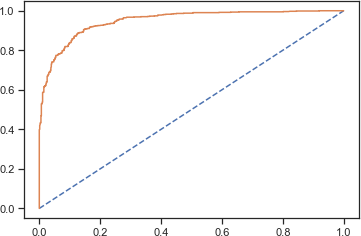
*Figure 64 AUC Ada Boost SMOTE Test Data .906*

*Figure 65 AUC Ada boost SMOTE Train Data .946*

**Gradient Boost**



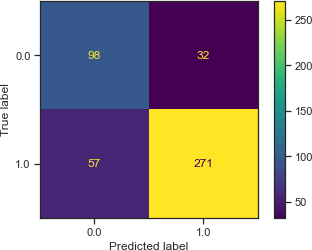
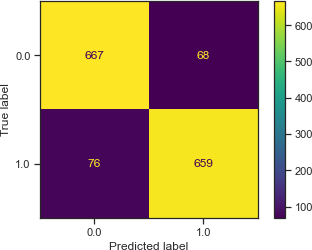
*Figure 66 Train and Test Confusion Matrix of Gradient Boost*



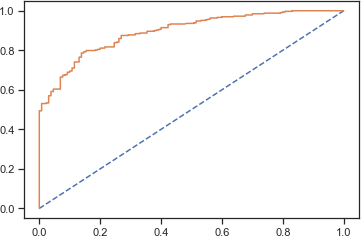
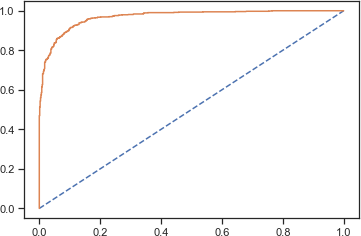
*Figure 68 AUC Train Data Gradient Boost - .949*

*Figure 67 AUC Test Data Gradient Boost 0.946*

**Gradient Boost with SMOTE**



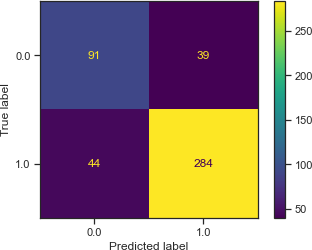
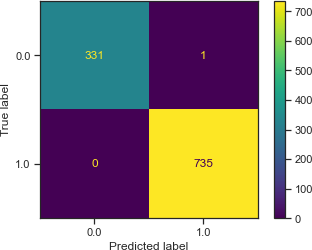
*Figure 69 Train and Test Confusion Matrix of Gradient Boost with SMOTE*



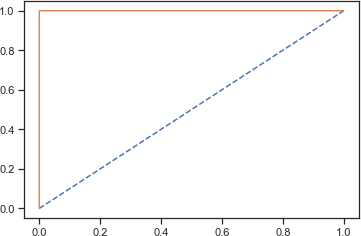
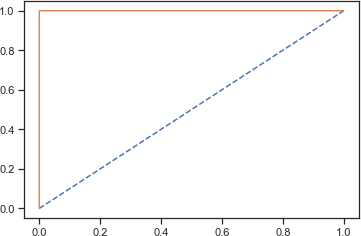
*Figure 71 AUC Train Data Gradient Boost SMOTE - .969*

*Figure 70 AUC Test Data Gradient Boost SMOTE - .969*

**Random Forest**

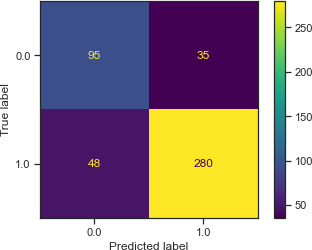
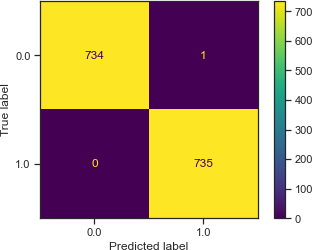


*Figure 72 Train and Test Confusion Matrix of Random Forest*

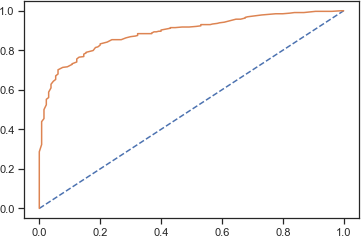
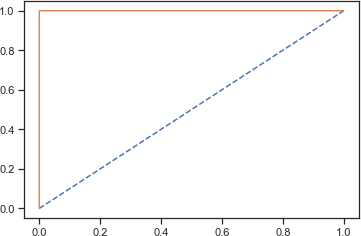
 

