Project: Machine Learning

Name: Varsha Srinivasan

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

PROBLEM 1

Problem Statement

You are hired by one of the leading news channels CNBE who wants to analyze recent elections. This survey was conducted on 1525 voters with 9 variables. You 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.

Introduction

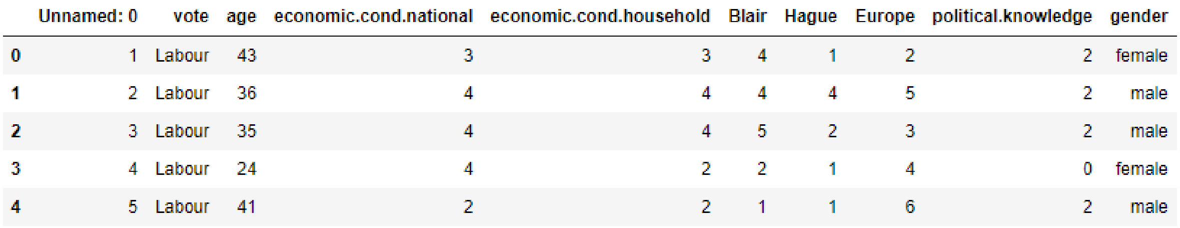
The purpose of this whole exercise is to perform exploratory data analysis and perform Classification using Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbor, Naïve Bayes, Bagging and Boosting. Also, to perform Model Tuning on them. The dataset consists of 1525 rows with their features like age, economic.cond.national, economic.cond.household, Hague, Europe, Blair, political.knowledge, gender and dependent variable vote. Insights are derived and recommendations are made.

Data Description

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

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

Sample of the dataset:



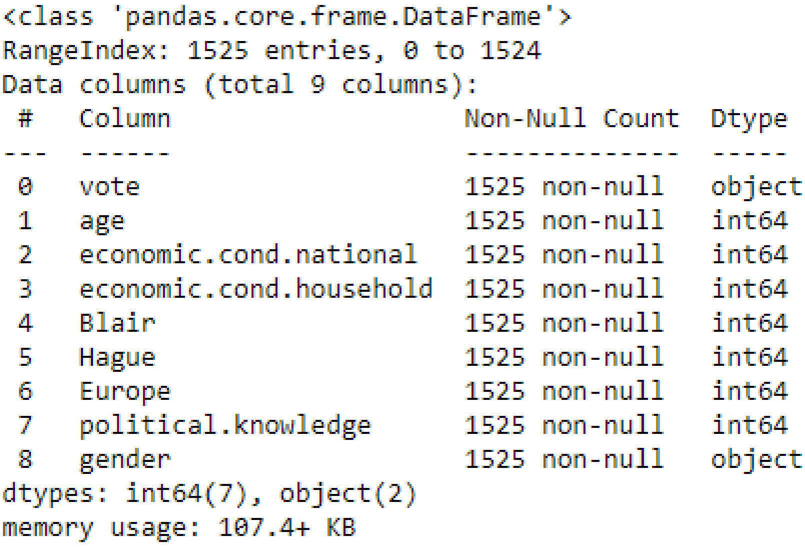
	Unnamed: 0	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	1	Labour	43	3	3	4	1	2	2	female
1	2	Labour	36	4	4	4	4	5	2	male
2	3	Labour	35	4	4	5	2	3	2	male
3	4	Labour	24	4	2	2	1	4	0	female
4	5	Labour	41	2	2	1	1	6	2	male

Table 1: Sample of the Dataset1

The data is read from the excel file and the above tables shows the first 5 rows of the dataset. The vote is the target variable. It is in object form hence Classification is done.

The 'Unnamed: 0' column isn't necessary. Hence, it is dropped.

Data Type and Missing Values

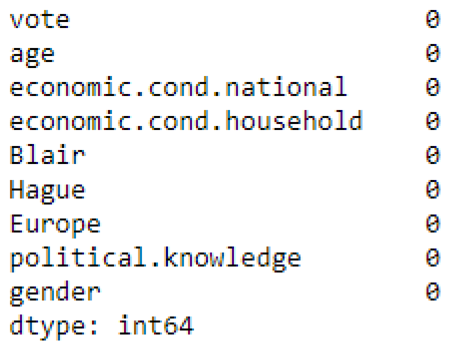


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   vote                     1525 non-null   object
 1   age                      1525 non-null   int64 
 2   economic.cond.national   1525 non-null   int64 
 3   economic.cond.household  1525 non-null   int64 
 4   Blair                    1525 non-null   int64 
 5   Hague                    1525 non-null   int64 
 6   Europe                   1525 non-null   int64 
 7   political.knowledge      1525 non-null   int64 
 8   gender                   1525 non-null   object
dtypes: int64(7), object(2)
memory usage: 107.4+ KB
```

Table 2: Dataset Info

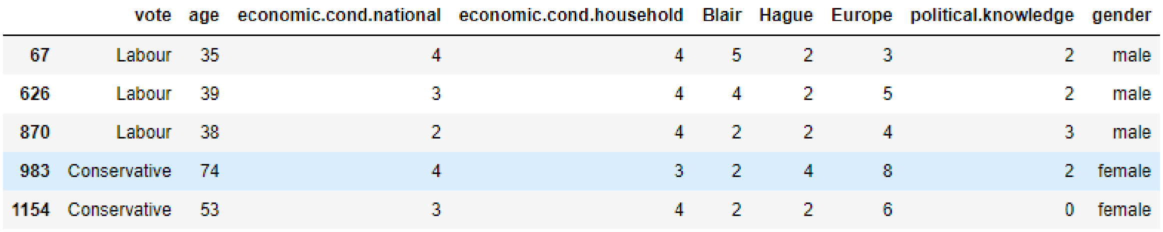
All the variables have numeric values except for vote and gender which has object data type. It has 1525 rows and 9 columns.

There is also no null values which can also be inferred from the output of isnull function.



```
vote                       0
age                        0
economic.cond.national     0
economic.cond.household    0
Blair                      0
Hague                      0
Europe                     0
political.knowledge        0
gender                     0
dtype: int64
```

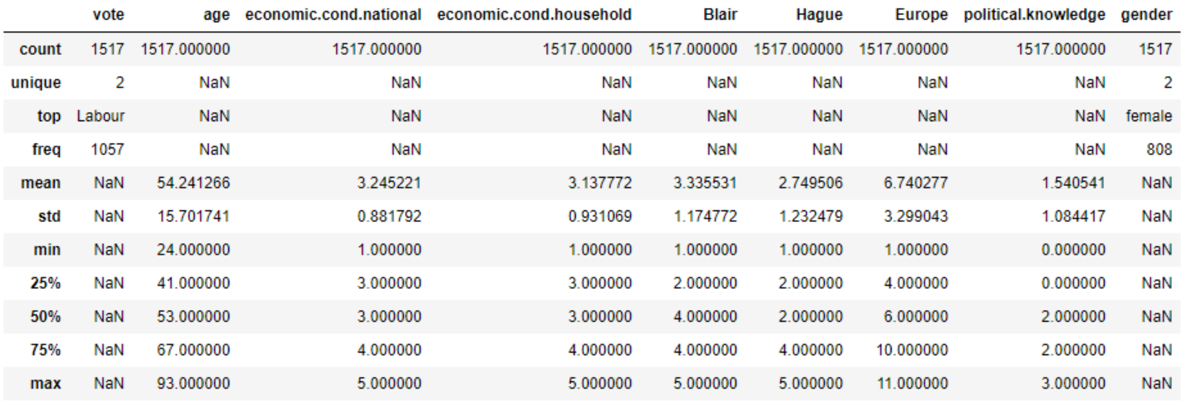
There are 8 duplicates in the dataset. Sample duplicates are:



	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
67	Labour	35	4	4	5	2	3	2	male
626	Labour	39	3	4	4	2	5	2	male
870	Labour	38	2	4	2	2	4	3	male
983	Conservative	74	4	3	2	4	8	2	female
1154	Conservative	53	3	4	2	2	6	0	female

Table 3: Duplicate Sample

The duplicates are removed using drop_duplicates function.



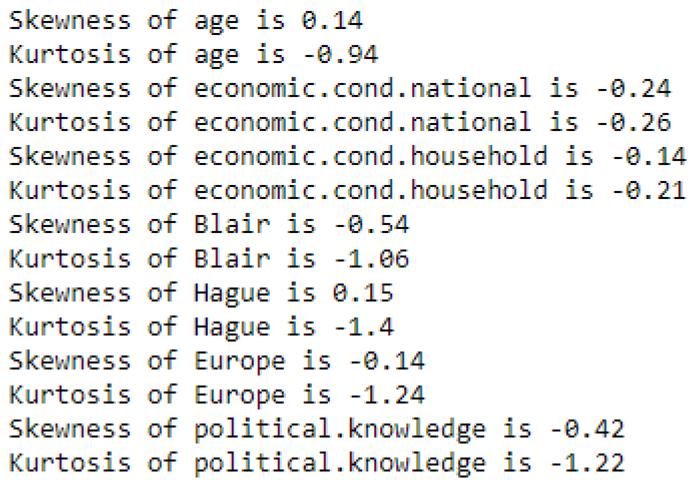
	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
count	1517	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517
unique	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2
top	Labour	NaN	NaN	NaN	NaN	NaN	NaN	NaN	female
freq	1057	NaN	NaN	NaN	NaN	NaN	NaN	NaN	808
mean	NaN	54.241266	3.245221	3.137772	3.335531	2.749506	6.740277	1.540541	NaN
std	NaN	15.701741	0.881792	0.931069	1.174772	1.232479	3.299043	1.084417	NaN
min	NaN	24.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	NaN
25%	NaN	41.000000	3.000000	3.000000	2.000000	2.000000	4.000000	0.000000	NaN
50%	NaN	53.000000	3.000000	3.000000	4.000000	2.000000	6.000000	2.000000	NaN
75%	NaN	67.000000	4.000000	4.000000	4.000000	4.000000	10.000000	2.000000	NaN
max	NaN	93.000000	5.000000	5.000000	5.000000	5.000000	11.000000	3.000000	NaN

Table 4:Description of Dataset

The total number of entries is 1517 after removing 8 duplicates in all features. All the variables have continuous values except for cut, color and clarity which has object data type. Age has a mean of around 54. Gender has female values the most. vote has the Labour values the most. economic.cond.national, economic.cond.household, Blair, Hague, Europe, political.knowledge has ordinal values.

Skewness and Kurtosis

Skewness essentially measures the symmetry of the distribution. In positively skewed, the mean of the data is greater than the median as a large number of data-pushed on the right-hand side. In negatively skewed, the mean of the data is less than the median as a large number of data-pushed on the left-hand.



```
Skewness of age is 0.14
Kurtosis of age is -0.94
Skewness of economic.cond.national is -0.24
Kurtosis of economic.cond.national is -0.26
Skewness of economic.cond.household is -0.14
Kurtosis of economic.cond.household is -0.21
Skewness of Blair is -0.54
Kurtosis of Blair is -1.06
Skewness of Hague is 0.15
Kurtosis of Hague is -1.4
Skewness of Europe is -0.14
Kurtosis of Europe is -1.24
Skewness of political.knowledge is -0.42
Kurtosis of political.knowledge is -1.22
```

If the skewness is between -0.5 & 0.5, the data are nearly symmetrical. If the skewness is between -1 & -0.5 (negative/left skewed) or between 0.5 & 1(positive/right skewed), the data are slightly skewed. If the skewness is lower than -1 (negative/left skewed) or greater than 1 (positive/right skewed), the data are extremely skewed.

All features are nearly symmetrically distributed.

Kurtosis refers to the degree of presence of outliers in the distribution. If kurtosis > 3, then it is called as Leptokurtic or heavy-tailed distribution as the kurtosis is more than normal distribution. If kurtosis = 3, then it is called as Mesokurtic as the kurtosis is same as the normal distribution. If kurtosis < 3, then it is called as Platykurtic or short-tailed distribution as the kurtosis is less than normal distribution.

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

Univariate Analysis

<u>Outliers Proportions</u>

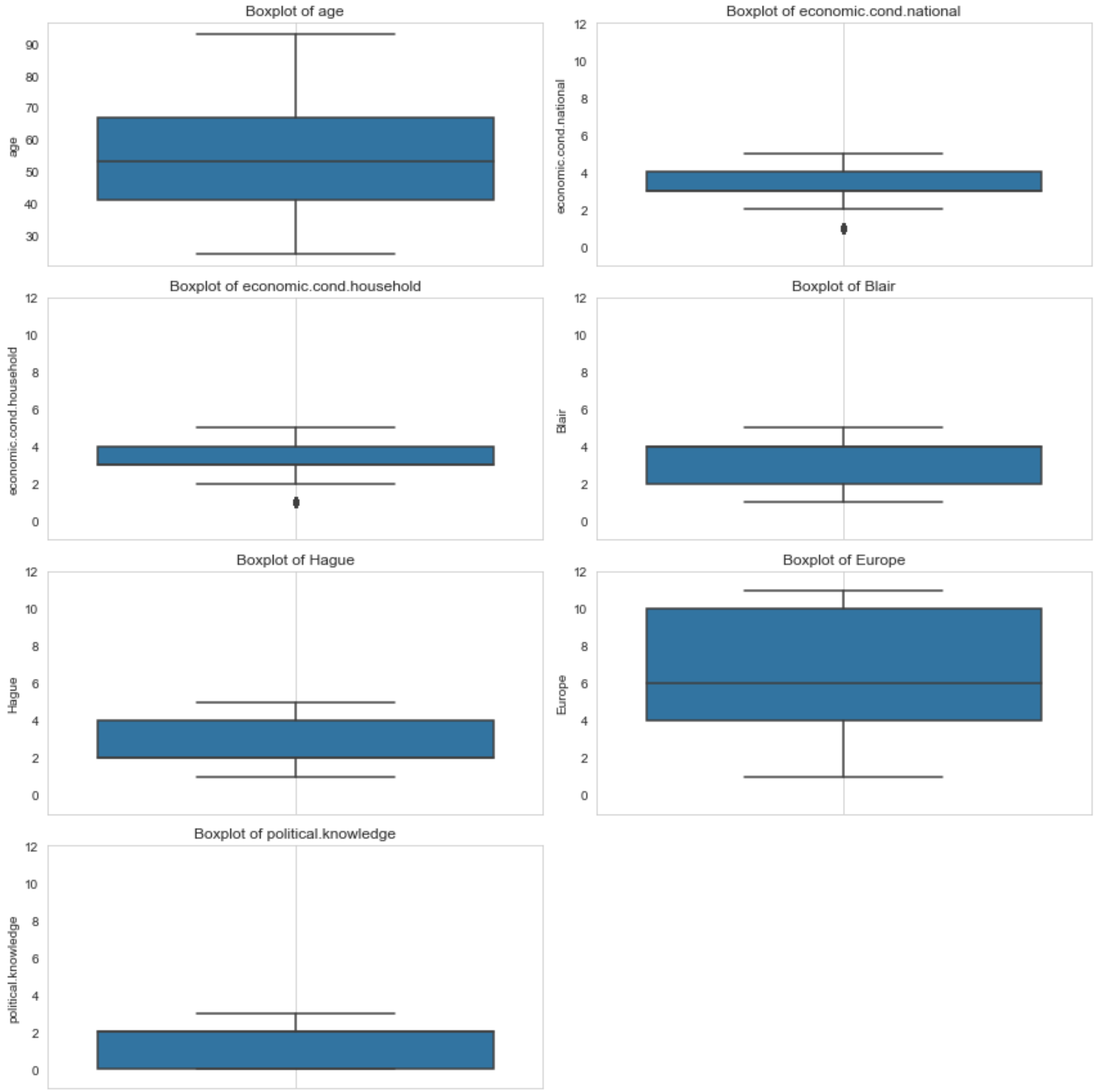
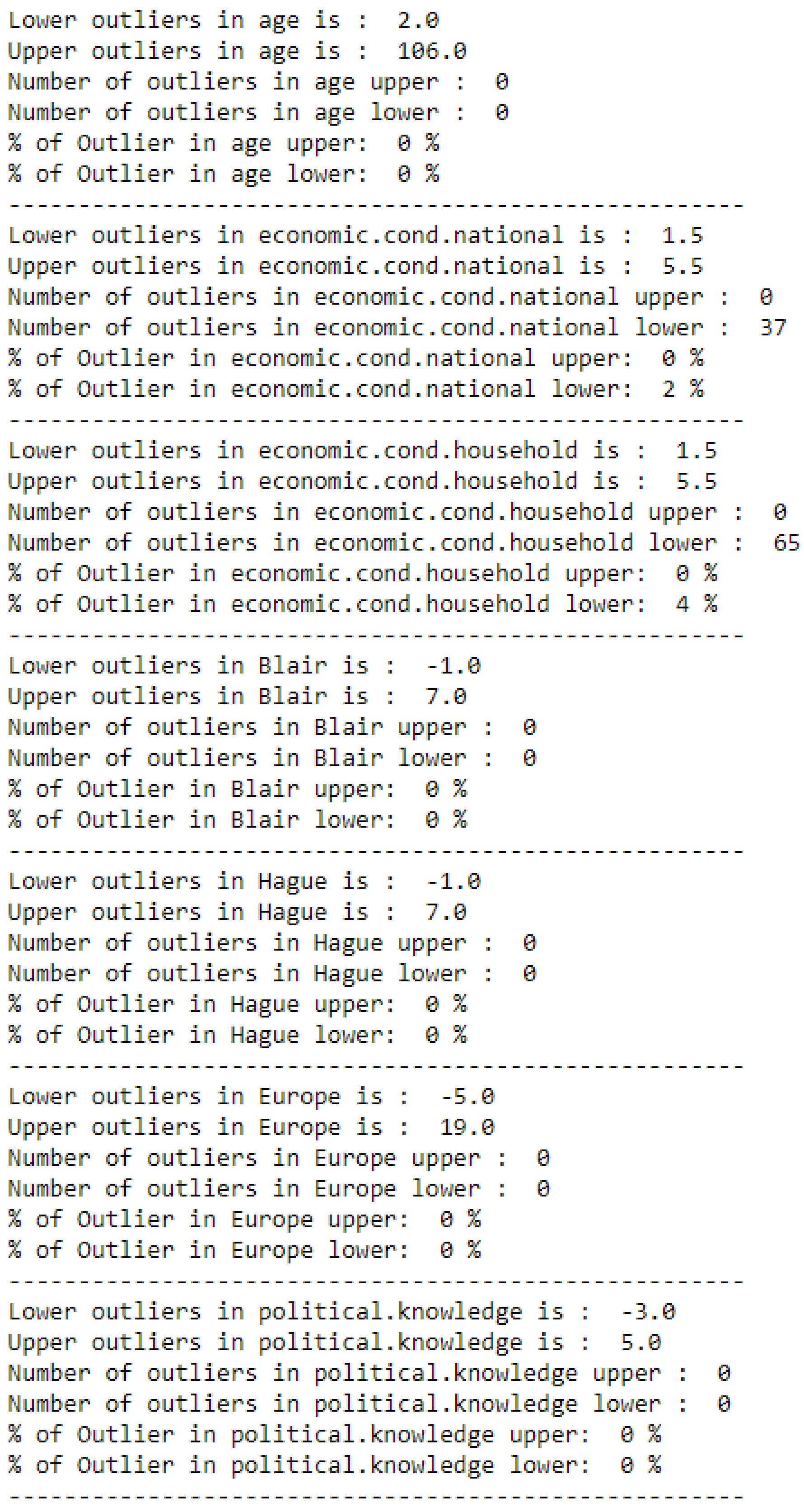


Figure 1: Boxplot with Outliers

There are outliers in economic.cond.national, economic.cond.household.

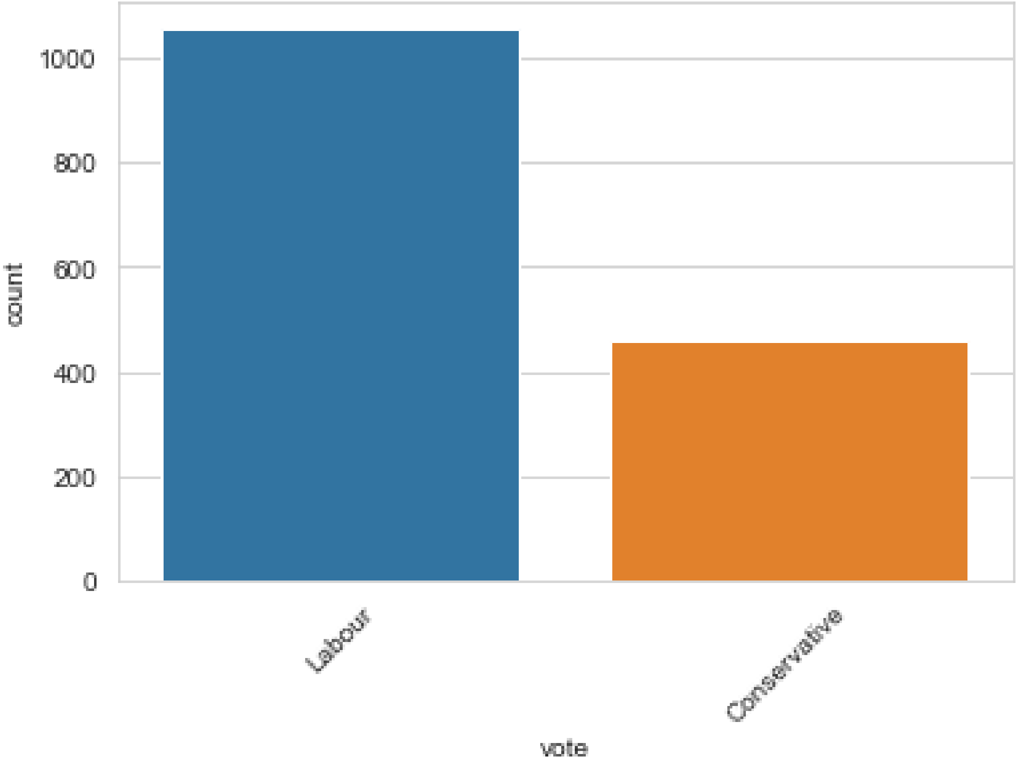


```
Lower outliers in age is :  2.0
Upper outliers in age is :  106.0
Number of outliers in age upper :  0
Number of outliers in age lower :  0
% of Outlier in age upper:  0 %
% of Outlier in age lower:  0 %
-----------------------------------------------------
Lower outliers in economic.cond.national is :  1.5
Upper outliers in economic.cond.national is :  5.5
Number of outliers in economic.cond.national upper :  0
Number of outliers in economic.cond.national lower :  37
% of Outlier in economic.cond.national upper:  0 %
% of Outlier in economic.cond.national lower:  2 %
-----------------------------------------------------
Lower outliers in economic.cond.household is :  1.5
Upper outliers in economic.cond.household is :  5.5
Number of outliers in economic.cond.household upper :  0
Number of outliers in economic.cond.household lower :  65
% of Outlier in economic.cond.household upper:  0 %
% of Outlier in economic.cond.household lower:  4 %
-----------------------------------------------------
Lower outliers in Blair is :  -1.0
Upper outliers in Blair is :  7.0
Number of outliers in Blair upper :  0
Number of outliers in Blair lower :  0
% of Outlier in Blair upper:  0 %
% of Outlier in Blair lower:  0 %
-----------------------------------------------------
Lower outliers in Hague is :  -1.0
Upper outliers in Hague is :  7.0
Number of outliers in Hague upper :  0
Number of outliers in Hague lower :  0
% of Outlier in Hague upper:  0 %
% of Outlier in Hague lower:  0 %
-----------------------------------------------------
Lower outliers in Europe is :  -5.0
Upper outliers in Europe is :  19.0
Number of outliers in Europe upper :  0
Number of outliers in Europe lower :  0
% of Outlier in Europe upper:  0 %
% of Outlier in Europe lower:  0 %
-----------------------------------------------------
Lower outliers in political.knowledge is :  -3.0
Upper outliers in political.knowledge is :  5.0
Number of outliers in political.knowledge upper :  0
Number of outliers in political.knowledge lower :  0
% of Outlier in political.knowledge upper:  0 %
% of Outlier in political.knowledge lower:  0 %
-----------------------------------------------------
```

The outliers need not be treated as there are only few and denotes the assessment scale.

Analysis of vote

```
Description of vote
---------------------------------------------------------------------------
-
count        1517
unique          2
top        Labour
freq         1057
Name: vote, dtype: object

Countplot of vote
---------------------------------------------------------------------------
-
```



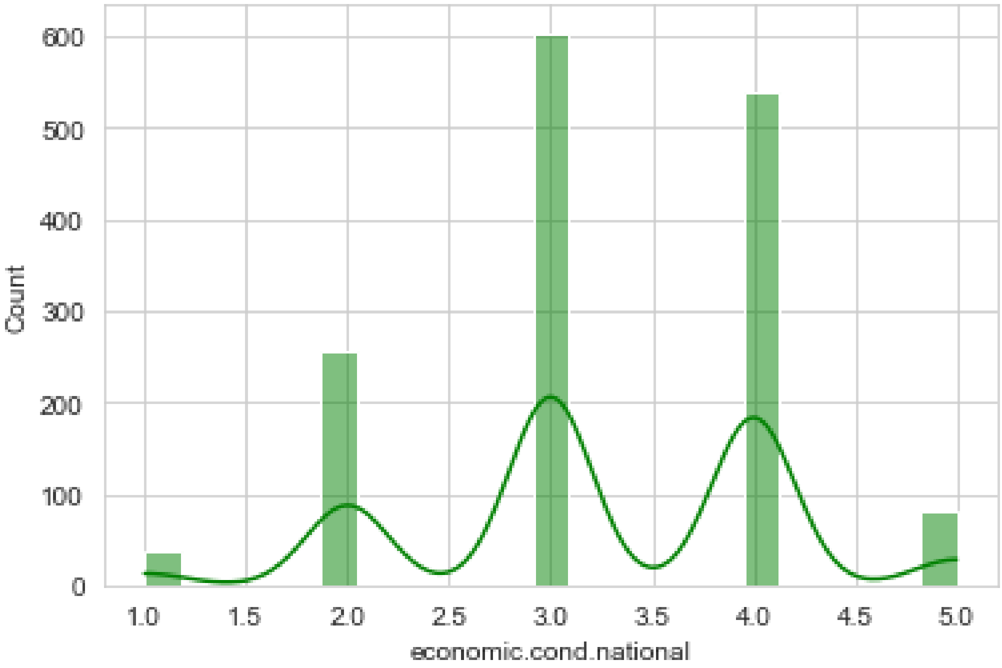
Labour values are more in number than Conservative values.

Analysis of economic.cond.national

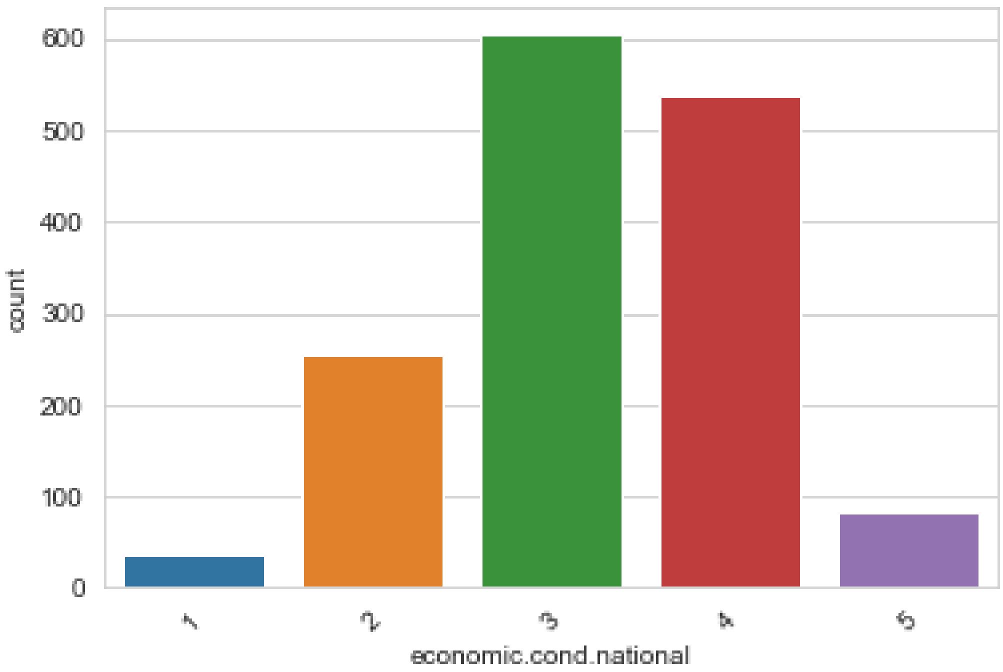
```
Value Count of economic.cond.national
---------------------------------------------------------------------------
-
3    604
4    538
2    256
5     82
1     37
Name: economic.cond.national, dtype: int64
```

Most of the voters have rated current national economic conditions as 3 (above average conditions).

```
Description of economic.cond.national
---------------------------------------------------------------------------
-
count    1517.000000
mean        3.245221
std         0.881792
min         1.000000
25%         3.000000
50%         3.000000
75%         4.000000
max         5.000000
Name: economic.cond.national, dtype: float64

Interquartile range (IQR) of is  1.0
Range of values:  4

Distribution of economic.cond.national
---------------------------------------------------------------------------
-
```



```
Countplot of economic.cond.national
---------------------------------------------------------------------------
-
```



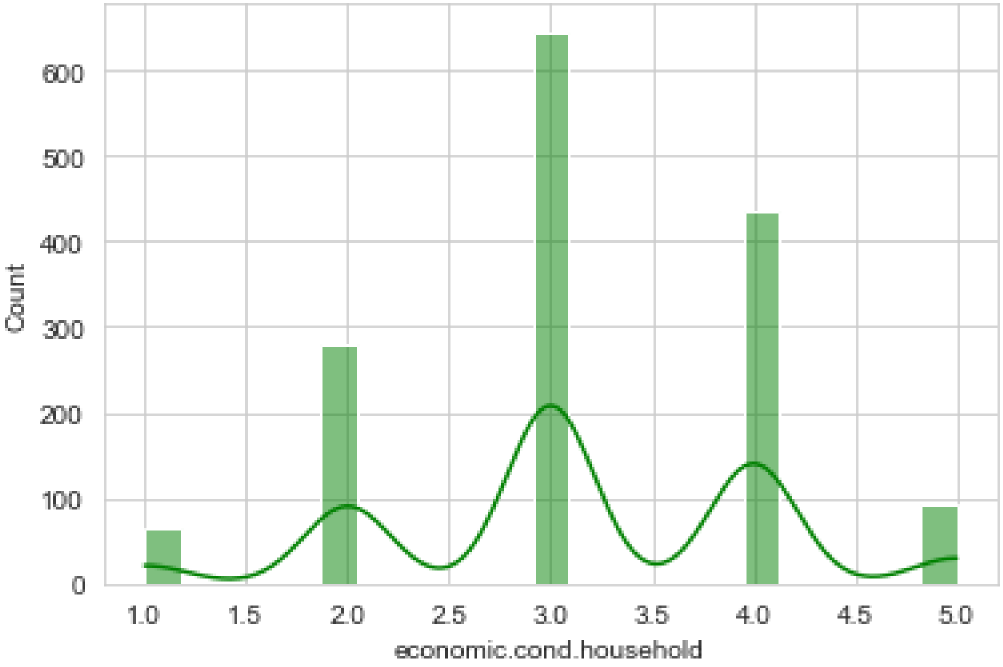
Analysis of economic.cond.household

```
Value Count of economic.cond.household
---------------------------------------------------------------------------
-
3    645
4    435
2    280
5     92
1     65
Name: economic.cond.household, dtype: int64
```

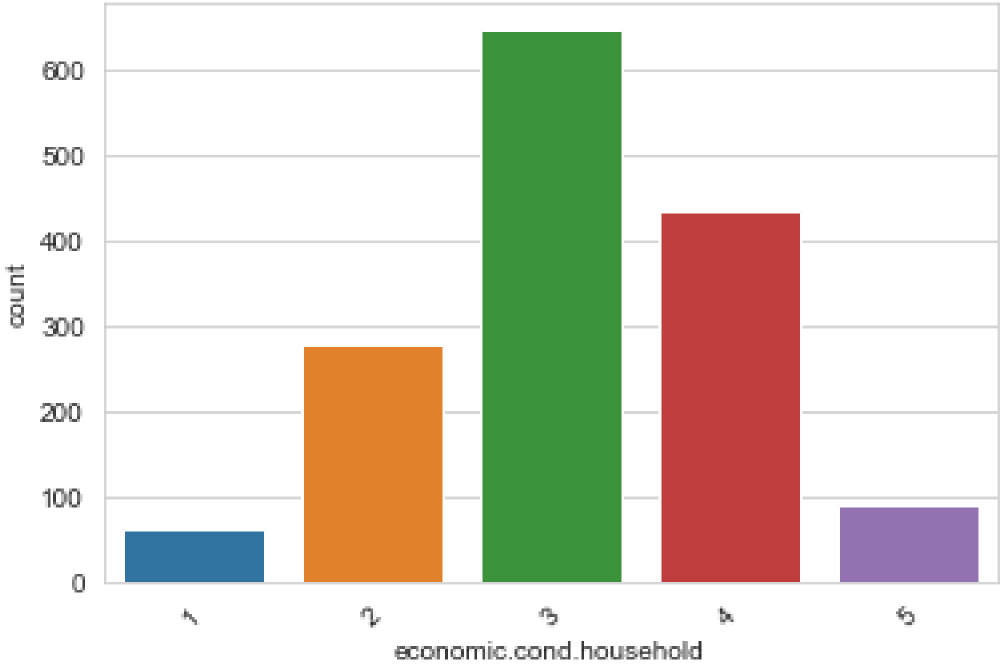
Most of the voters have rated current household economic conditions as 3 (above average conditions).

```
Description of economic.cond.household
---------------------------------------------------------------------------
-
count    1517.000000
mean        3.137772
std         0.931069
min         1.000000
25%         3.000000
50%         3.000000
75%         4.000000
max         5.000000
Name: economic.cond.household, dtype: float64

Interquartile range (IQR) of is  1.0
```

```
Range of values:  4

Distribution of economic.cond.household
---------------------------------------------------------------------------
-
```



```
Countplot of economic.cond.household
---------------------------------------------------------------------------
-
```

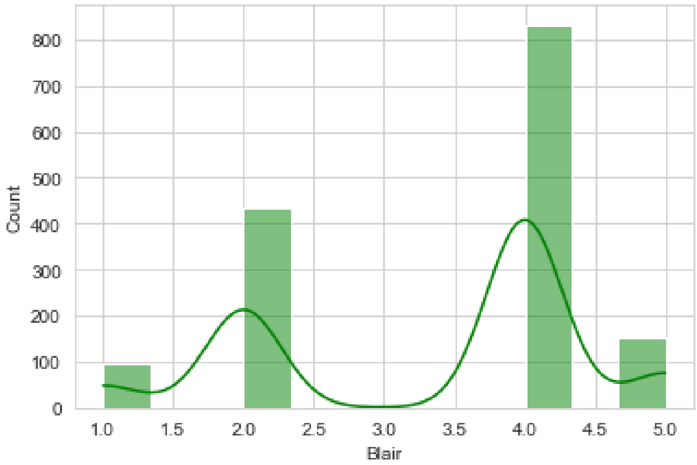


Analysis of Blair

```
Value Count of Blair
----------------------------------------------------------------------------
-
4    833
2    434
5    152
1     97
3      1
Name: Blair, dtype: int64
```

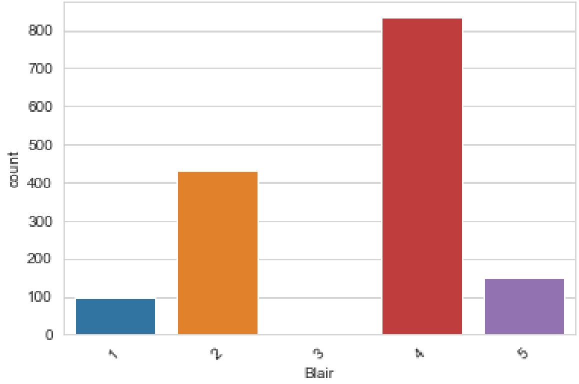
Most of them have given a high rating of 4 to the Labour leader.

```
Description of Blair
----------------------------------------------------------------------------
-
count    1517.000000
mean        3.335531
std         1.174772
min         1.000000
25%         2.000000
50%         4.000000
75%         4.000000
max         5.000000
Name: Blair, dtype: float64

Interquartile range (IQR) of is  2.0
Range of values:  4

Distribution of Blair
----------------------------------------------------------------------------
-
```



```
Countplot of Blair
```

```
---------------------------------------------------------------------------
-
```



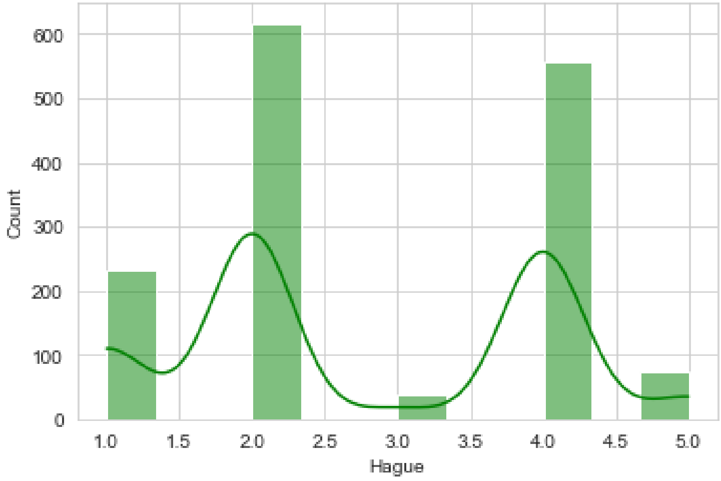
Analysis of Hague

```
Value Count of Hague
---------------------------------------------------------------------------
-
2    617
4    557
1    233
5     73
3     37
Name: Hague, dtype: int64
```

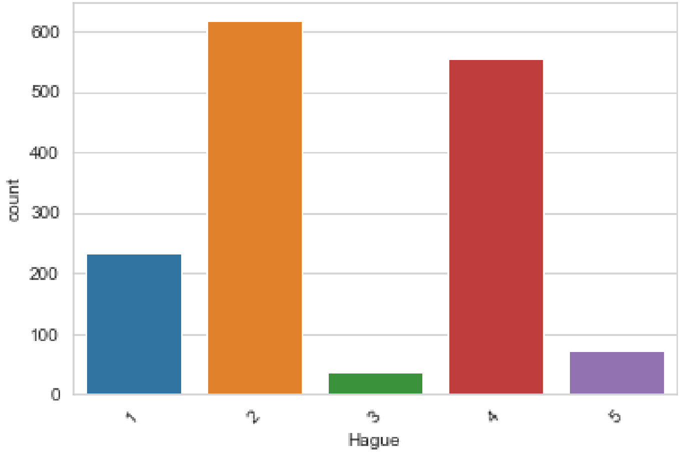
Most of them have given a low rating of 2 to the Conservative leader.

```
Description of Hague
---------------------------------------------------------------------------
-
count    1517.000000
mean        2.749506
std         1.232479
min         1.000000
25%         2.000000
50%         2.000000
75%         4.000000
max         5.000000
Name: Hague, dtype: float64

Interquartile range (IQR) of is  2.0
Range of values:  4

Distribution of Hague
```

```
---------------------------------------------------------------------------
-
```



```
Countplot of Hague
---------------------------------------------------------------------------
-
```



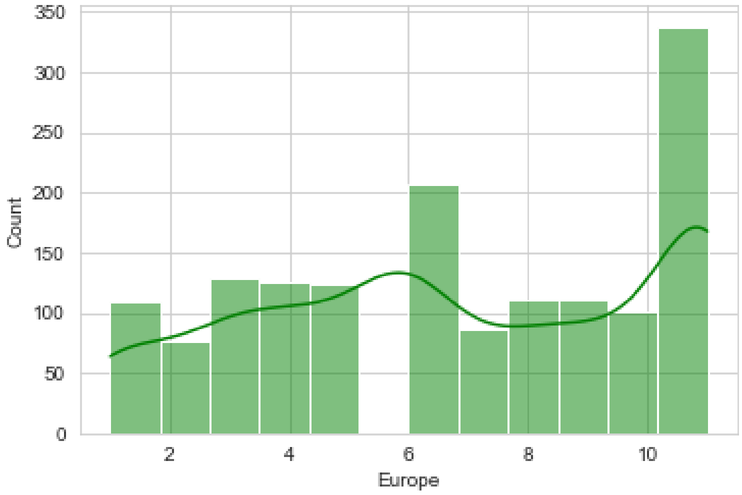
Analysis of Europe

```
Value Count of Europe
---------------------------------------------------------------------------
-
11     338
6      207
3      128
4      126
5      123
9      111
8      111
1      109
10     101
```

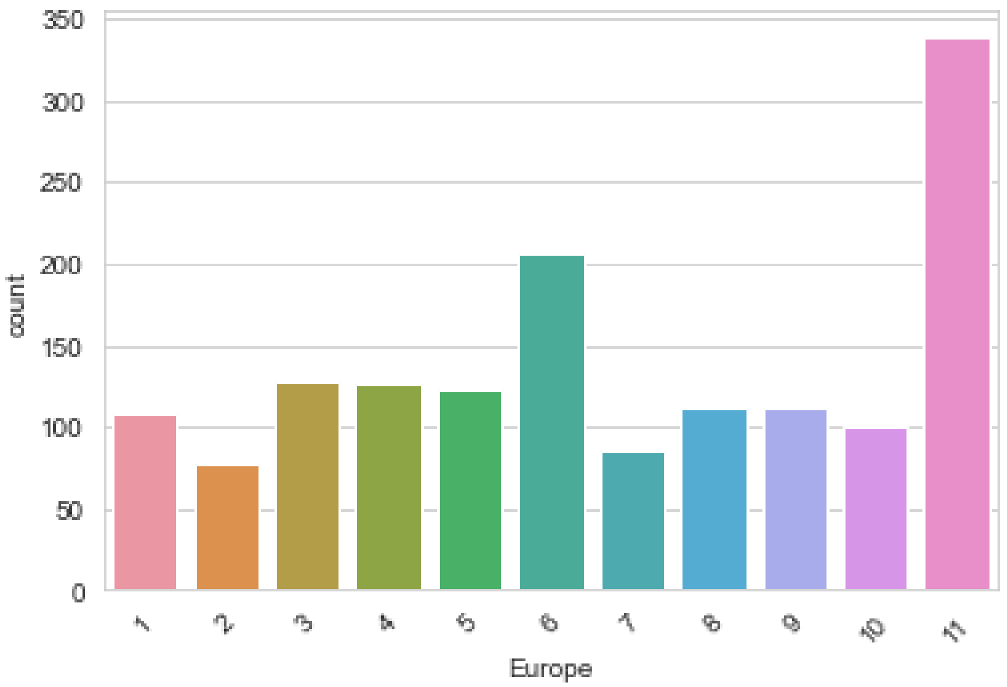
```
7       86
2       77
Name: Europe, dtype: int64
```

Most of the voters have given an high score of 11 which indicates they are ‘Eur osceptic’.

```
Description of Europe
----------------------------------------------------------------------------
-
count    1517.000000
mean        6.740277
std         3.299043
min         1.000000
25%         4.000000
50%         6.000000
75%        10.000000
max        11.000000
Name: Europe, dtype: float64

Interquartile range (IQR) of is  6.0
Range of values:  10

Distribution of Europe
```



```
Countplot of Europe
----------------------------------------------------------------------------
-
```



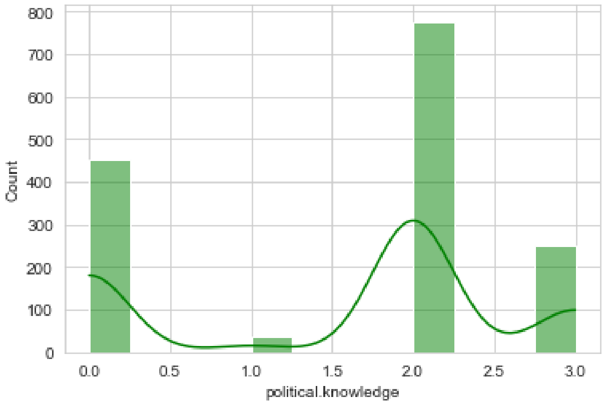
Analysis of political.knowledge

```
Value Count of political.knowledge
---------------------------------------------------------------------------
-
2    776
0    454
3    249
1     38
Name: political.knowledge, dtype: int64
```

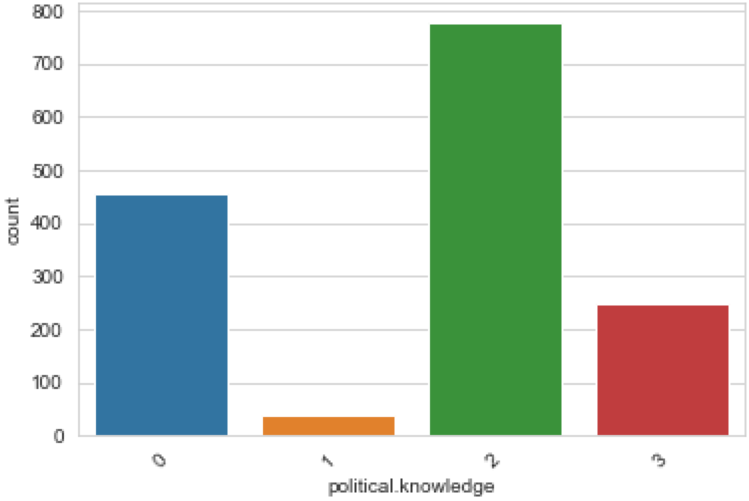
Most of the voters have given the rating 2 indicating a good knowledge on parties' positions on European integration.