*Figure 73 AUC Test and Train Data for Random Forest = 1*

**Since the model is overfitting applying the SMOTE model continues to overfit the model.**

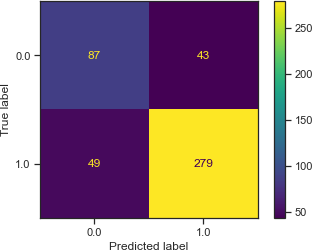
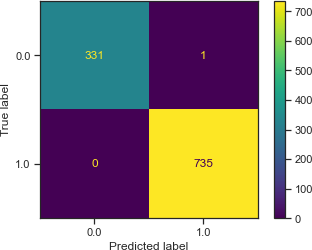


*Figure 74 Train and Test Confusion Matrix of Random Forest SMOTE*

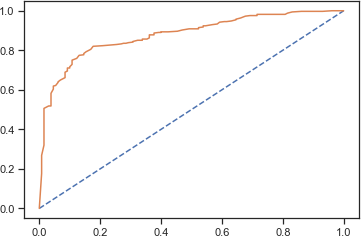
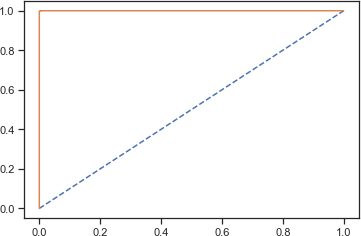


*Figure 75 AUC Test and Train Data for Random Forest with SMOTE = 1*

**Bagging**



*Figure 76 Train and Test Confusion Matrix of Bagging model*



*Figure 77 Test and Train Data for Bagging 1.0*

# Based on these predictions, what are the insights? (5 marks)

* From the given insights we can understand that most of the models perform well on the given dataset.
* However we can see from the Model score, accuracy and AUC that the Gradient Boost model work well for both the train and testing data.
* The model can be adopted to undertaken and make predictions for class 1.
* The Logit and LDA model has performed well for
* Bagging and Random Forest had performed poorly by overfitting the data in the Election Dataset
* We can consider even fine tuning the models to improve the accuracy of both the test and train data.
* Most models bring about an accuracy of predicting values of above 80%. All models effectively can work as long as overfitting ones are not undertaken to take any decisions.
* The labour party has voters of a younger age, proper information needs to be campaigned in order to influence the voters of the conservative party.
* The integration of European regulations has a fair chance of reducing votes, proper measures must be undertaken to educate the voters to move in the correct direction. Labour party voters may choose to not vote for the same if European integration is not educated effectively.

**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**

**(Hint: use .words(), .raw(), .sent() for extracting counts)**

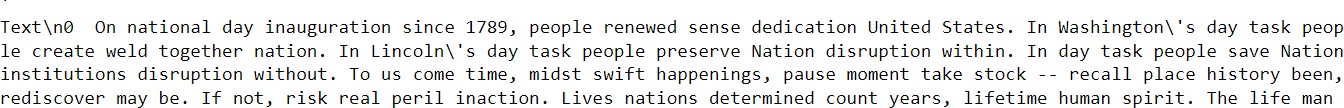
* 1. **Find the number of characters, words, and sentences for the mentioned documents. Characters**

Number of Characters in the President Roosevelt Speech 7571 Number of Characters in the President Kennedy Speech 7618 Number of Characters in the President Nixon Speech 9991 **Words**

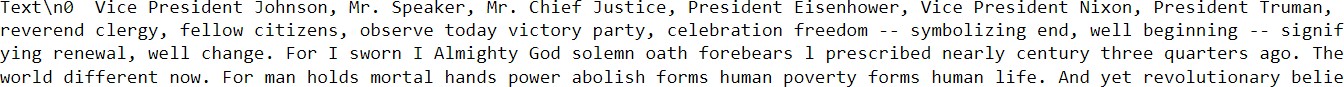
Number of words in the Roosvelt Speech is : 1360 Number of words in the Kennedy Speech is : 1390 Number of words in the Nixon Speech is : 1819 **Sentences**

Number of Sentences in the President Roosevelt Speech 67 Number of Sentences in the President Kennedy Speech 52 Number of Sentences in the President Nixon Speech 68

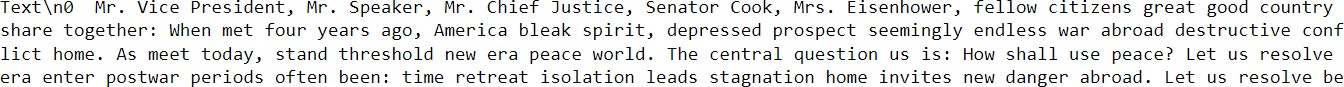
* 1. **Remove all the stopwords from all three speeches. Removing Stop words from President Roosvelt Speech**



**Removing Stop words from President Kennedy Speech**



**Removing Stop words from President Nixon Speech**



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

**President Roosevelt**

|  |  |
| --- | --- |
| **Word** | **Frequency** |
| Nation | 12 |
| Know | 10 |
| Spirit | 9 |

**President Kennedy**

|  |  |
| --- | --- |
| **Word** | **Frequency** |
| Let | 16 |
| Us | 12 |
| World | 8 |

**President Nixon**

|  |  |
| --- | --- |
| **Word** | **Frequency** |
| Us | 26 |
| Let | 22 |
| America | 21 |

# Plot the word cloud of each of the speeches of the variable. (after removing the stopwords) – 3 Marks [ refer to the End-to-End Case Study done in the Mentored Learning Session ]

President Roosevelt Word Cloud



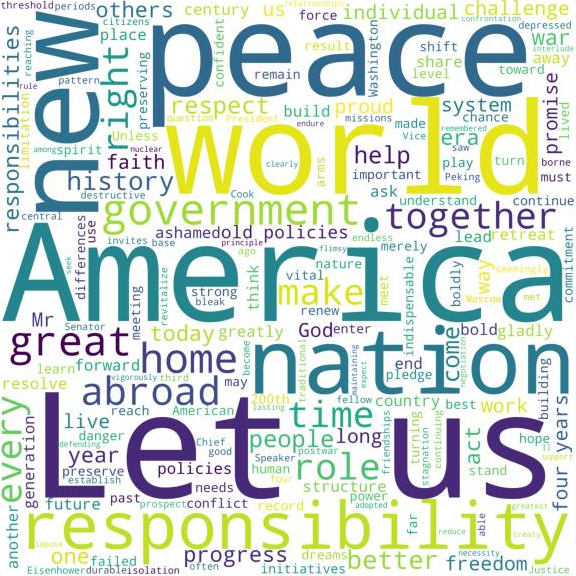
*Figure 78 President Roosevelt Word Cloud*

President Kennedy Word Cloud



*Figure 79 President Kennedy Word Cloud*

President Nixon Word Cloud



*Figure 80 President Nixon Word Cloud*

[Figure 1 Dataframe of Election Data 3](#_bookmark1)

[Figure 2 Exploring Missing values 4](#_bookmark2)

[Figure 3 Information of the Dataset 4](#_bookmark3)

[Figure 4 Duplicate Values within the Dataset 5](#_bookmark4)

[Figure 5 Boxplot of the Dataset 5](#_bookmark5)

[Figure 6 Skewness of Dataframe 6](#_bookmark6)

[Figure 7 Distribution and Boxplot of Age 6](#_bookmark8)

[Figure 8 Distribution and Boxplot of Economic Conditional National 7](#_bookmark9)

[Figure 9 Distribution and Boxplot of Economic Conditional Household 7](#_bookmark10)

[Figure 10 Distribution and Boxplot of Blair 8](#_bookmark11)

[Figure 11 Distribution and Boxplot of Hague 8](#_bookmark12)

[Figure 12 Distribution and Boxplot of Europe 9](#_bookmark13)

[Figure 13 Distribution and Boxplot of Political Knowledge 9](#_bookmark14)

[Figure 14 Countplot of Vote 10](#_bookmark15)

[Figure 15 Countplot of Gender 10](#_bookmark16)

[Figure 16 Economic Household VS Europe 11](#_bookmark17)

[Figure 17 Economic Conditional National VS Europe 11](#_bookmark18)

[Figure 18 Political Knowledge VS Europe 12](#_bookmark19)

[Figure 19 Age VS Vote 13](#_bookmark20)

[Figure 20 Economic Condition National VS Vote 13](#_bookmark21)

[Figure 21 Economic Household VS Vote 13](#_bookmark22)

[Figure 22 Europe VS Vote 14](#_bookmark23)

[Figure 23 Political Knowledge VS Vote 14](#_bookmark24)

[Figure 24 Pairplot of the Election Dataset 15](#_bookmark25)

[Figure 25 Heat Map on the Dataset 16](#_bookmark26)

[Figure 26 Encoding the Categorical Variables 16](#_bookmark28)

[Figure 27 LR Train and Test Data Confusion Matrix 20](#_bookmark32)

[Figure 28 LR GridsearchCV Train and Test Data Confusion Matrix 21](#_bookmark33)

Figure 29 LR GridsearchCV Test Data ACU – 0.89 21

[Figure 30 LR GridsearchCV Train Data ACU – 0.89 21](#_bookmark34)