```
Description of political.knowledge
---------------------------------------------------------------------------
-
count    1517.000000
mean        1.540541
std         1.084417
min         0.000000
25%         0.000000
50%         2.000000
75%         2.000000
max         3.000000
Name: political.knowledge, dtype: float64

Interquartile range (IQR) of is  2.0
```

```
Range of values:  3

Distribution of political.knowledge
---------------------------------------------------------------------------
-
```



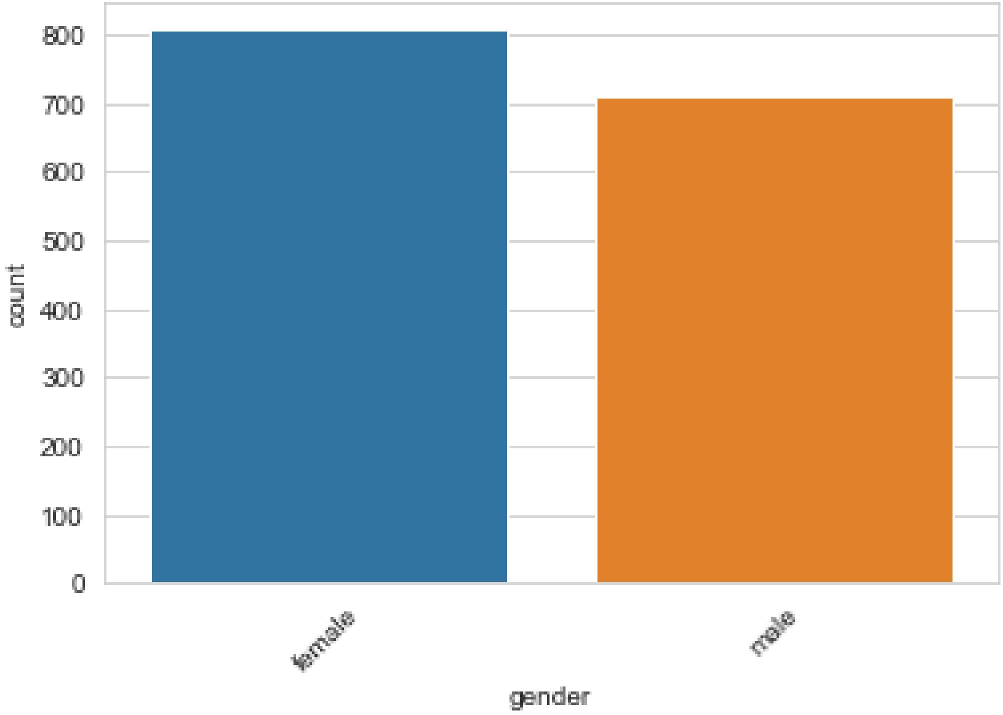
```
Countplot of political.knowledge
---------------------------------------------------------------------------
-
```



Analysis of gender

```
Value Count of gender
---------------------------------------------------------------------------
-
female    808
male      709
Name: gender, dtype: int64

Description of gender
---------------------------------------------------------------------------
-
count       1517
```

```
unique          2
top        female
freq          808
Name: gender, dtype: object

Countplot of gender
---------------------------------------------------------------------------
-
```

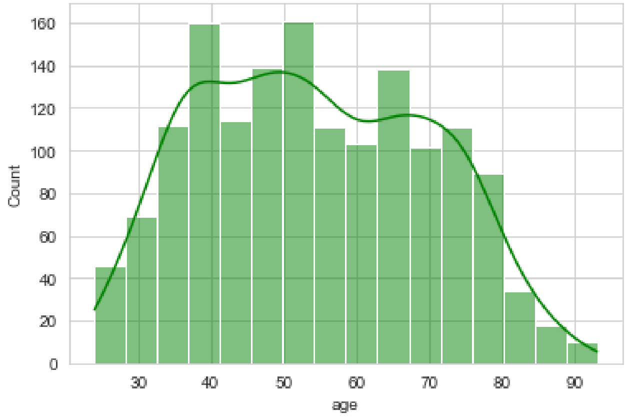


Most of the voters responded to the survey are females.

Analysis of age

```
Description of age
---------------------------------------------------------------------------
-
count    1517.000000
mean       54.241266
std        15.701741
min        24.000000
25%        41.000000
50%        53.000000
75%        67.000000
max        93.000000
Name: age, dtype: float64

Interquartile range (IQR) of is  26.0
Range of values:  69
```

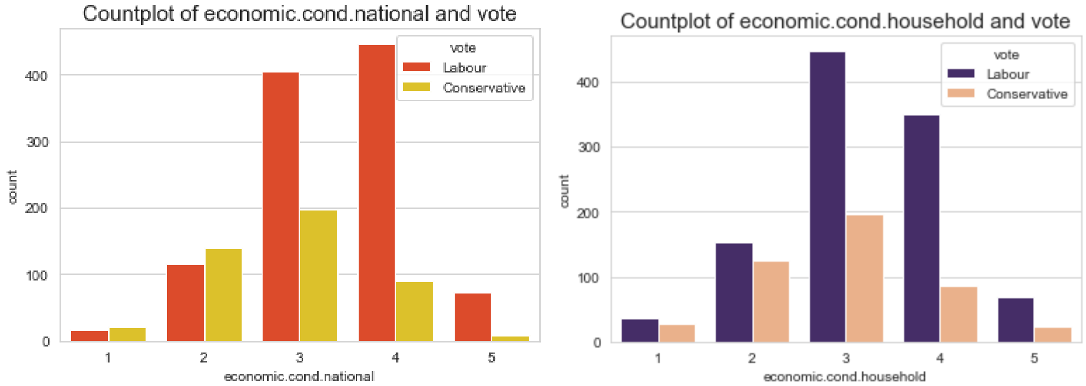
```
Distribution of age
----------------------------------------------------------------------------
-
```



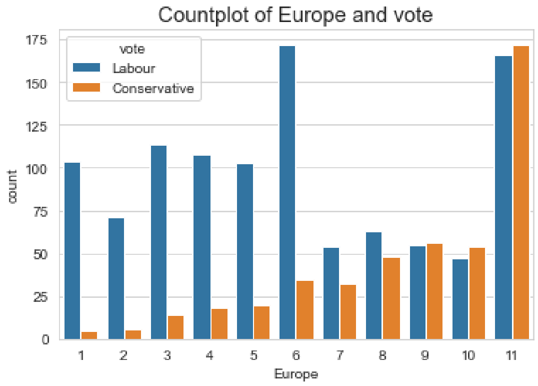
Age is almost symmetrically distributed.

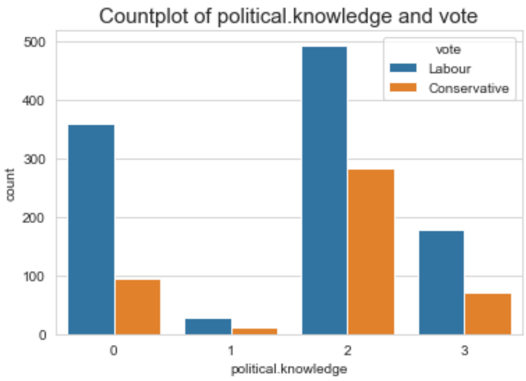
Bivariate Analysis

With vote

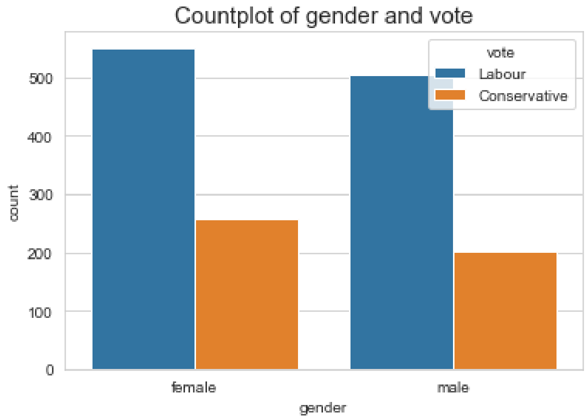


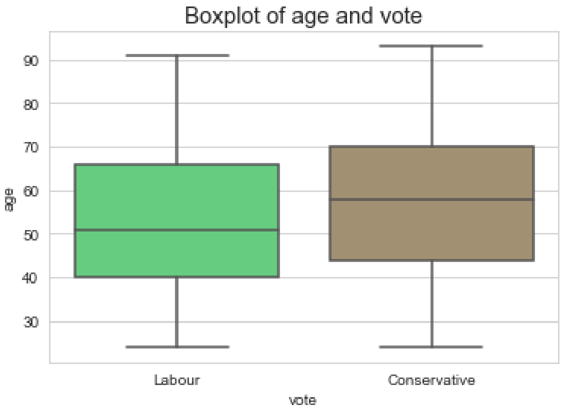
Most of the voters assessed the ratings of economic conditions and household conditions as good favouring to ‘Labour’ Party.


Countplot of Europe and vote


Countplot of political.knowledge and vote

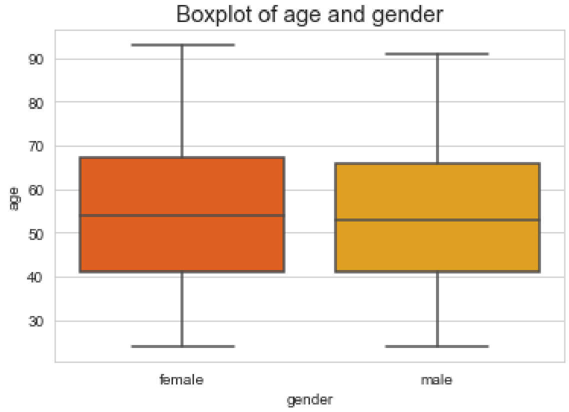
Most people having Eurosceptic sentiment favour Conservative property.

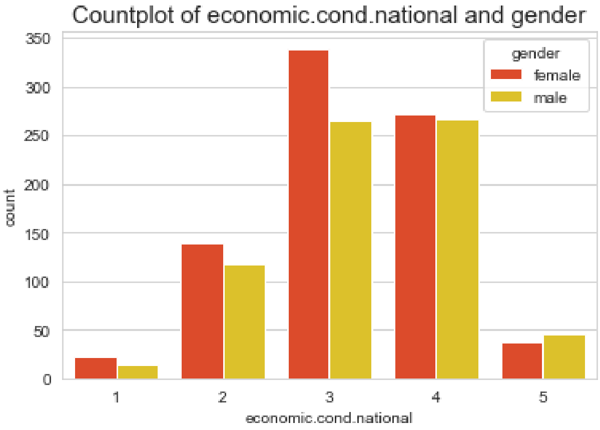

Countplot of gender and vote


Boxplot of age and vote

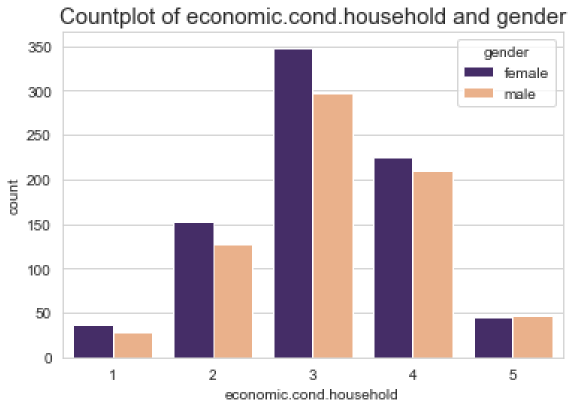
People older than 55 mostly favour Conservative party.

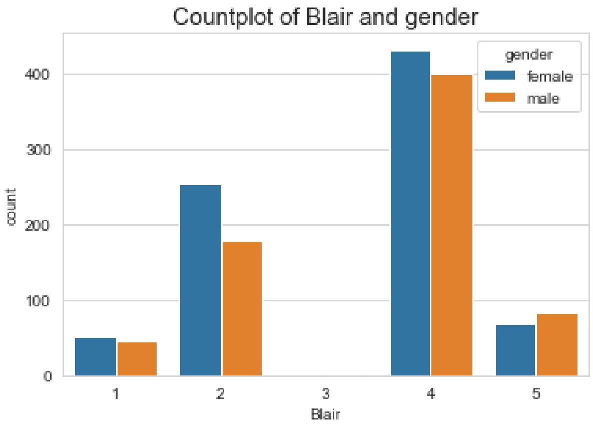
With gender


Boxplot of age and gender

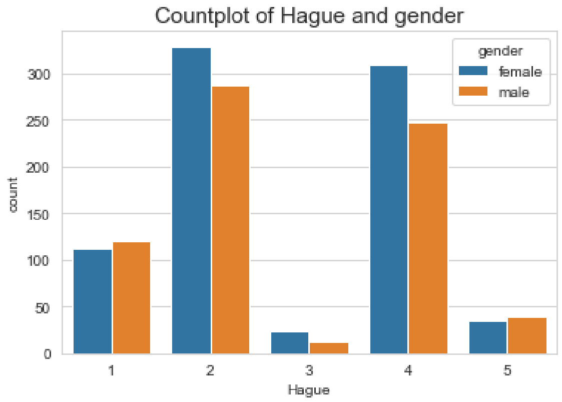

Countplot of economic.cond.national and gender

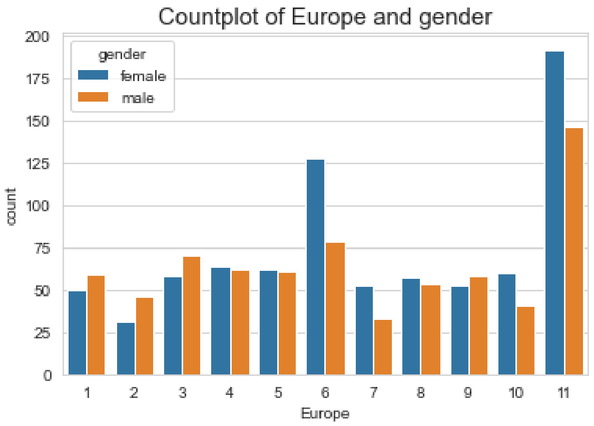
The number of females who has given a high score to national economic conditions is less than males.

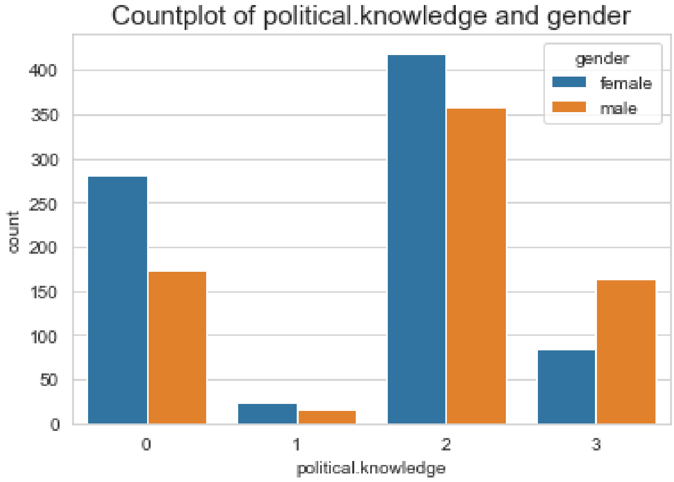




The number of females who has given a low score to household economic conditions is higher than males.







More females are Eurosceptic than males.

PAIR PLOT

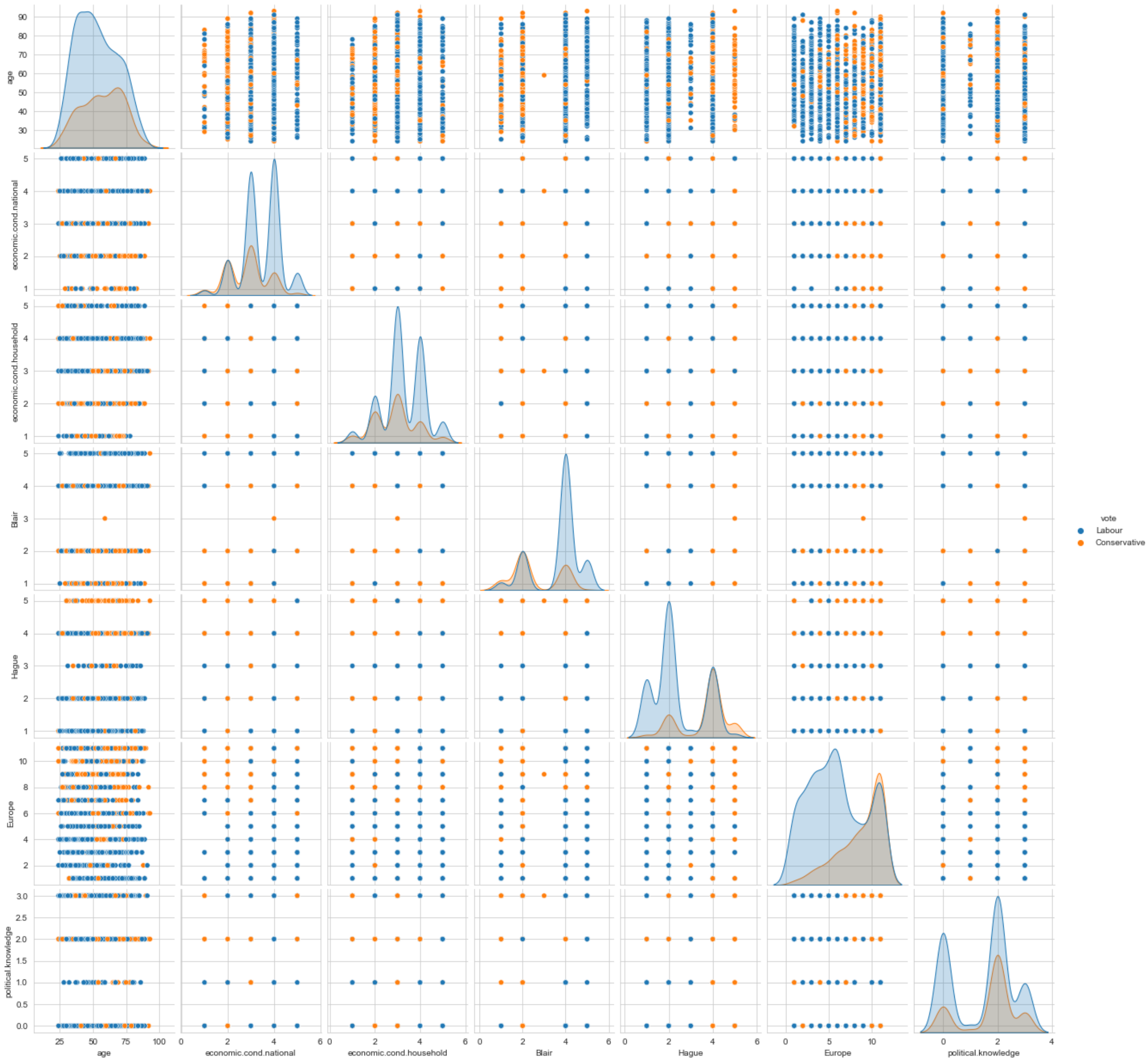


Figure 2: Pairplot

CORRELATION HEATMAP

From pairplot and heatmap we can infer that there is a slight positive correlation between economic.cond.household and other variables such as economic.cond.national, Blair. Hague is slightly positively correlated with Europe. Blair and Hague, Blair and Europe are slightly negatively correlated.

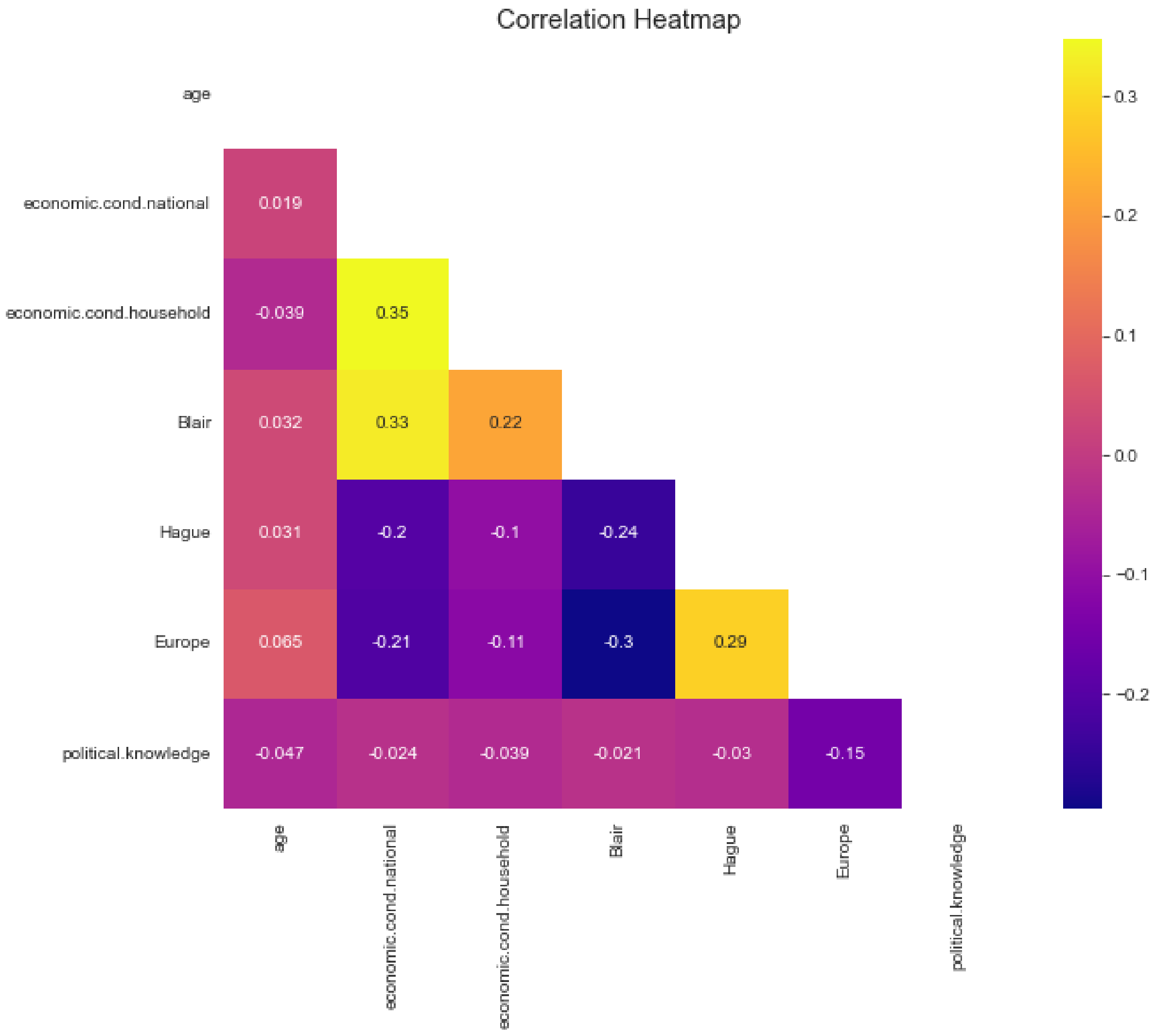


Figure 3: Correlation Heatmap

Multivariate Analysis

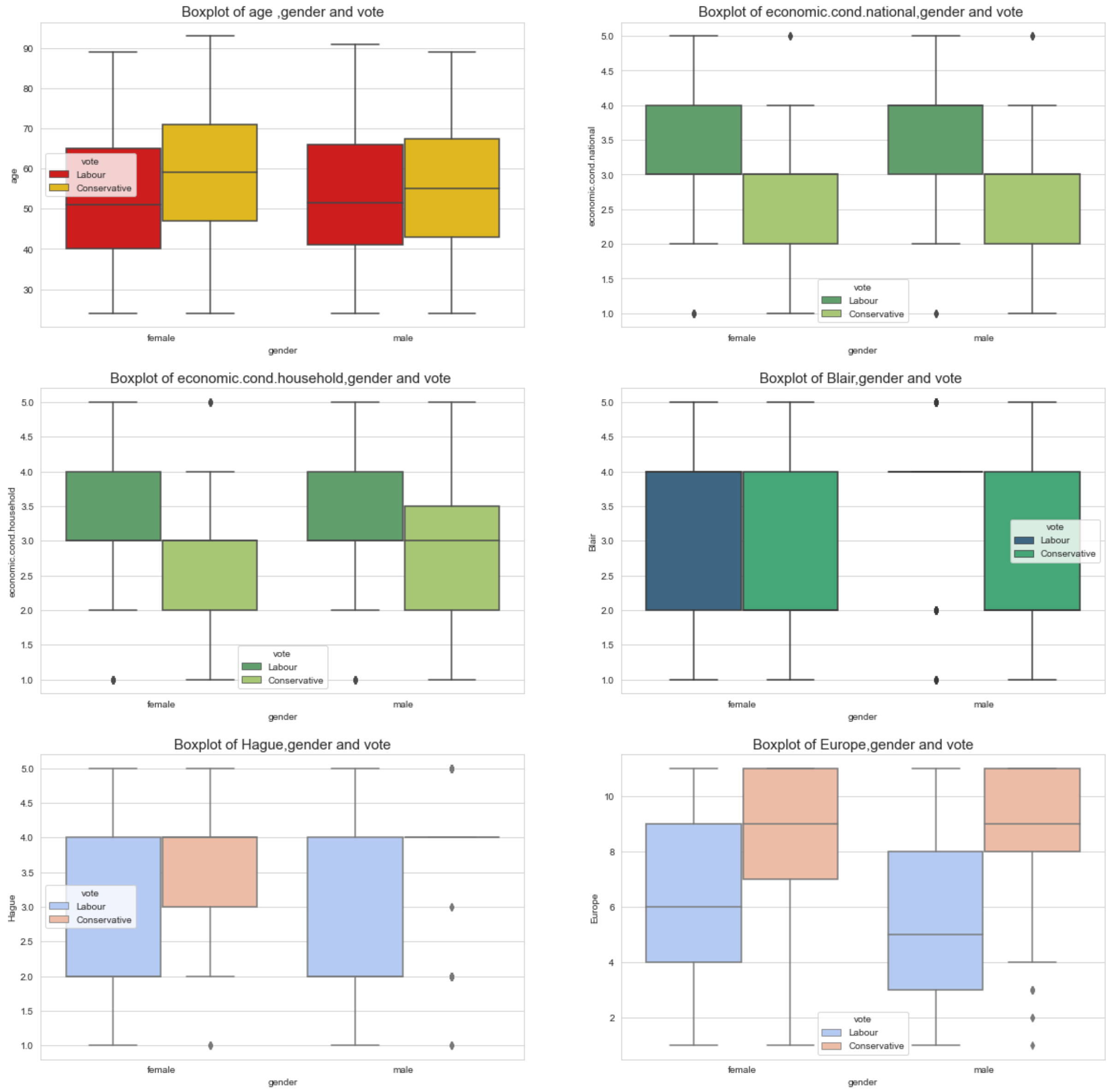
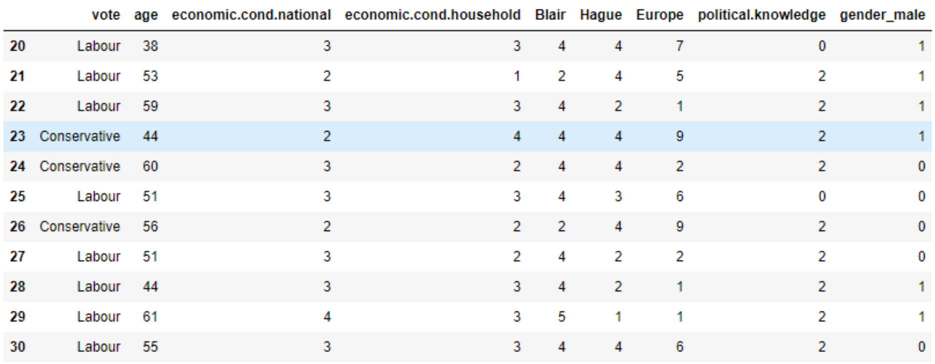


Figure 4: Boxplots for Multivariate Analysis1

Most of the females favouring Conservative party have an average age higher than males favouring Conservative party.

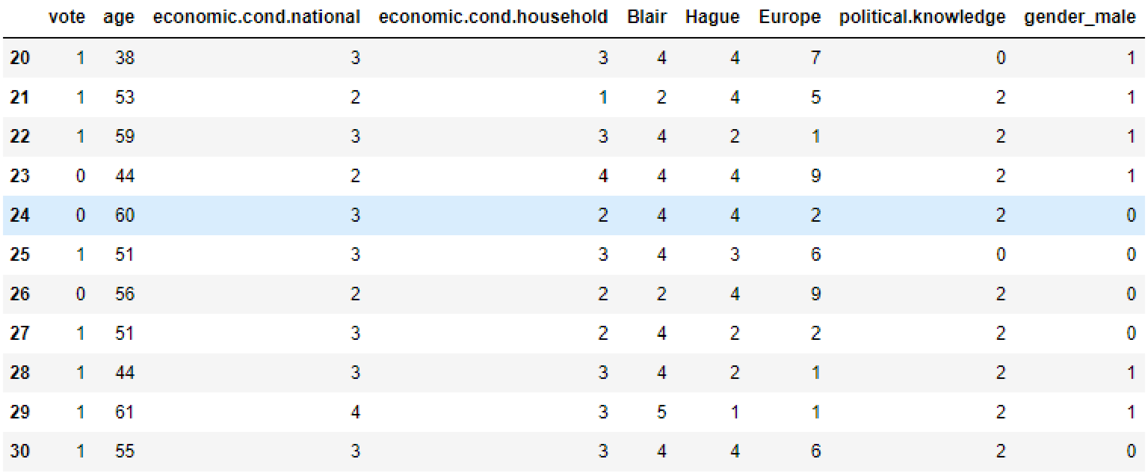
1.3. Encode the data (having string values) for Modelling. Is Scaling necessary here or not? Data Split: Split the data into train and test (70:30).

get_dummies used to encode ‘gender’ column with drop_first set to true.



	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
20	Labour	38	3	3	4	4	7	0	1
21	Labour	53	2	1	2	4	5	2	1
22	Labour	59	3	3	4	2	1	2	1
23	Conservative	44	2	4	4	4	9	2	1
24	Conservative	60	3	2	4	4	2	2	0
25	Labour	51	3	3	4	3	6	0	0
26	Conservative	56	2	2	2	4	9	2	0
27	Labour	51	3	2	4	2	2	2	0
28	Labour	44	3	3	4	2	1	2	1
29	Labour	61	4	3	5	1	1	2	1
30	Labour	55	3	3	4	4	6	2	0

LabelEncoder used to encode column 'vote'. Labour is encoded as 1 and Conservative as 0.



	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
20	1	38	3	3	4	4	7	0	1
21	1	53	2	1	2	4	5	2	1
22	1	59	3	3	4	2	1	2	1
23	0	44	2	4	4	4	9	2	1
24	0	60	3	2	4	4	2	2	0
25	1	51	3	3	4	3	6	0	0
26	0	56	2	2	2	4	9	2	0
27	1	51	3	2	4	2	2	2	0
28	1	44	3	3	4	2	1	2	1
29	1	61	4	3	5	1	1	2	1
30	1	55	3	3	4	4	6	2	0

Scaling not necessary for lda and logistic regression but required for KNN,RF which use distance based calculation. Decision trees and ensemble methods do not require feature scaling to be performed as they are not sensitive to the variance in the data.

The data is split into train and test data in the ratio 70:30.

The class count of training labels is:

```
1    754
0    307
Name: vote, dtype: int64
```

The class is heavily imbalanced hence SMOTE is applied for Oversampling. It is applied on train data only.

The class count of training labels after applying SMOTE is:

```
0    754
```

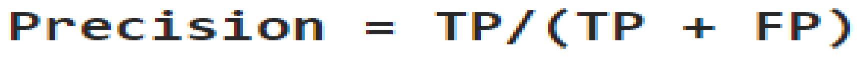
```
   1      754
Name: vote, dtype: int64
```

Basic models and GridSearchCV models are built on unbalanced train data.

1.4. Apply Logistic Regression and LDA

Classification Report

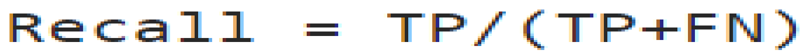
Precision is the ability of a classifier not to label an instance positive that is actually negative.



Precision = TP/(TP + FP)

Equation 1:Precision

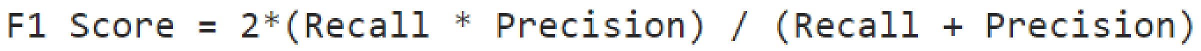
Recall is the fraction of positives that were correctly identified.



Recall = TP/(TP+FN)

Equation 2:Recall

F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0.



F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Equation 3:F1 Score

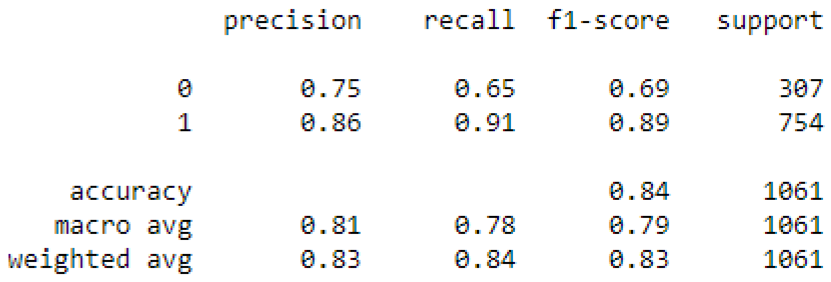
Recall is the important metric as it is inversely proportional to Type 2(False Negative) error. In this problem decreasing the Type 2 error is important.

Model 1 - Basic Logistic Regression

A default Logistic Regression model is built with its values set to default.

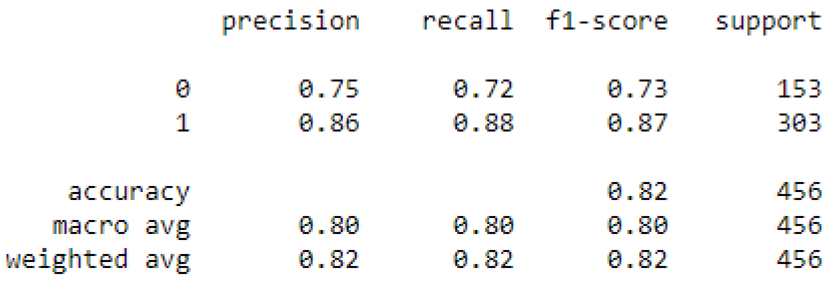
```
LogisticRegression()
```

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.75      0.65      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.84      1061
   macro avg       0.81      0.78      0.79      1061
weighted avg       0.83      0.84      0.83      1061
```

Classification Report for Test Data

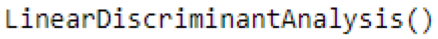


```
              precision    recall  f1-score   support

           0       0.75      0.72      0.73       153
           1       0.86      0.88      0.87       303

    accuracy                           0.82       456
   macro avg       0.80      0.80      0.80       456
weighted avg       0.82      0.82      0.82       456
```

There is no underfitting or overfitting since the accuracy of train data is 0.84 and test data is 0.82 which shows very small difference in values between two.

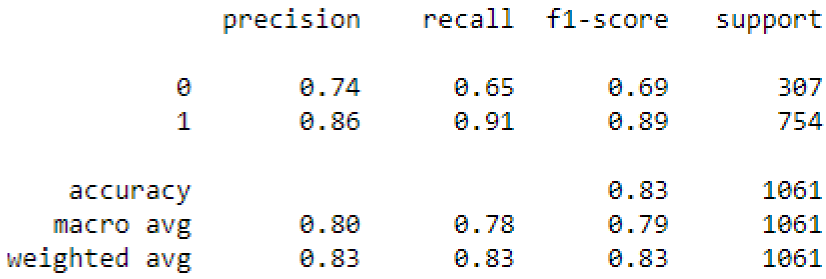
Model 2 - Basic Linear Discriminant Analysis (LDA)

A default LDA model is built with its values set to default.



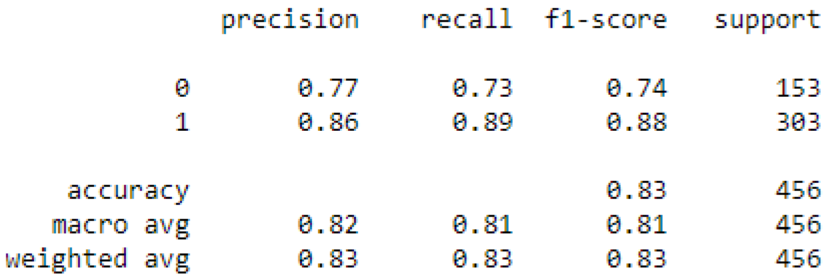
```
LinearDiscriminantAnalysis()
```

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.74      0.65      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.83      1061
   macro avg       0.80      0.78      0.79      1061
weighted avg       0.83      0.83      0.83      1061
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.77      0.73      0.74       153
           1       0.86      0.89      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.81      0.81       456
weighted avg       0.83      0.83      0.83       456
```

There is no underfitting or overfitting since the accuracy of train data is 0.83 and test data is 0.83(equal values).

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

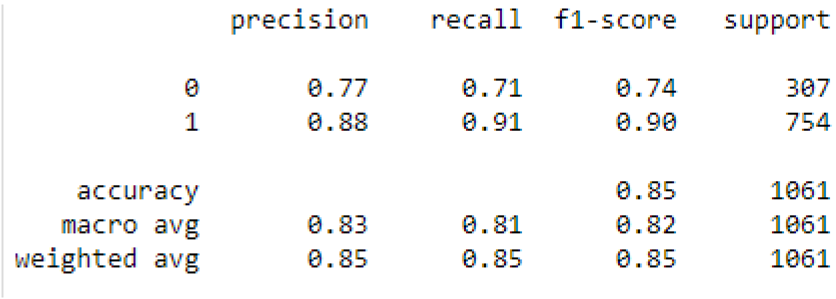
Model 3 - Basic KNN Model

KNN is a distance based algorithm hence scaling is done the independent train and test data.

A default KNN model is built with its values set to default.

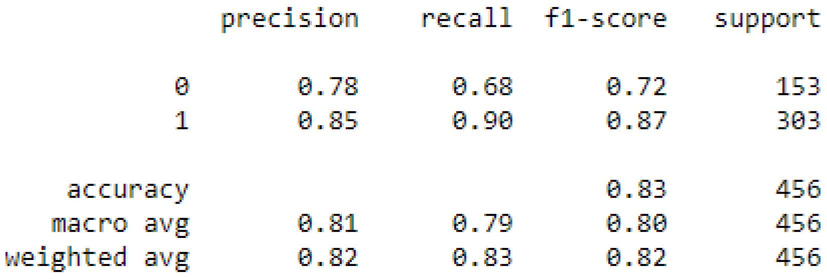
```
KNeighborsClassifier()
```

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.77      0.71      0.74       307
           1       0.88      0.91      0.90       754

    accuracy                           0.85      1061
   macro avg       0.83      0.81      0.82      1061
weighted avg       0.85      0.85      0.85      1061
```

Classification Report for Test Data



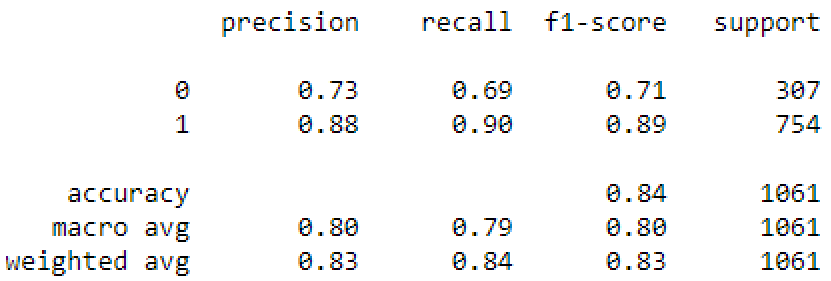
```
              precision    recall  f1-score   support

           0       0.78      0.68      0.72       153
           1       0.85      0.90      0.87       303

    accuracy                           0.83       456
   macro avg       0.81      0.79      0.80       456
weighted avg       0.82      0.83      0.82       456
```

There is no underfitting or overfitting since the accuracy of train data is 0.85 and test data is 0.83 which shows very small difference in values between two.

Model 4- Basic Naïve Bayes Model

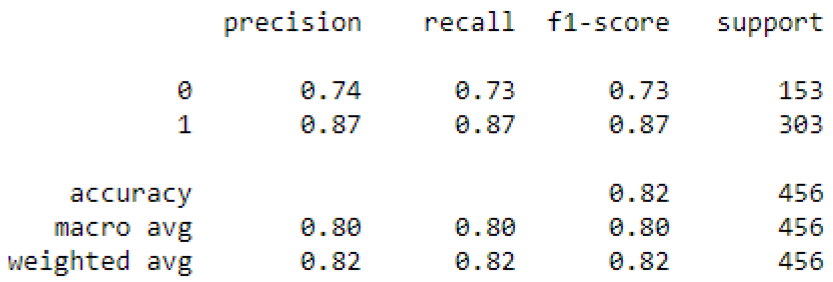
A default Naïve Bayes model is built with its values set to default.

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.73      0.69      0.71       307
           1       0.88      0.90      0.89       754

    accuracy                           0.84      1061
   macro avg       0.80      0.79      0.80      1061
weighted avg       0.83      0.84      0.83      1061
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.74      0.73      0.73       153
           1       0.87      0.87      0.87       303

    accuracy                           0.82       456
   macro avg       0.80      0.80      0.80       456
weighted avg       0.82      0.82      0.82       456
```

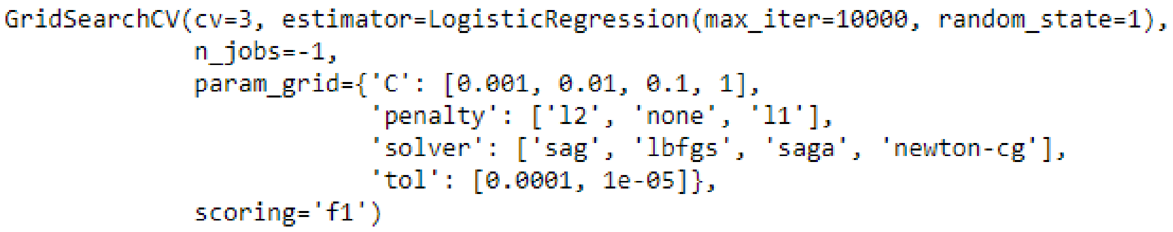
There is no underfitting or overfitting since the accuracy of train data is 0.84 and test data is 0.82 which shows very small difference in values between two.

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

Model 5 - GridSearchCV Logistic Regression

Since unbalanced data is used and both precision , recall is important ; f1 score is chosen as the scoring parameter value.

The parameters chosen for GridSearchCV Logistic Regression are :



```
GridSearchCV(cv=3, estimator=LogisticRegression(max_iter=10000, random_state=1),
             n_jobs=-1,
             param_grid={'C': [0.001, 0.01, 0.1, 1],
                         'penalty': ['l2', 'none', 'l1'],
                         'solver': ['sag', 'lbfgs', 'saga', 'newton-cg'],
                         'tol': [0.0001, 1e-05]},
             scoring='f1')
```

Solver - Algorithm to use in the optimization problem

C - Inverse of regularization strength

Penalty – norm of the penalty

tol - Tolerance for stopping criteria.

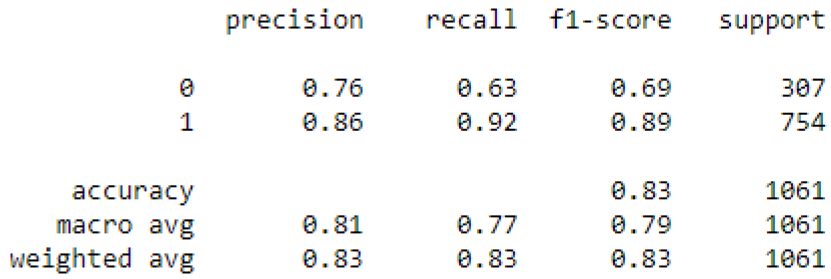
n_jobs - Number of CPU cores used when parallelizing over classes if multi_class='ovr'

Cross validation is given as 3 which is one of the best values.

The best estimator is the one which has the best values for the parameters combination with best f1 score.

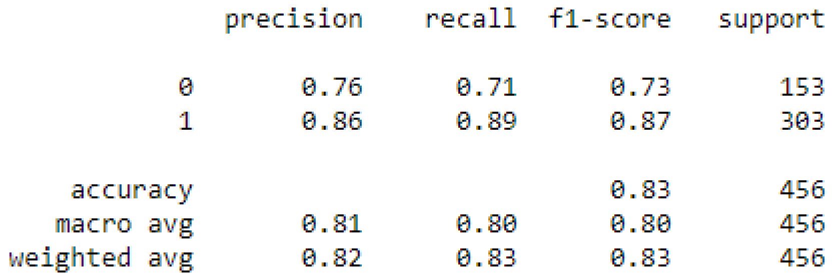
```
{'C': 0.1, 'penalty': 'l1', 'solver': 'saga', 'tol': 1e-05}

LogisticRegression(C=0.1, max_iter=10000, penalty='l1', random_state=1,
                   solver='saga', tol=1e-05)
                                                                  In [507]:
```

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.76      0.63      0.69       307
           1       0.86      0.92      0.89       754

    accuracy                           0.83      1061
   macro avg       0.81      0.77      0.79      1061
weighted avg       0.83      0.83      0.83      1061
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.76      0.71      0.73       153
           1       0.86      0.89      0.87       303

    accuracy                           0.83       456
   macro avg       0.81      0.80      0.80       456
weighted avg       0.82      0.83      0.83       456
```

There is no underfitting or overfitting since the accuracy of train data is 0.83 and test data is 0.83 which are equal.

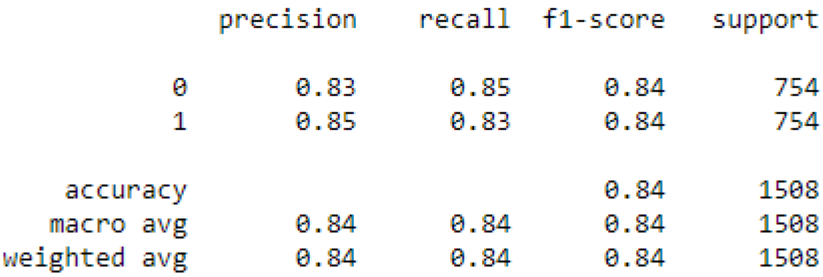
Model 6 - SMOTE Logistic Regression

Synthetic Minority Oversampling Technique (SMOTE) is a statistical technique for increasing the number of cases in the dataset in a balanced fashion.