[Figure 31 LR SMOTE Train and Test Data Confusion Matrix 21](#_bookmark35)

Figure 32 LR SMOTE Test Data ACU – 0.890 21

[Figure 33 LR SMOTE Train Data ACU – 0.897 21](#_bookmark36)

[Figure 34 LDA Train and Test Set 22](#_bookmark37)

[Figure 35 AUC for the Training Data: 0.889 / AUC for Test Data 0.884 22](#_bookmark38)

[Figure 36 LDA Gridsearch CV Train and Test Matrix 22](#_bookmark39)

Figure 37 AUC for LDA Gridsearch CV Test - 0.8904 23

[Figure 38 AUC for LDA Gridsearch CV Train - 0.8904 23](#_bookmark40)

[Figure 39 LDA SMOTE Train and Test Matrix 23](#_bookmark41)

Figure 40 AUC for LDA SMOTE Test - 0.897 23

[Figure 41 AUC for LDA SMOTE CV Train - 0.897 23](#_bookmark42)

[Figure 42 Naive Bayes Train and Test Matrix 24](#_bookmark43)

Figure 43 AUC Naive Bayes Test Data 0.887 24

[Figure 44 AUC Naive Bayes Train Data 0.887 24](#_bookmark44)

[Figure 45 Naive Bayes GridsearchCV Train and Test Matrix 24](#_bookmark45)

[Figure 47 AUC Naive Bayes GridsearchCV Train Data 0.887 25](#_bookmark46)

Figure 46 AUC Naive Bayes GridsearchCV Test Data 0.887 25

[Figure 48 Naive Bayes SMOTE Train and Test Matrix 25](#_bookmark47)

Figure 49 AUC Naive Bayes SMOTE Test Data 0.887 25

[Figure 50 AUC Naive Bayes SMOTE Train Data 0.897 25](#_bookmark48)

[Figure 51 KNN Train and Test Matrix 26](#_bookmark49)

[Figure 52 AUC Train Data 0.901 Figure 53 AUC for KNN Test Data 0.901 26](#_bookmark50)

[Figure 54 KNN Gridsearch CV Train and Test Confusion Matrix 26](#_bookmark51)

Figure 55 AUC KNN GridsearchCV Test Data 0.900 27

[Figure 56 AUC KNN Gridsearchcv Train Data - 0.900 27](#_bookmark52)

[Figure 57 Train and Test Confusion Matrix for KNN SMOTE 27](#_bookmark53)

[Figure 58 AUC KNN SMOTE Training Data .966 Figure 59 AUC KNN Test data 0.952 27](#_bookmark54)

[Figure 60 Train and Test Confusion Matrix of AdaBoost 28](#_bookmark55)

[Figure 61 AUC Ada Boost Test Data .913 28](#_bookmark56)

[Figure 62 AUC Ada Boost Training Data .913 28](#_bookmark57)

[Figure 63 60 Train and Test Confusion Matrix of AdaBoost SMOTE 28](#_bookmark58)

Figure 64 AUC Ada Boost SMOTE Test Data .906 29

[Figure 65 AUC Ada boost SMOTE Train Data .946 29](#_bookmark59)

[Figure 66 Train and Test Confusion Matrix of Gradient Boost 29](#_bookmark60)

Figure 67 AUC Test Data Gradient Boost 0.946 29

[Figure 68 AUC Train Data Gradient Boost - .949 29](#_bookmark61)

[Figure 69 Train and Test Confusion Matrix of Gradient Boost with SMOTE 30](#_bookmark62)

Figure 70 AUC Test Data Gradient Boost SMOTE - .969 30

[Figure 71 AUC Train Data Gradient Boost SMOTE - .969 30](#_bookmark63)

[Figure 72 Train and Test Confusion Matrix of Random Forest 30](#_bookmark64)

[Figure 73 AUC Test and Train Data for Random Forest = 1 31](#_bookmark65)

[Figure 74 Train and Test Confusion Matrix of Random Forest SMOTE 31](#_bookmark66)

[Figure 75 AUC Test and Train Data for Random Forest with SMOTE = 1 31](#_bookmark67)

[Figure 76 Train and Test Confusion Matrix of Bagging model 32](#_bookmark68)

[Figure 77 Test and Train Data for Bagging 1.0 32](#_bookmark69)

[Figure 78 President Roosevelt Word Cloud 35](#_bookmark75)

[Figure 79 President Kennedy Word Cloud 35](#_bookmark76)

[Figure 80 President Nixon Word Cloud 36](#_bookmark77)