Using this data the default Logistic Regression model is built.

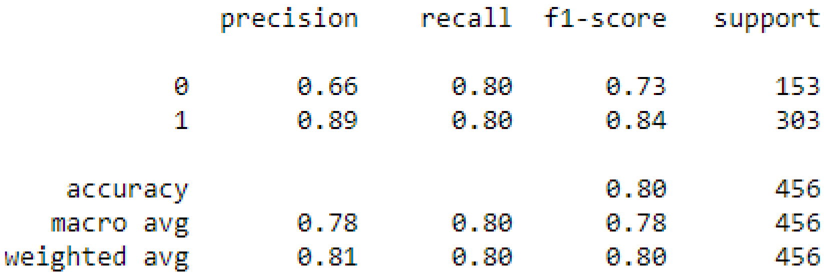
```
LogisticRegression()
```

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.83      0.85      0.84       754
           1       0.85      0.83      0.84       754

    accuracy                           0.84      1508
   macro avg       0.84      0.84      0.84      1508
weighted avg       0.84      0.84      0.84      1508
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.66      0.80      0.73       153
           1       0.89      0.80      0.84       303

    accuracy                           0.80       456
   macro avg       0.78      0.80      0.78       456
weighted avg       0.81      0.80      0.80       456
```

There is no underfitting or overfitting since the accuracy of train data is 0.84 and test data is 0.80 which has a small difference in values between the two.

Model 7 - GridSearchCV Linear Discriminant Analysis (LDA)

The parameters chosen for GridSearchCV LDA are:

```
GridSearchCV(cv=3, estimator=LinearDiscriminantAnalysis(), n_jobs=-1,
             param_grid={'solver': ['svd', 'lsqr', 'eigen']}, scoring='f1')
```

Solver to uses possible values:

‘svd’: Singular value decomposition . Does not compute the covariance matrix, therefore this solver is recommended for data with a large number of features.

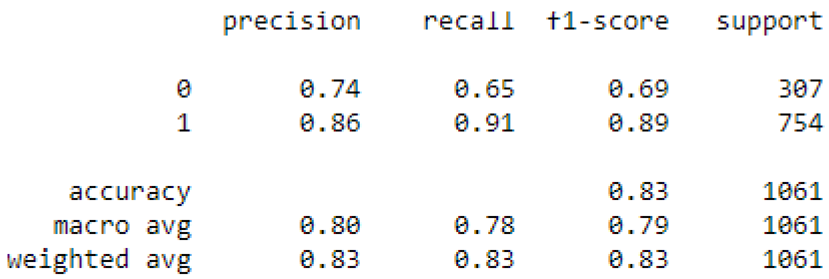
‘lsqr’: Least squares solution, can be combined with shrinkage.

‘eigen’: Eigenvalue decomposition, can be combined with shrinkage.

The best estimator is:

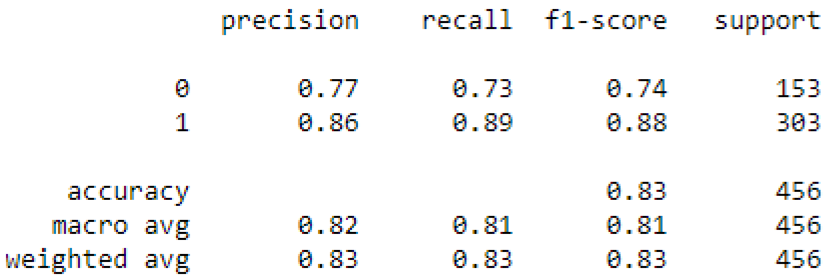
```
{'solver': 'svd'}

LinearDiscriminantAnalysis()
```

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.74      0.65      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.83      1061
   macro avg       0.80      0.78      0.79      1061
weighted avg       0.83      0.83      0.83      1061
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.77      0.73      0.74       153
           1       0.86      0.89      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.81      0.81       456
weighted avg       0.83      0.83      0.83       456
```

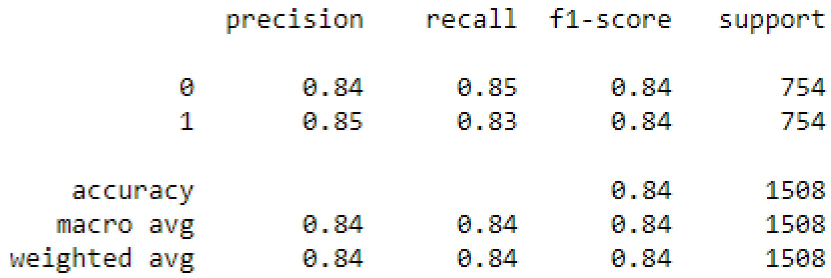
Train and Test accuracy have equal value of 0.83. So there is no underfitting or overfitting.

Model 8 - SMOTE LDA¶

Balanced train data is used to build LDA model

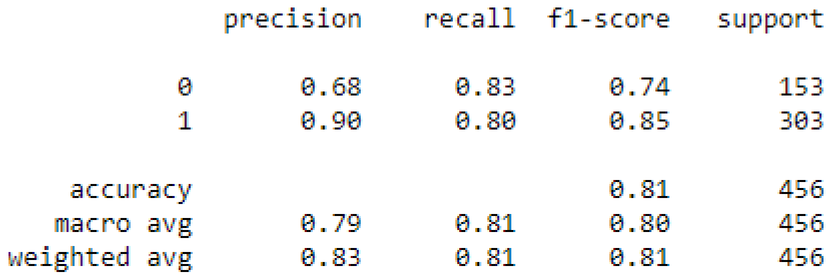
```
LinearDiscriminantAnalysis()
```

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.84      0.85      0.84       754
           1       0.85      0.83      0.84       754

    accuracy                           0.84      1508
   macro avg       0.84      0.84      0.84      1508
weighted avg       0.84      0.84      0.84      1508
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.68      0.83      0.74       153
           1       0.90      0.80      0.85       303

    accuracy                           0.81       456
   macro avg       0.79      0.81      0.80       456
weighted avg       0.83      0.81      0.81       456
```

There is no underfitting or overfitting since the accuracy of train data is 0.84 and test data is 0.81 which has a small difference in values between the two.

Model 9 - GridSearchCV KNN Model

The parameters chosen for GridSearchCV KNN are:

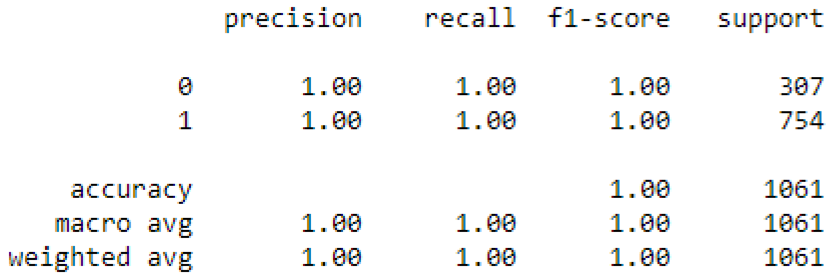
```
GridSearchCV(cv=3, estimator=KNeighborsClassifier(), n_jobs=-1,
             param_grid={'metric': ['chebyshev', 'euclidean', 'manhattan'],
                         'n_neighbors': [3, 5, 7, 9, 11, 13, 15, 17, 19, 21
],'weights': ['uniform', 'distance']},scoring='f1')
```

Weights – weight used in prediction
metric - distance metric
n_neighbors -Number of neighbors to use

The best estimator is:

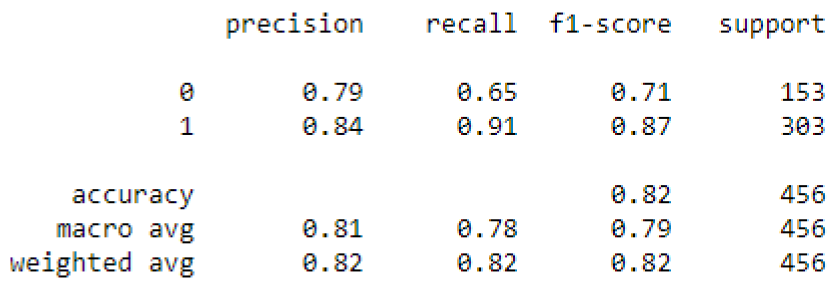
```
{'metric': 'manhattan', 'n_neighbors': 19, 'weights': 'distance'}

KNeighborsClassifier(metric='manhattan', n_neighbors=19, weights='distance'
)
```

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       307
           1       1.00      1.00      1.00       754

    accuracy                           1.00      1061
   macro avg       1.00      1.00      1.00      1061
weighted avg       1.00      1.00      1.00      1061
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.79      0.65      0.71       153
           1       0.84      0.91      0.87       303

    accuracy                           0.82       456
   macro avg       0.81      0.78      0.79       456
weighted avg       0.82      0.82      0.82       456
```

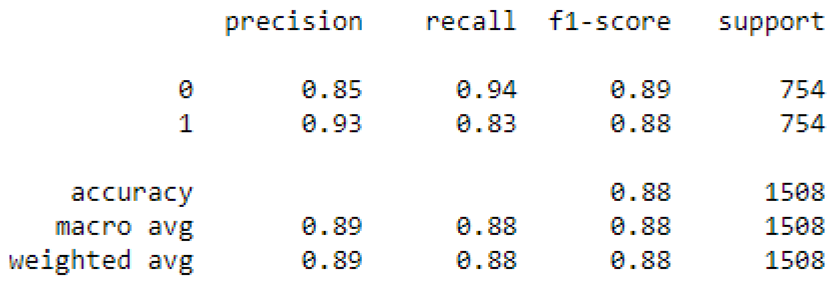
The model has very high train accuracy score (1.0) and test data score is 0.82 . Hence it is overfitted as the difference is more than 10%.

Model 10 - SMOTE KNN

Balanced data is used to build the KNN Model.

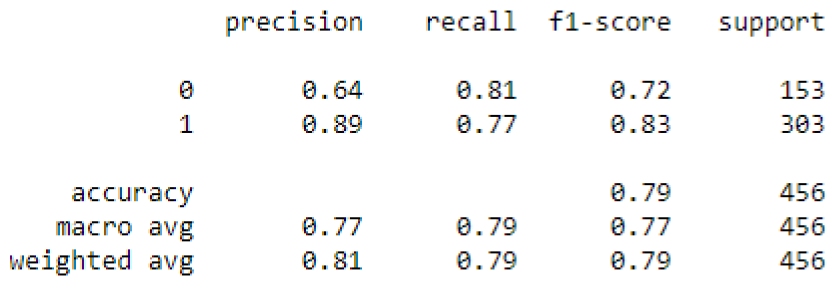
```
KNeighborsClassifier()
```

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.85      0.94      0.89       754
           1       0.93      0.83      0.88       754

    accuracy                           0.88      1508
   macro avg       0.89      0.88      0.88      1508
weighted avg       0.89      0.88      0.88      1508
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.64      0.81      0.72       153
           1       0.89      0.77      0.83       303

    accuracy                           0.79       456
   macro avg       0.77      0.79      0.77       456
weighted avg       0.81      0.79      0.79       456
```

The model has high train accuracy score (0.88) and test data score is 0.79 . Hence it is almost overfitted as the difference is almost 10%.

Model 11- GridSearchCV Naïve Bayes Model

The parameters chosen for GridSearchCV Naïve Bayes are:

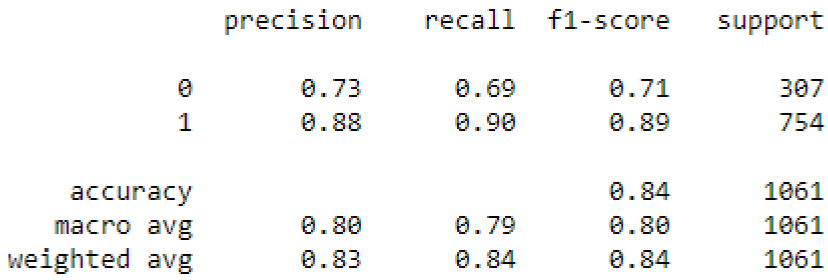
```
GridSearchCV(cv=3, estimator=GaussianNB(), n_jobs=-1,
             param_grid={'var_smoothing': [1e-08, 1e-07, 1e-06, 1e-05, 0.00
01]}, scoring='f1')
```

var_smoothing - Portion of the largest variance of all features that is added to variances for calculation stability.

The best estimator is:

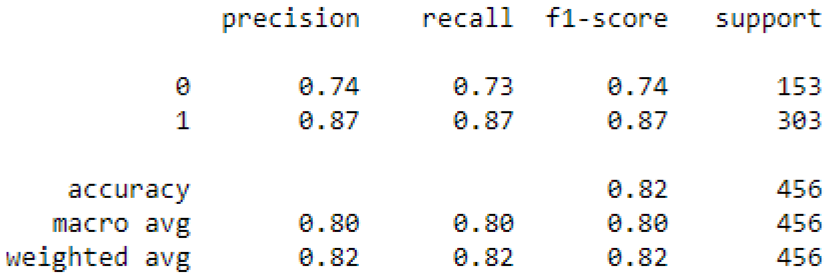
```
{'var_smoothing': 0.0001}

GaussianNB(var_smoothing=0.0001)
```

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.73      0.69      0.71       307
           1       0.88      0.90      0.89       754

    accuracy                           0.84      1061
   macro avg       0.80      0.79      0.80      1061
weighted avg       0.83      0.84      0.84      1061
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.74      0.73      0.74       153
           1       0.87      0.87      0.87       303

    accuracy                           0.82       456
   macro avg       0.80      0.80      0.80       456
weighted avg       0.82      0.82      0.82       456
```

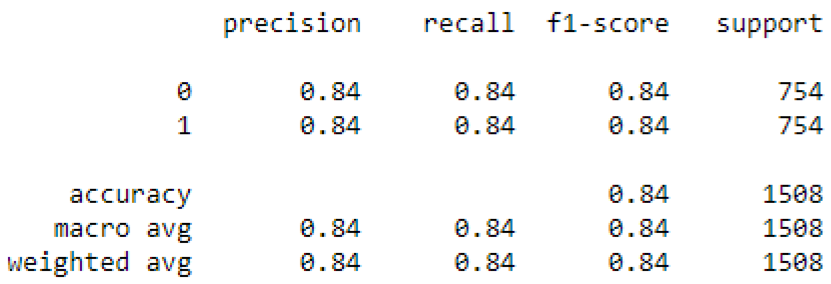
There is no underfitting or overfitting since the accuracy of train data is 0.84 and test data is 0.82 which has a small difference in values between the two.

Model 12 - SMOTE Naïve Bayes Model

Balanced Data is used to build Naïve Bayes Model

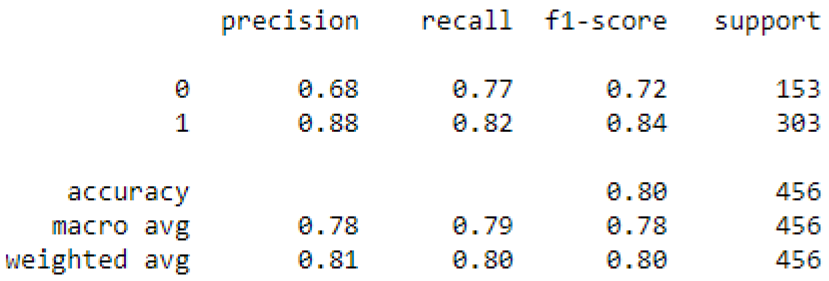
```
GaussianNB()
```

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.84      0.84      0.84       754
           1       0.84      0.84      0.84       754

    accuracy                           0.84      1508
   macro avg       0.84      0.84      0.84      1508
weighted avg       0.84      0.84      0.84      1508
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.68      0.77      0.72       153
           1       0.88      0.82      0.84       303

    accuracy                           0.80       456
   macro avg       0.78      0.79      0.78       456
weighted avg       0.81      0.80      0.80       456
```

There is no underfitting or overfitting since the accuracy of train data is 0.84 and test data is 0.80 which has a small difference in values between the two.

Model 13 - Basic Bagging Classifier Random Forest

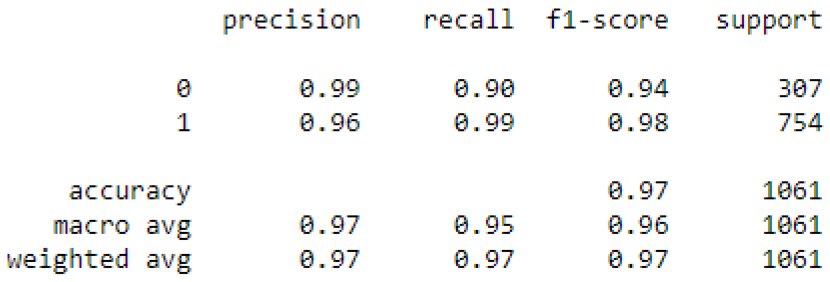
The default Bagging Classifier is applied on Random Forest .

```
BaggingClassifier(base_estimator=RandomForestClassifier(), n_estimators=100
,random_state=1)
```

base_estimator- The base estimator to fit on random subsets of the dataset.

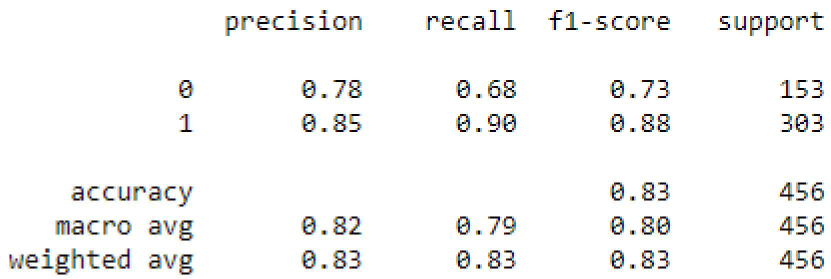
n_estimators **-** The number of trees in the forest.

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.99      0.90      0.94       307
           1       0.96      0.99      0.98       754

    accuracy                           0.97      1061
   macro avg       0.97      0.95      0.96      1061
weighted avg       0.97      0.97      0.97      1061
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.78      0.68      0.73       153
           1       0.85      0.90      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.79      0.80       456
weighted avg       0.83      0.83      0.83       456
```

The model has very high train accuracy score (0.97) and test data score is 0.83 . Hence it is overfitted as the difference is more than 10%.

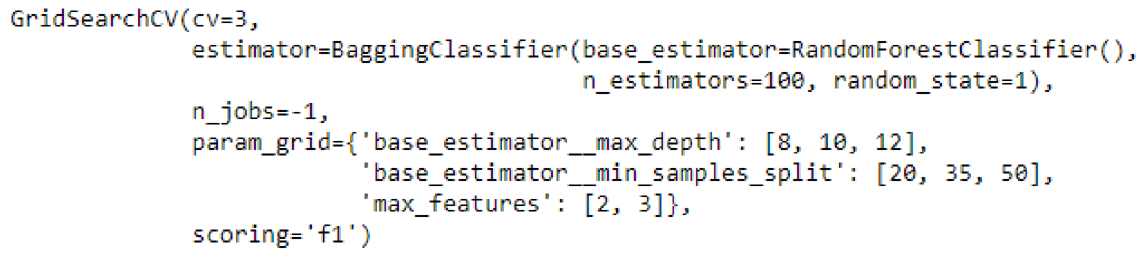
Feature Importance

The feature importance of this model is :

```
                          Feature_Imp
age                          0.201741
Europe                       0.193012
Hague                        0.176092
Blair                        0.138551
economic.cond.national       0.099165
economic.cond.household      0.080253
political.knowledge          0.076198
gender_male                  0.034988
```

Model 14 - GridSearchCV Bagging Classifier Random Forest

The parameters chosen for GridSearchCV Bagging Classifier applied on Random Forest(RF) are:



```
GridSearchCV(cv=3,
             estimator=BaggingClassifier(base_estimator=RandomForestClassifier(),
                                         n_estimators=100, random_state=1),
             n_jobs=-1,
             param_grid={'base_estimator__max_depth': [8, 10, 12],
                         'base_estimator__min_samples_split': [20, 35, 50],
                         'max_features': [2, 3]},
             scoring='f1')
```

base_estimator__max_depth -The maximum depth of the tree

base_estimator__min_samples_split -The minimum number of samples required to split an internal node max_features -The number of features to consider when looking for the best split

The best estimator is

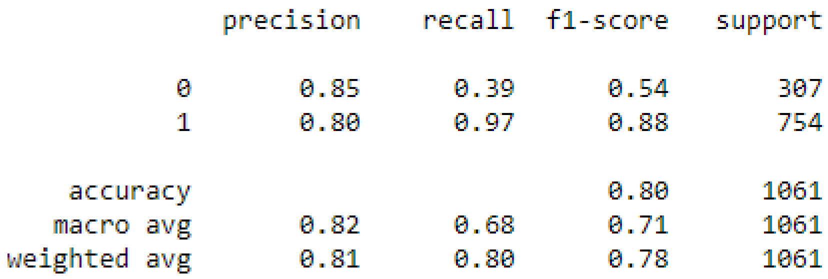
```
{'base_estimator__max_depth': 8, 'base_estimator__min_samples_split': 35, '
max_features': 3}

BaggingClassifier(base_estimator=RandomForestClassifier(max_depth=8,
min_samples_split=35), max_features=3, n_estimators=100, random_state=1)
```

X_train number of rows are 1061, min sample split is 2%-3% of training set.

Values of max-depth is suggested to be taken from 8-15 to avoid overfitting and underfitting.

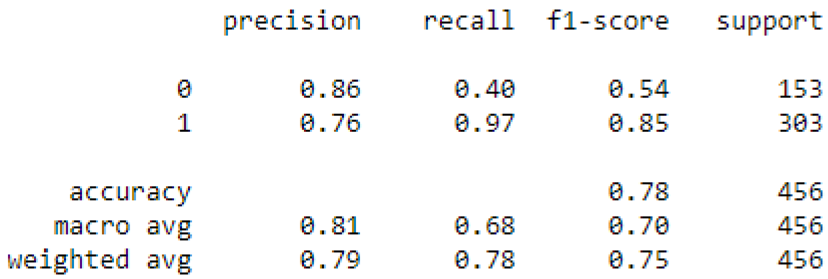
Value of Max feature is taken as square root of number of independent variable to half of the number of independent variables

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.85      0.39      0.54       307
           1       0.80      0.97      0.88       754

    accuracy                           0.80      1061
   macro avg       0.82      0.68      0.71      1061
weighted avg       0.81      0.80      0.78      1061
```

Classification Report for Test Data



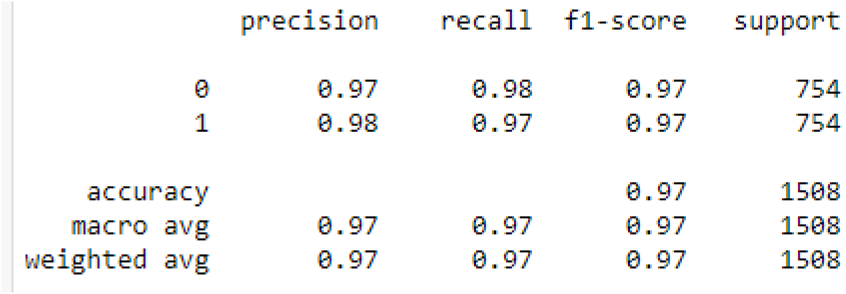
```
              precision    recall  f1-score   support

           0       0.86      0.40      0.54       153
           1       0.76      0.97      0.85       303

    accuracy                           0.78       456
   macro avg       0.81      0.68      0.70       456
weighted avg       0.79      0.78      0.75       456
```

There is no underfitting or overfitting since the accuracy of train data is 0.80 and test data is 0.78 which has a small difference in values between the two.

Model 15 - SMOTE Bagging Classifier Random Forest

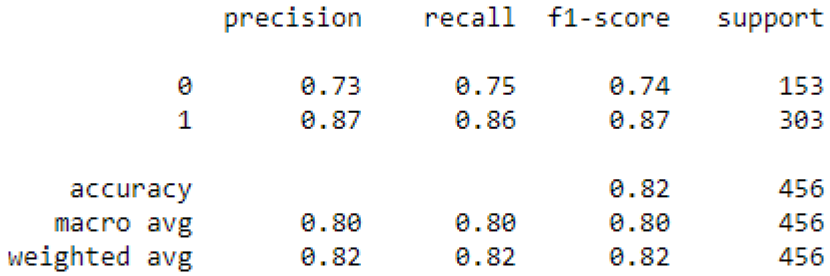
Balanced data Is used for Bagging Classifier with Random Forest

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.97      0.98      0.97       754
           1       0.98      0.97      0.97       754

    accuracy                           0.97      1508
   macro avg       0.97      0.97      0.97      1508
weighted avg       0.97      0.97      0.97      1508
```

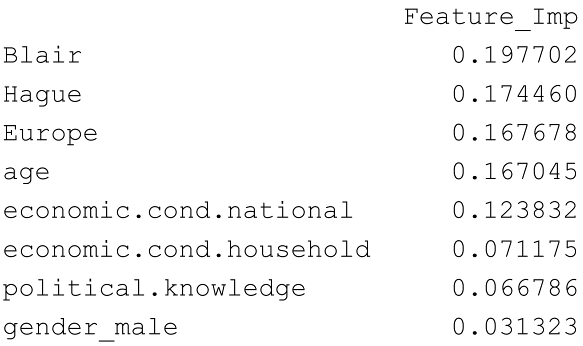
Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.73      0.75      0.74       153
           1       0.87      0.86      0.87       303

    accuracy                           0.82       456
   macro avg       0.80      0.80      0.80       456
weighted avg       0.82      0.82      0.82       456
```

The model has very high train accuracy score (0.97) and test data score is 0.82 . Hence it is overfitted as the difference is more than 10%.

Feature Importance

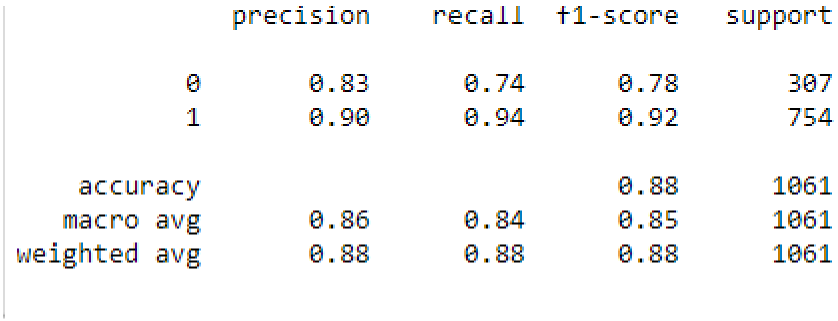


```
                         Feature_Imp
Blair                       0.197702
Hague                       0.174460
Europe                      0.167678
age                         0.167045
economic.cond.national      0.123832
economic.cond.household     0.071175
political.knowledge         0.066786
gender_male                 0.031323
```

Model 16 - Basic Gradient Boosting

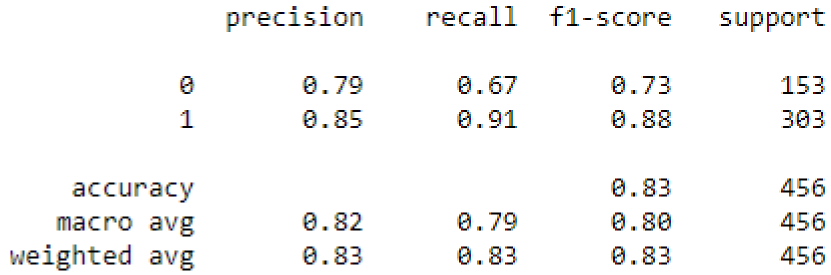
The default Gradient Boosting model is used.

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.83      0.74      0.78       307
           1       0.90      0.94      0.92       754

    accuracy                           0.88      1061
   macro avg       0.86      0.84      0.85      1061
weighted avg       0.88      0.88      0.88      1061
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.79      0.67      0.73       153
           1       0.85      0.91      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.79      0.80       456
weighted avg       0.83      0.83      0.83       456
```

There is no underfitting or overfitting since the accuracy of train data is 0.88 and test data is 0.83 which has a small difference in values between the two.

Feature Importance

```
                         Feature_Imp
Hague                       0.368407
Blair                       0.194182
Europe                      0.170843
political.knowledge         0.088971
economic.cond.national      0.077137
age                         0.072937
economic.cond.household     0.026135
gender_male                 0.001389
```

Model 17 - GridSearchCV Gradient Boosting

The parameters chosen for GridSearchCV Gradient Boosting are:

```
GridSearchCV(cv=3, estimator=GradientBoostingClassifier(random_state=1),
             n_jobs=-1,
             param_grid={'criterion': ['friedman_mse', 'squared_error'],
                         'learning_rate': [0.01, 0.1, 0.15],
                         'n_estimators': [50, 100, 150, 200]},
             scoring='f1')
```

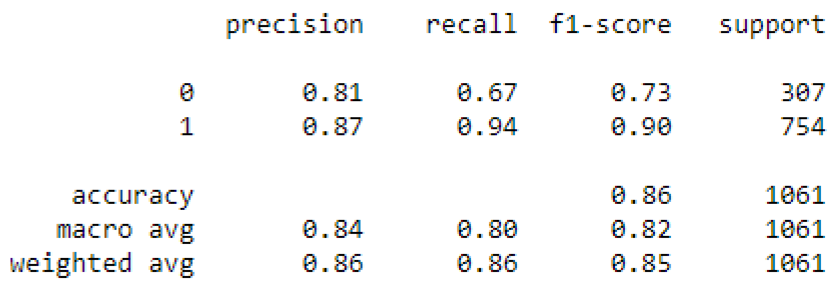
learning_rate- Learning rate shrinks the contribution of each tree by learning_rate.

Criterion - The function to measure the quality of a split

The best estimator is:

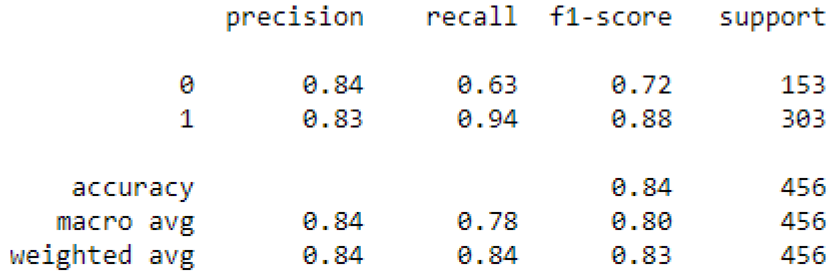
```
{'criterion': 'friedman_mse', 'learning_rate': 0.01, 'n_estimators': 200}

GradientBoostingClassifier(learning_rate=0.01, n_estimators=200, random_sta
te=1)
```

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.81      0.67      0.73       307
           1       0.87      0.94      0.90       754

    accuracy                           0.86      1061
   macro avg       0.84      0.80      0.82      1061
weighted avg       0.86      0.86      0.85      1061
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.84      0.63      0.72       153
           1       0.83      0.94      0.88       303

    accuracy                           0.84       456
   macro avg       0.84      0.78      0.80       456
weighted avg       0.84      0.84      0.83       456
```

There is no underfitting or overfitting since the accuracy of train data is 0.86 and test data is 0.84 which has a small difference in values between the two.

Feature Importance

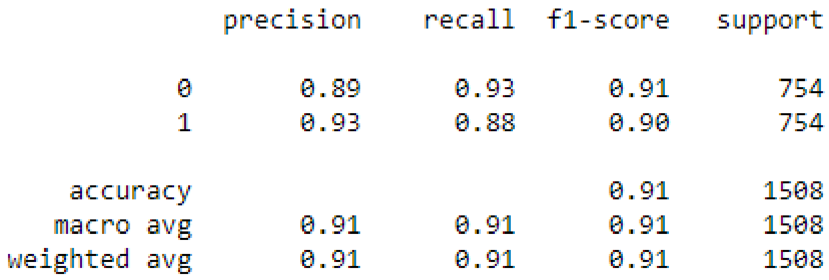
```
                         Feature_Imp
Hague                       0.397301
Blair                       0.207496
Europe                      0.179551
political.knowledge         0.085718
```

```
economic.cond.national      0.073024
age                         0.045363
economic.cond.household     0.011548
gender_male                 0.000000
```

Model 18 - SMOTE Gradient Boosting

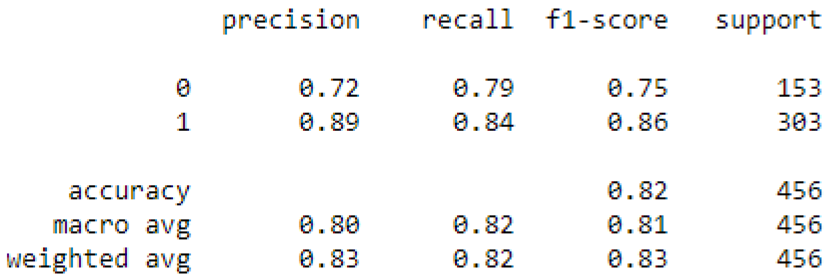
Balanced train data is used to build default Gradient Boosting model.

Classification Report for Train Data



```
              precision    recall  f1-score   support

           0       0.89      0.93      0.91       754
           1       0.93      0.88      0.90       754

    accuracy                           0.91      1508
   macro avg       0.91      0.91      0.91      1508
weighted avg       0.91      0.91      0.91      1508
```

Classification Report for Test Data



```
              precision    recall  f1-score   support

           0       0.72      0.79      0.75       153
           1       0.89      0.84      0.86       303

    accuracy                           0.82       456
   macro avg       0.80      0.82      0.81       456
weighted avg       0.83      0.82      0.83       456
```

The model has high train accuracy score (0.91) and test data score is 0.82 . Hence it is almost overfitted as the difference is almost 10%.

Feature Importance

```
                         Feature_Imp
Blair                       0.410982
Hague                       0.221016
Europe                      0.115826
economic.cond.national      0.112698
age                         0.059080
political.knowledge         0.054870
economic.cond.household     0.019101
gender_male                 0.006426
```

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.

Confusion Matrix *:*

TN / True Negative: when a case was negative and predicted negative(Top Left of Confusion Matrix)

TP / True Positive: when a case was positive and predicted positive(Bottom Rig ht)

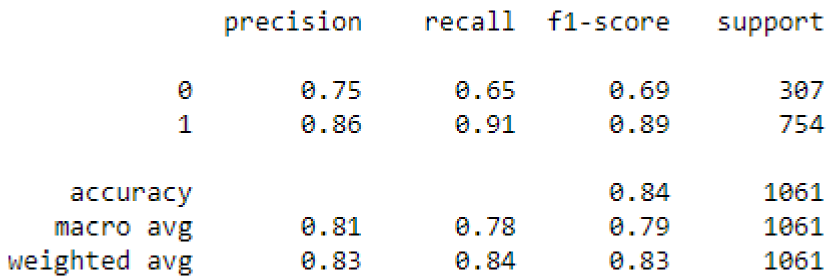
FN / False Negative: when a case was positive but predicted negative(Type 2 er ror) (Bottom Left)

FP / False Positive: when a case was negative but predicted positive(Type 1 err or)(Top Right)

In this problem **False Negative** is an important metric as it denotes claimed cas es as unclaimed ones which generates loss for the Insurance firm.

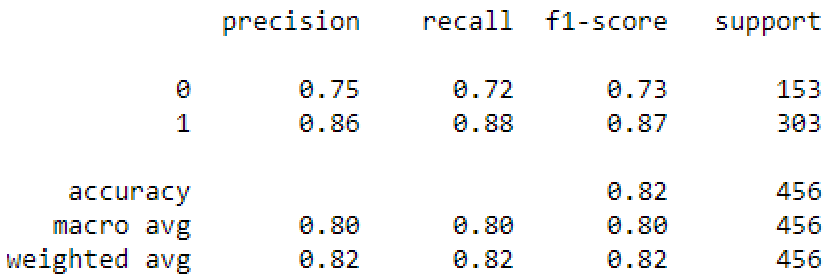
Model 1 - Basic Logistic Regression

Classification Report for Train Data



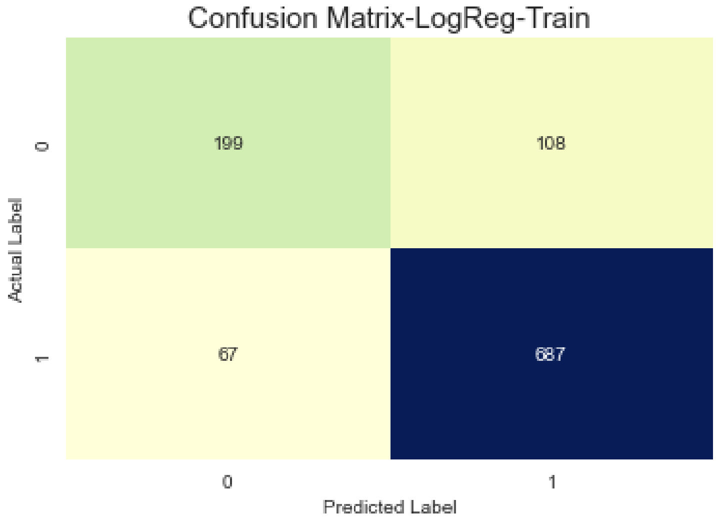
```
              precision    recall  f1-score   support

           0       0.75      0.65      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.84      1061
   macro avg       0.81      0.78      0.79      1061
weighted avg       0.83      0.84      0.83      1061
```

Classification Report for Test Data

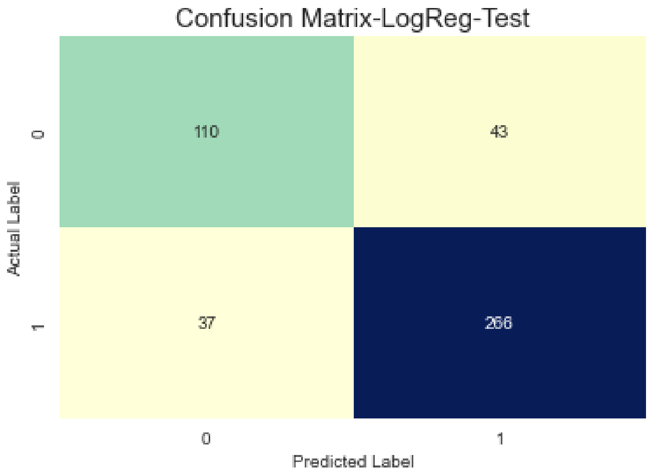


```
              precision    recall  f1-score   support

           0       0.75      0.72      0.73       153
           1       0.86      0.88      0.87       303

    accuracy                           0.82       456
   macro avg       0.80      0.80      0.80       456
weighted avg       0.82      0.82      0.82       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data



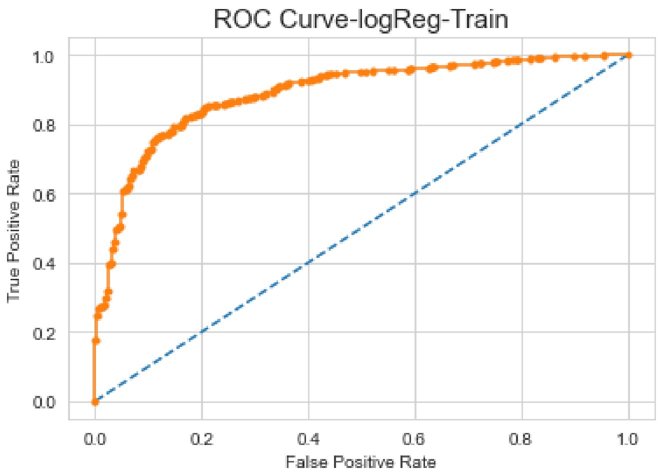
Accuracy Score- Train and Test

Accuracy score of Logistic Regression Trained data is 0.84

Accuracy score of Logistic Regression Tested data is 0.82

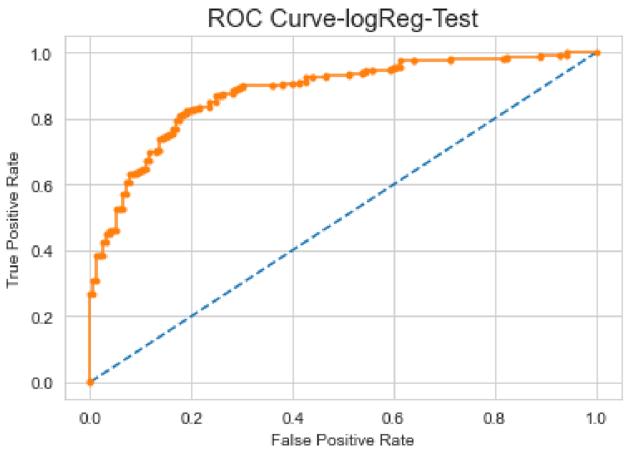
ROC Curve and AUC Score – Train

```
AUC Score for logReg train data: 0.890
```



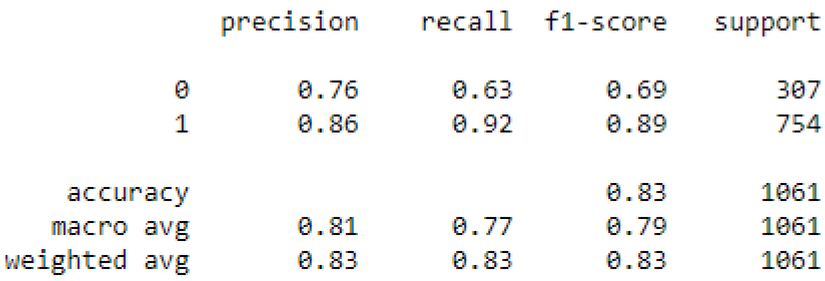
ROC Curve and AUC Score - Test

```
AUC Score for logReg test data: 0.879
```



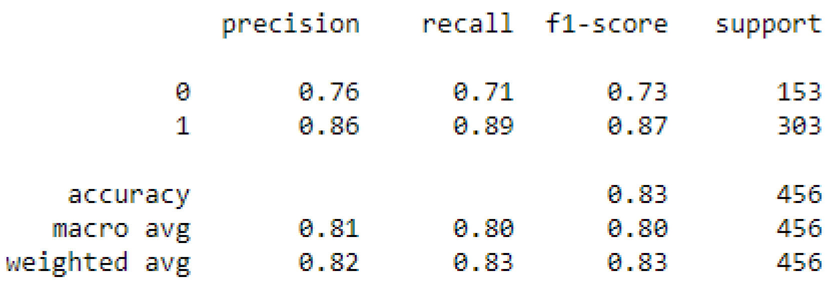
Model 5 - GridSearchCV Logistic Regression

Classification Report for Train Data



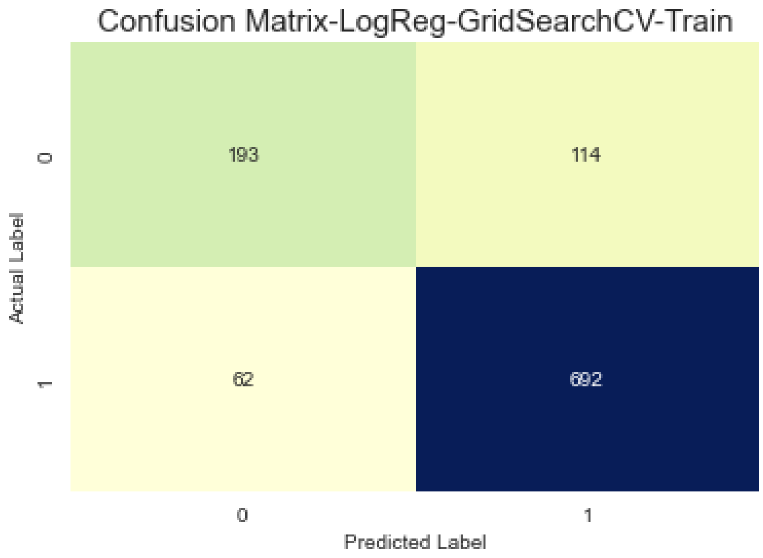
```
              precision    recall  f1-score   support

           0       0.76      0.63      0.69       307
           1       0.86      0.92      0.89       754

    accuracy                           0.83      1061
   macro avg       0.81      0.77      0.79      1061
weighted avg       0.83      0.83      0.83      1061
```

Classification Report for Test Data

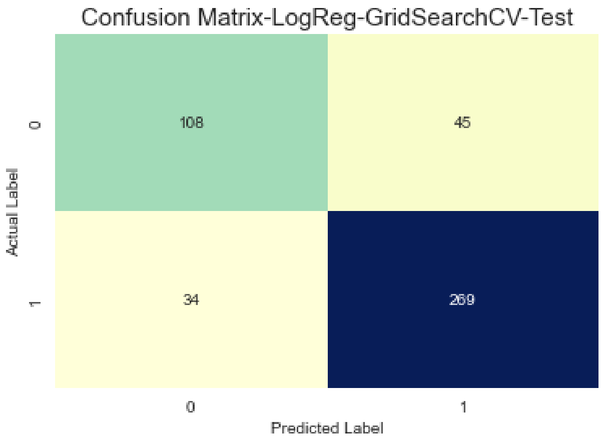


```
              precision    recall  f1-score   support

           0       0.76      0.71      0.73       153
           1       0.86      0.89      0.87       303

    accuracy                           0.83       456
   macro avg       0.81      0.80      0.80       456
weighted avg       0.82      0.83      0.83       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

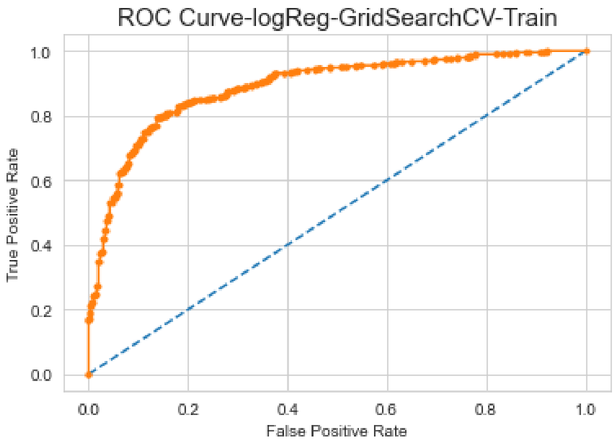


Accuracy Score- Train and Test

```
Accuracy score of Logistic Regression GridSearchCV Trained data is  0.83
Accuracy score of Logistic Regression GridSearchCV Tested data is  0.83
```

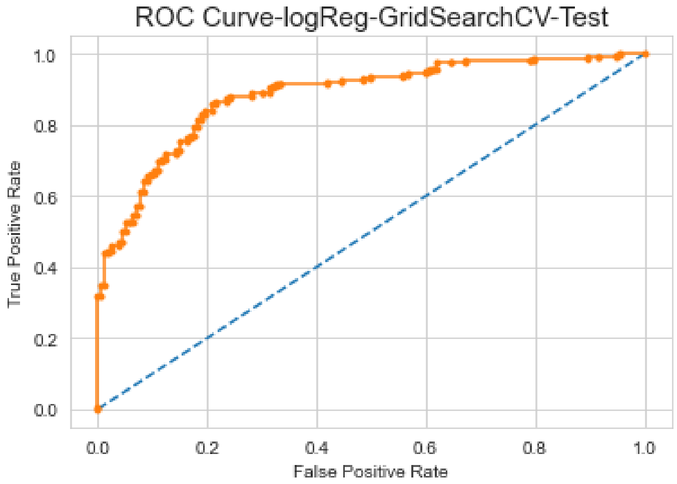
ROC Curve and AUC Score – Train

```
AUC Score for logReg GridSearchCV train data: 0.889
```



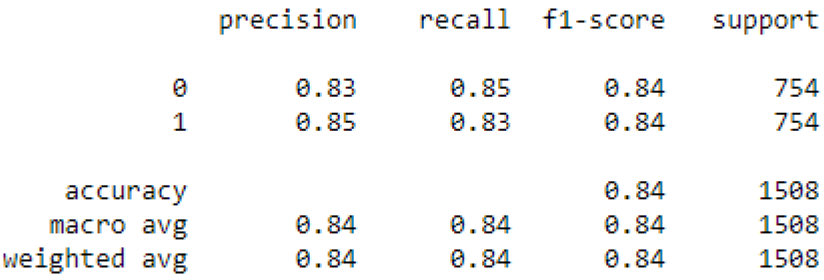
ROC Curve and AUC Score - Test

```
AUC Score for logReg GridSearchCV test data: 0.883
```



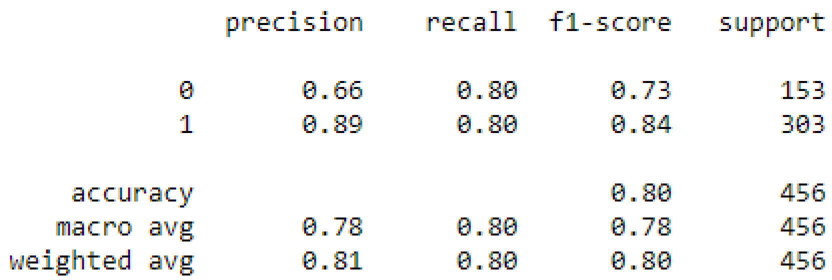
Model 6 - SMOTE Logistic Regression

Classification Report for Train Data



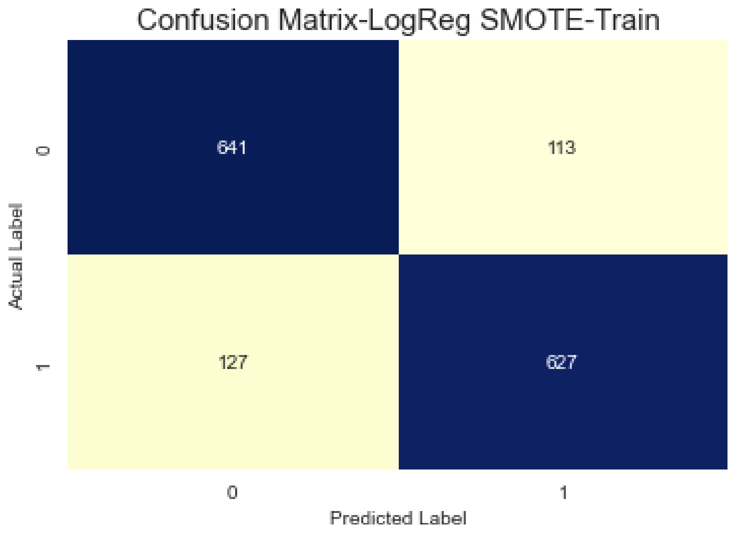
```
              precision    recall  f1-score   support

           0       0.83      0.85      0.84       754
           1       0.85      0.83      0.84       754

    accuracy                           0.84      1508
   macro avg       0.84      0.84      0.84      1508
weighted avg       0.84      0.84      0.84      1508
```

Classification Report for Test Data

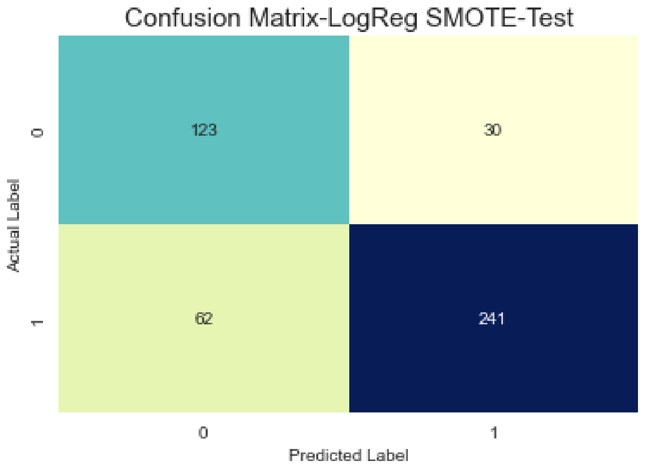


```
              precision    recall  f1-score   support

           0       0.66      0.80      0.73       153
           1       0.89      0.80      0.84       303

    accuracy                           0.80       456
   macro avg       0.78      0.80      0.78       456
weighted avg       0.81      0.80      0.80       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

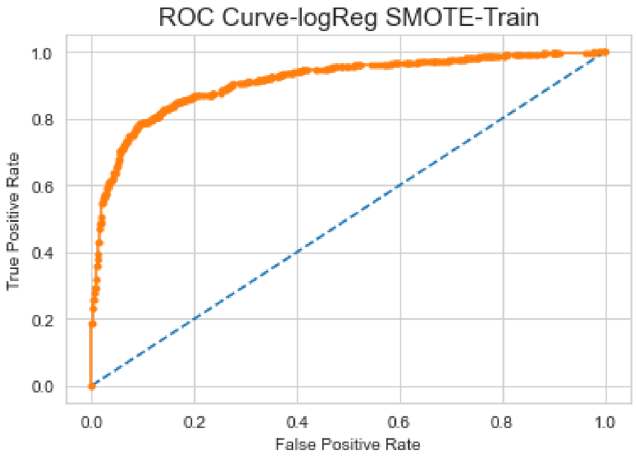


Accuracy Score- Train and Test

```
Accuracy score of Logistic Regression SMOTE Trained data is  0.84
Accuracy score of Logistic Regression SMOTE Tested data is  0.8
```

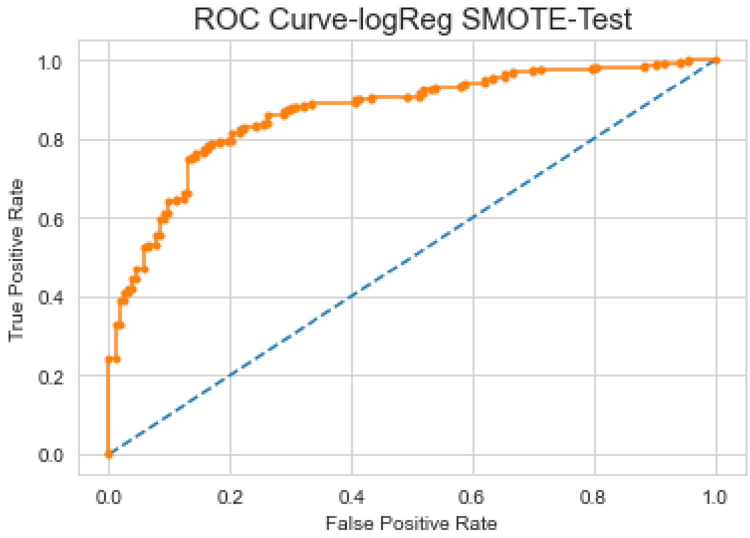
ROC Curve and AUC Score - Train

```
AUC Score for logReg SMOTE train data: 0.911
```



ROC Curve and AUC Score - Test

```
AUC Score for logReg SMOTE test data: 0.867
```

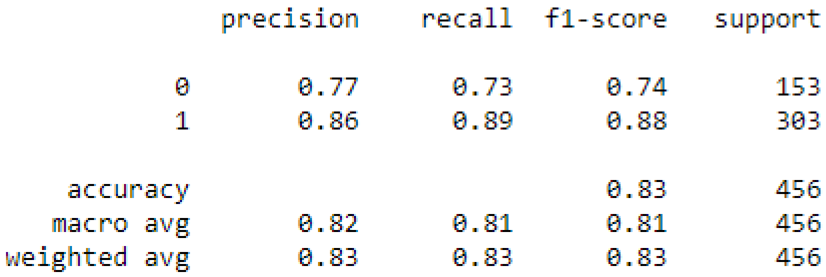


Model 2 - Basic LDA Model

Classification Report for Train Data

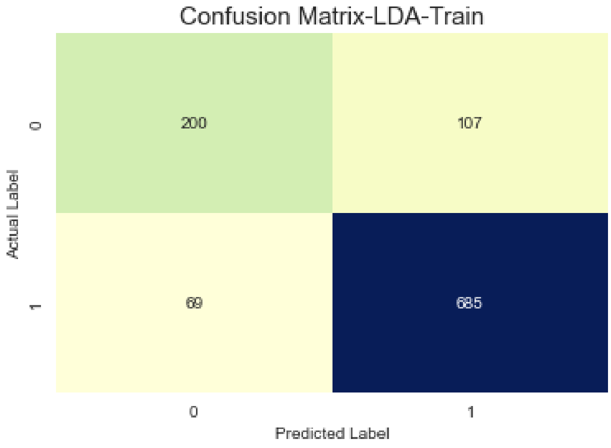
```
              precision    recall  f1-score   support

           0       0.74      0.65      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.83      1061
   macro avg       0.80      0.78      0.79      1061
weighted avg       0.83      0.83      0.83      1061
```

Classification Report for Test Data

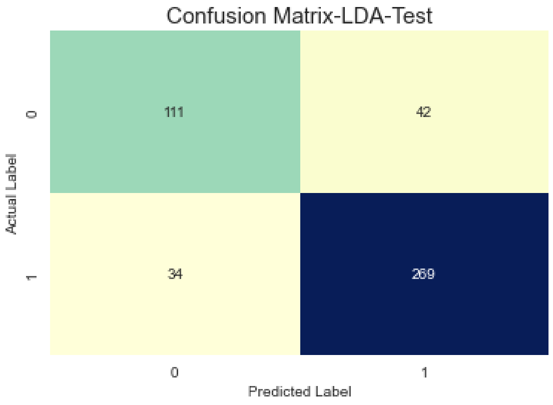


```
              precision    recall  f1-score   support

           0       0.77      0.73      0.74       153
           1       0.86      0.89      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.81      0.81       456
weighted avg       0.83      0.83      0.83       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

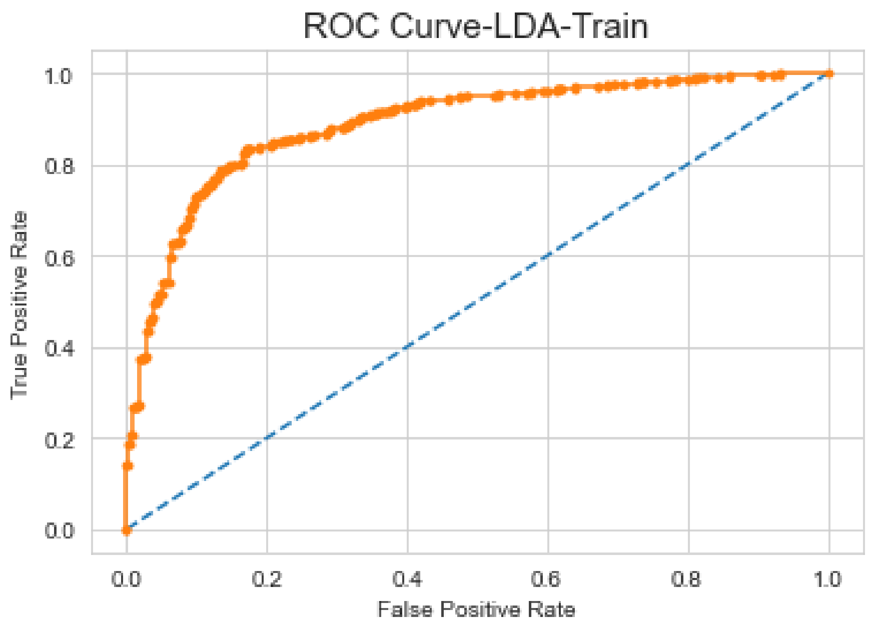


Accuracy Score- Train and Test

```
Accuracy score of LDA Trained data is  0.83
Accuracy score of LDA Tested data is  0.83
```

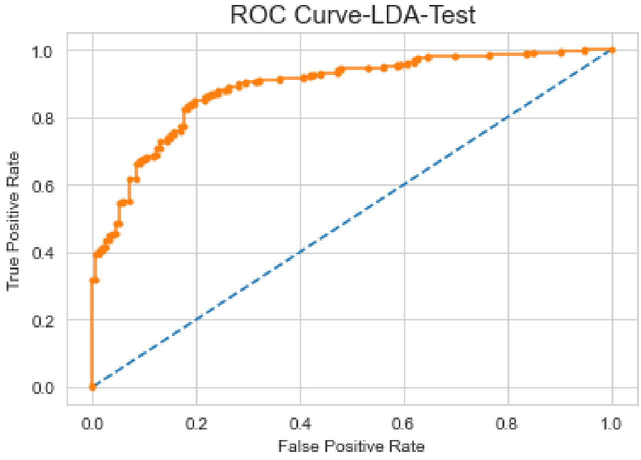
ROC Curve and AUC Score – Train

```
AUC Score for LDA train data: 0.889
```



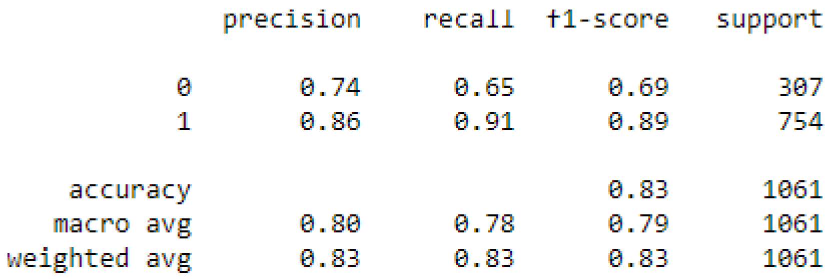
ROC Curve and AUC Score – Test

```
AUC Score for LDA test data: 0.888
```



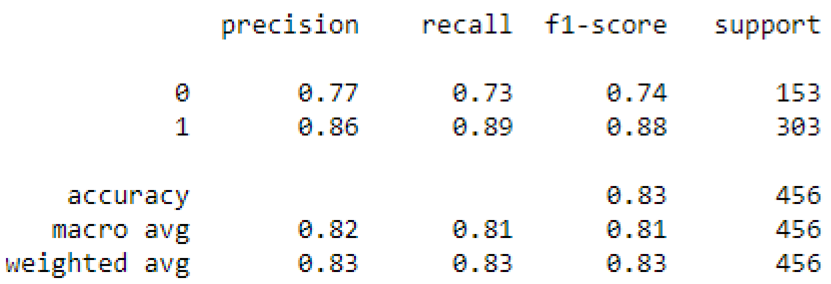
Model 7 - GridSearchCV LDA

Classification Report for Train Data



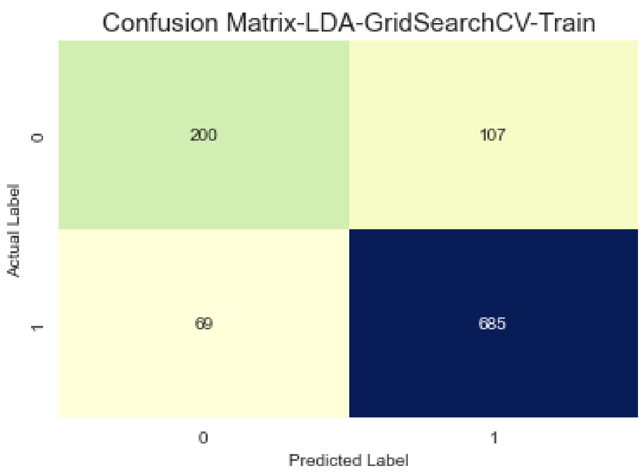
```
              precision    recall  f1-score   support

           0       0.74      0.65      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.83      1061
   macro avg       0.80      0.78      0.79      1061
weighted avg       0.83      0.83      0.83      1061
```

Classification Report for Test Data

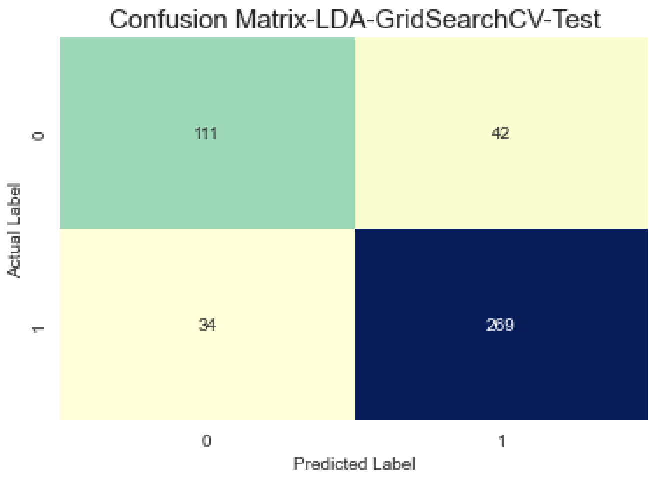


```
              precision    recall  f1-score   support

           0       0.77      0.73      0.74       153
           1       0.86      0.89      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.81      0.81       456
weighted avg       0.83      0.83      0.83       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

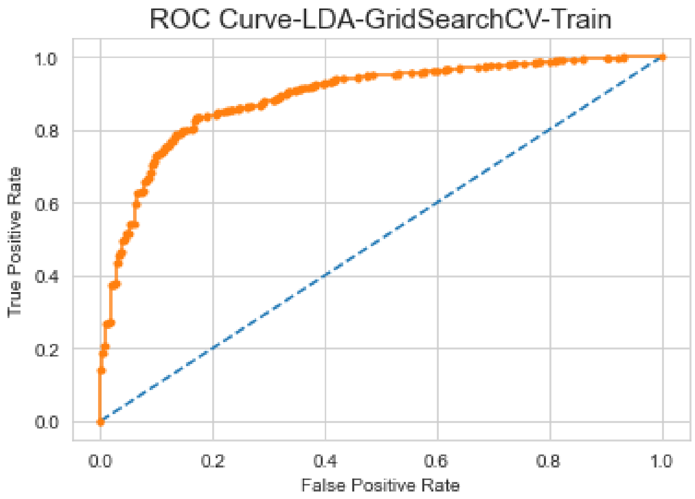


Accuracy Score- Train and Test

```
Accuracy score of LDA GridSearchCV Trained data is  0.83
Accuracy score of LDA GridSearchCV Tested data is  0.83
```

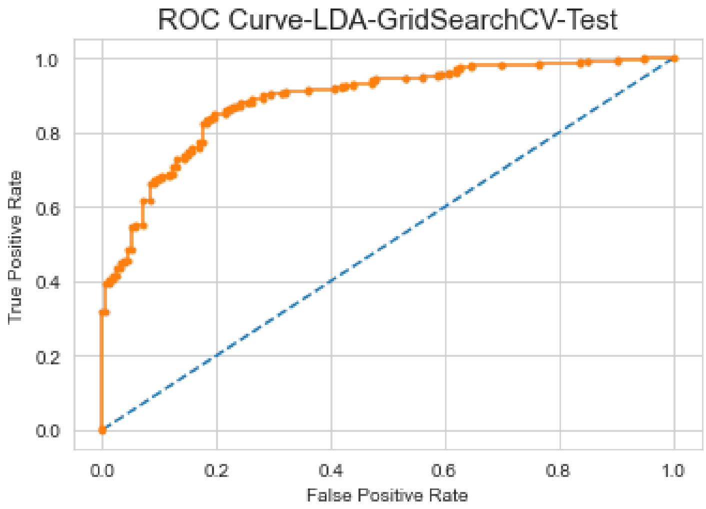
ROC Curve and AUC Score – Train

```
AUC Score for LDA GridSearchCV train data: 0.889
```



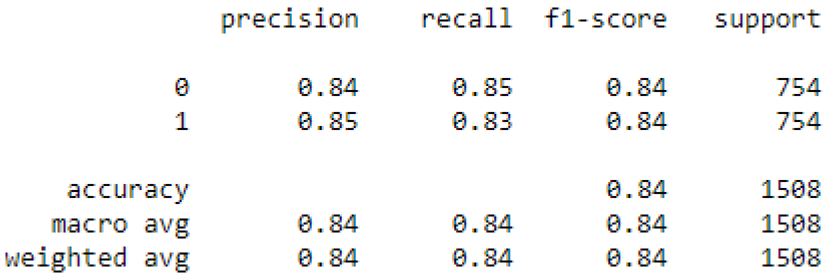
ROC Curve and AUC Score – Test

```
AUC Score for LDA GridSearchCV test data: 0.888
```



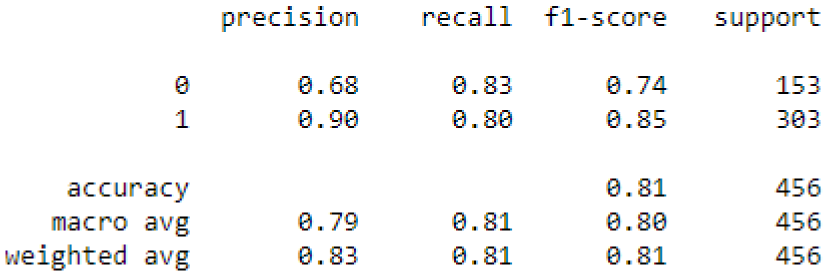
Model 8 - SMOTE LDA Model

Classification Report for Train Data



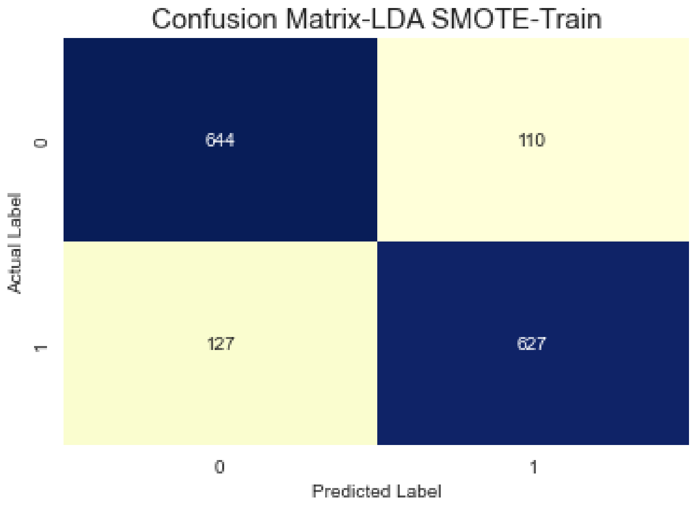
```
              precision    recall  f1-score   support

           0       0.84      0.85      0.84       754
           1       0.85      0.83      0.84       754

    accuracy                           0.84      1508
   macro avg       0.84      0.84      0.84      1508
weighted avg       0.84      0.84      0.84      1508
```

Classification Report for Test Data

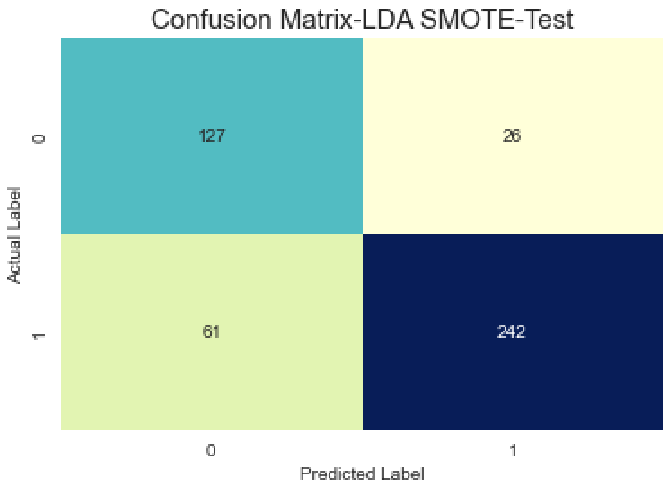


```
              precision    recall  f1-score   support

           0       0.68      0.83      0.74       153
           1       0.90      0.80      0.85       303

    accuracy                           0.81       456
   macro avg       0.79      0.81      0.80       456
weighted avg       0.83      0.81      0.81       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

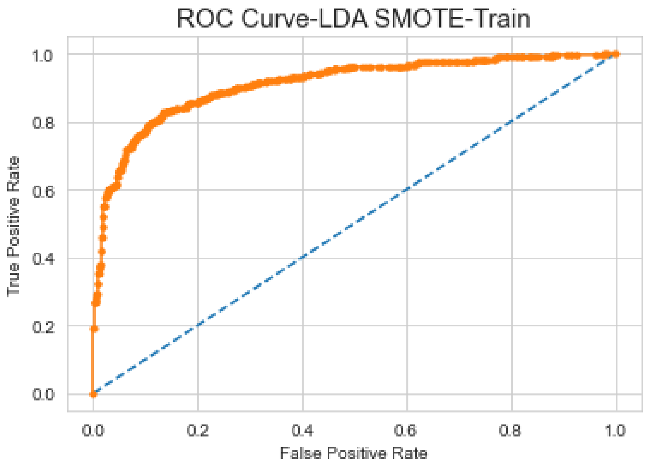


Accuracy Score- Train and Test

```
Accuracy score of LDA SMOTE Trained data is  0.84
Accuracy score of LDA SMOTE Tested data is  0.81
```

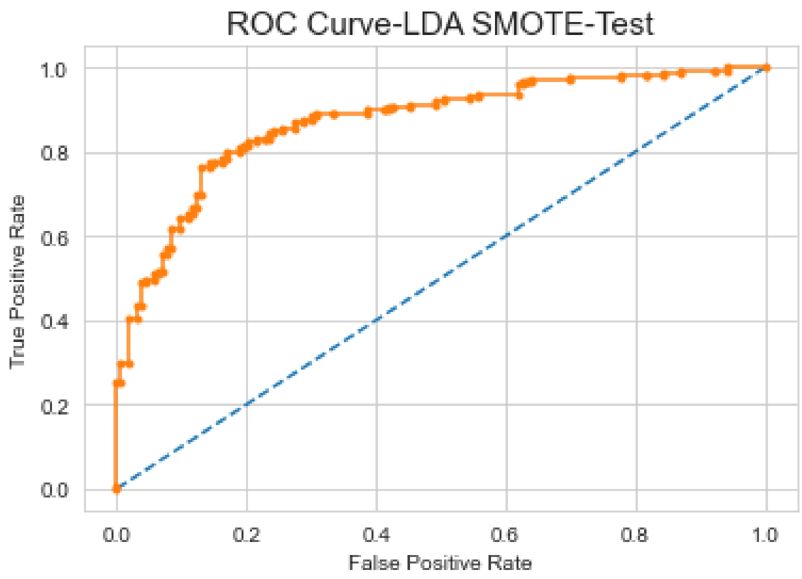
ROC Curve and AUC Score – Train

```
AUC Score for LDA SMOTEtrain data: 0.911
```



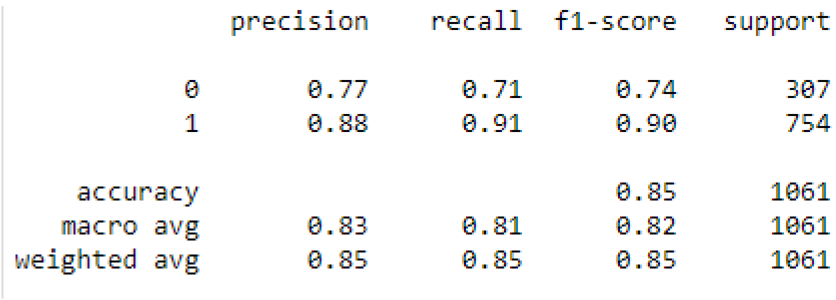
ROC Curve and AUC Score – Test

```
AUC Score for LDA SMOTE test data: 0.871
```



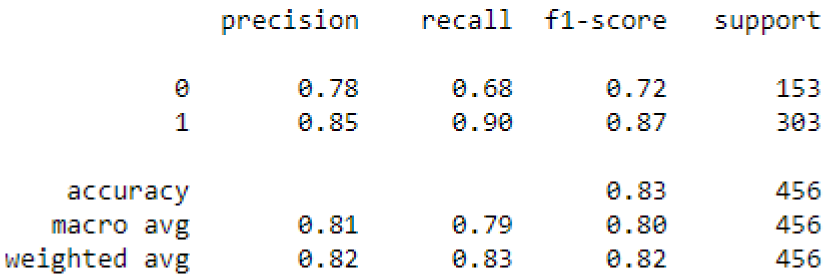
Model 3 - Basic KNN Model

Classification Report for Train Data



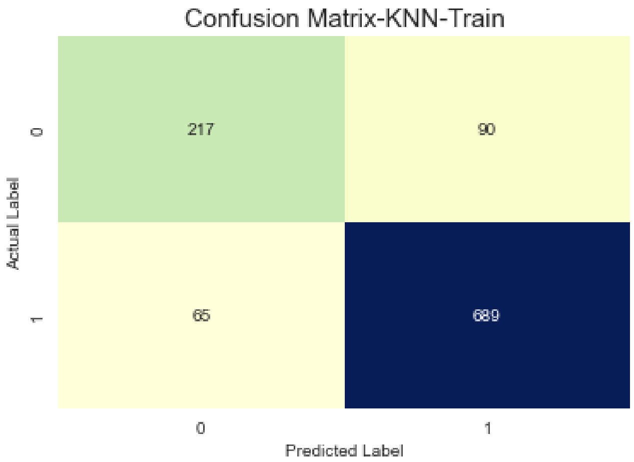
```
              precision    recall  f1-score   support

           0       0.77      0.71      0.74       307
           1       0.88      0.91      0.90       754

    accuracy                           0.85      1061
   macro avg       0.83      0.81      0.82      1061
weighted avg       0.85      0.85      0.85      1061
```

Classification Report for Test Data

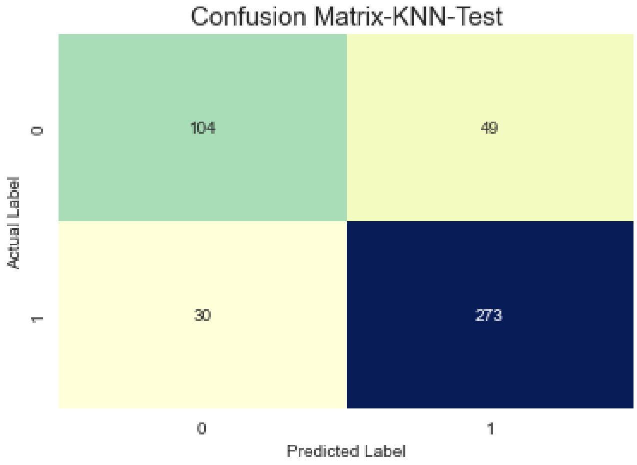


```
              precision    recall  f1-score   support

           0       0.78      0.68      0.72       153
           1       0.85      0.90      0.87       303

    accuracy                           0.83       456
   macro avg       0.81      0.79      0.80       456
weighted avg       0.82      0.83      0.82       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

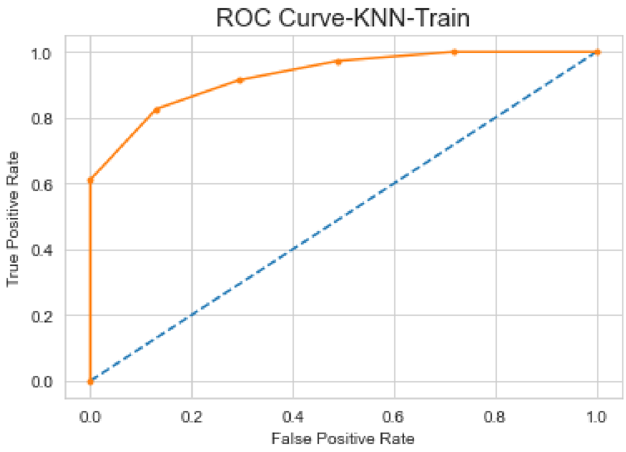


Accuracy Score- Train and Test

```
Accuracy score of KNN Trained data is  0.85
Accuracy score of KNN Tested data is  0.83
```

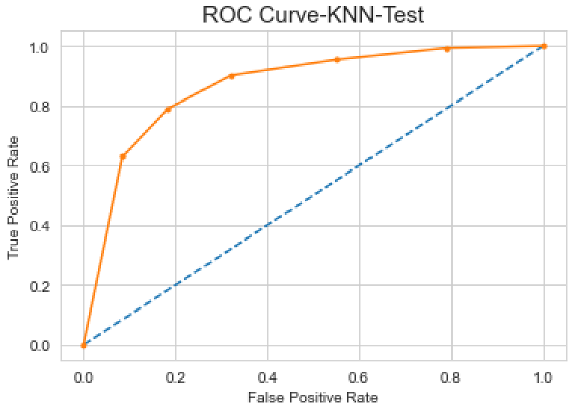
ROC Curve and AUC Score – Train

```
AUC Score for KNN train data: 0.928
```



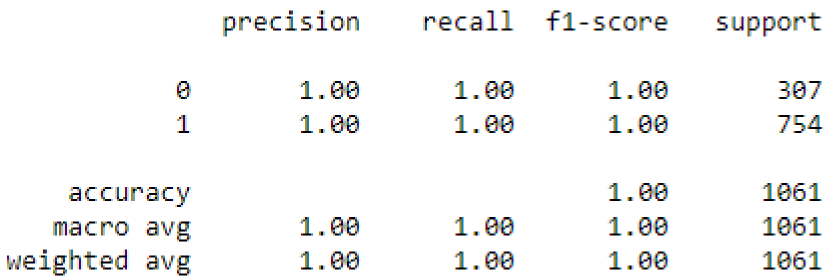
ROC Curve and AUC Score – Test

```
AUC Score for KNN test data: 0.868
```



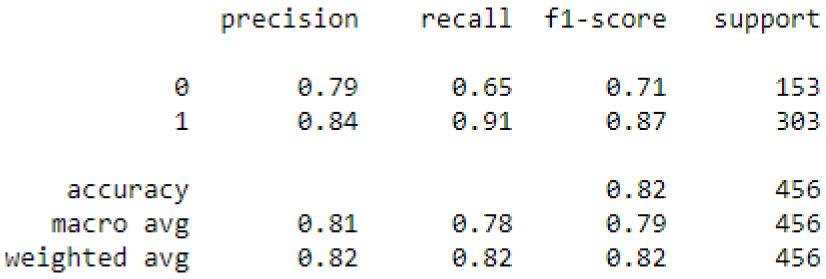
Model 9 - GridSearchCV KNN Model

Classification Report for Train Data



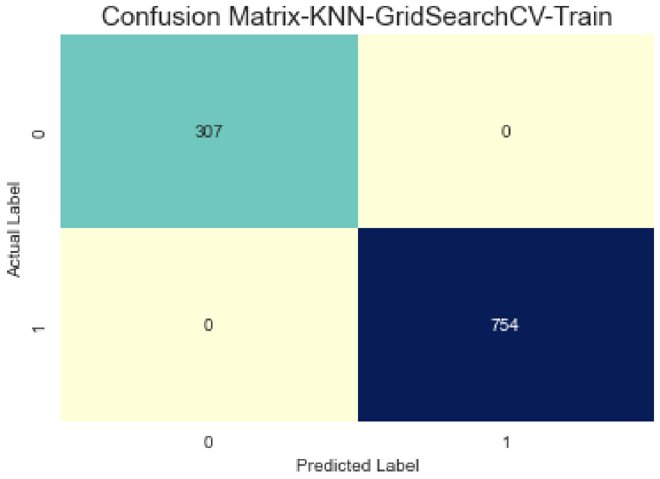
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       307
           1       1.00      1.00      1.00       754

    accuracy                           1.00      1061
   macro avg       1.00      1.00      1.00      1061
weighted avg       1.00      1.00      1.00      1061
```

Classification Report for Test Data

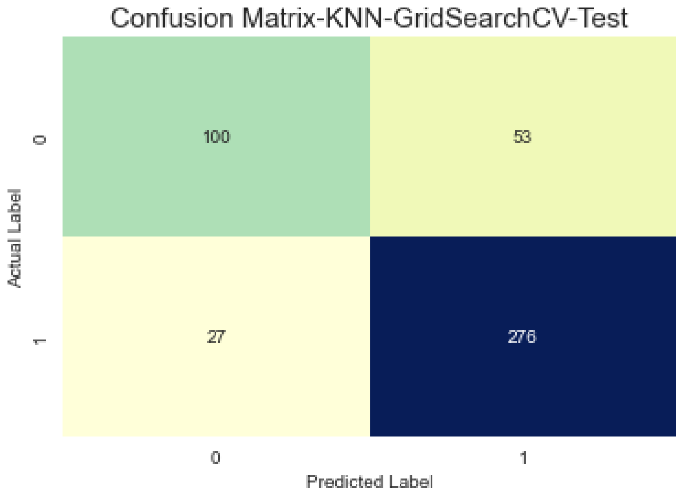


```
              precision    recall  f1-score   support

           0       0.79      0.65      0.71       153
           1       0.84      0.91      0.87       303

    accuracy                           0.82       456
   macro avg       0.81      0.78      0.79       456
weighted avg       0.82      0.82      0.82       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

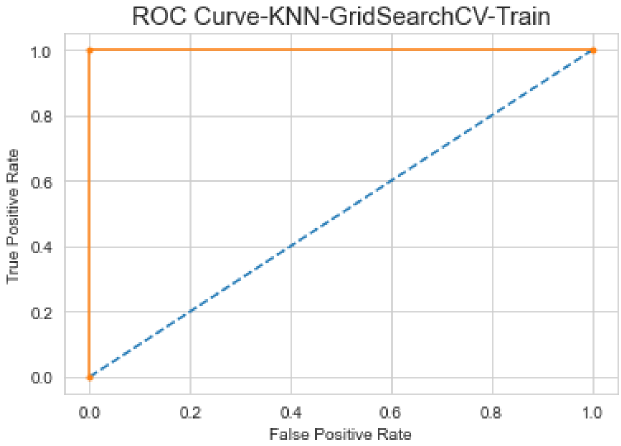


Accuracy Score- Train and Test

```
Accuracy score of KNN GridSearchCV Trained data is  1.0
Accuracy score of KNN GridSearchCV Tested data is  0.82
```

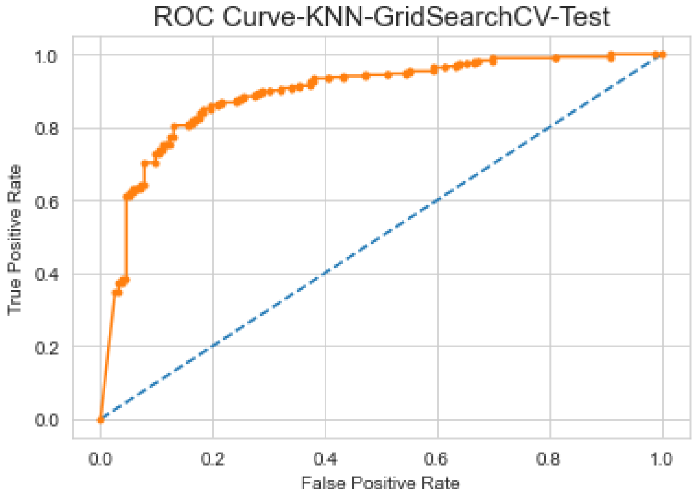
ROC Curve and AUC Score – Train

```
AUC Score for KNN GridSearchCV train data: 1.000
```



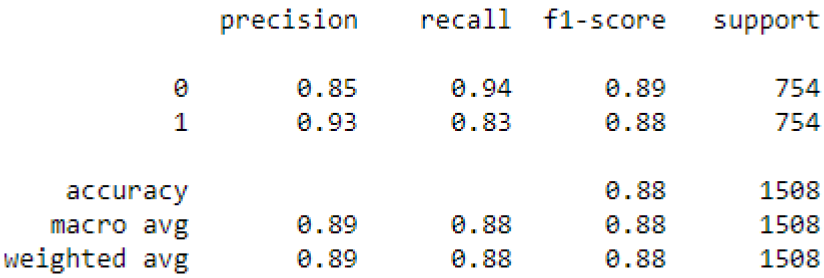
ROC Curve and AUC Score – Test

```
AUC Score for KNN GridSearchCV test data: 0.892
```



Model 10 - SMOTE KNN Model

Classification Report for Train Data

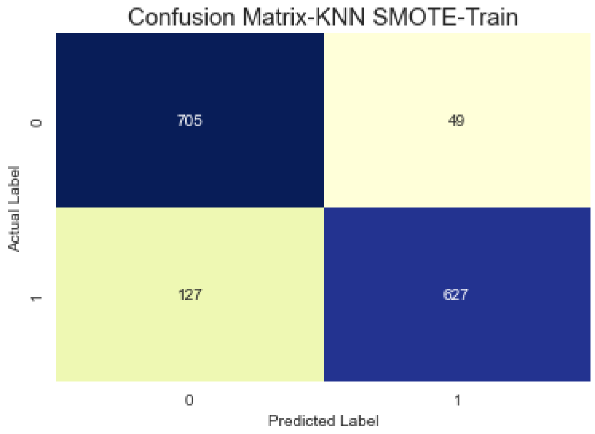


```
              precision    recall  f1-score   support

           0       0.85      0.94      0.89       754
           1       0.93      0.83      0.88       754

    accuracy                           0.88      1508
   macro avg       0.89      0.88      0.88      1508
weighted avg       0.89      0.88      0.88      1508
```

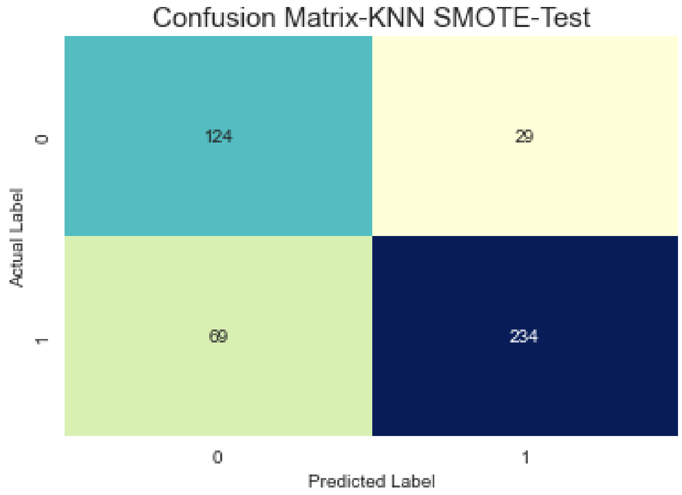
Classification Report for Test Data

```
              precision    recall  f1-score   support

           0       0.64      0.81      0.72       153
           1       0.89      0.77      0.83       303

    accuracy                           0.79       456
   macro avg       0.77      0.79      0.77       456
weighted avg       0.81      0.79      0.79       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

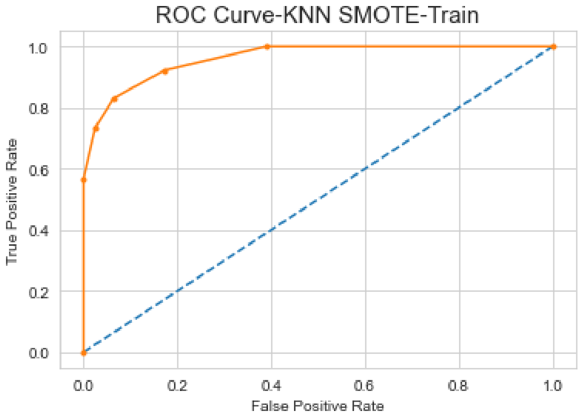


Accuracy Score- Train and Test

```
Accuracy score of KNN SMOTE Trained data is  0.88
Accuracy score of KNN SMOTE Tested data is  0.79
```

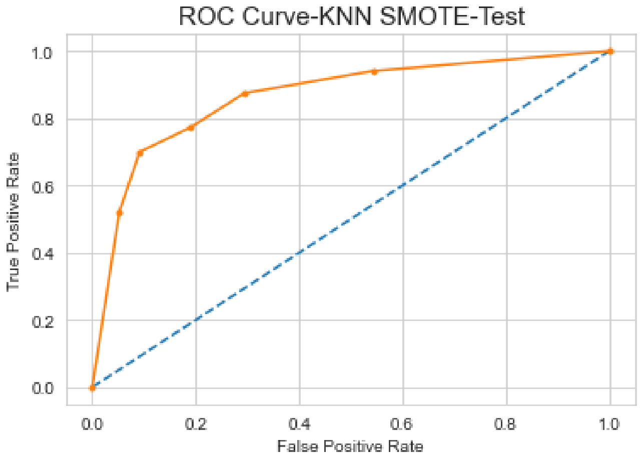
ROC Curve and AUC Score - Train

```
AUC Score for KNN SMOTE train data: 0.961
```



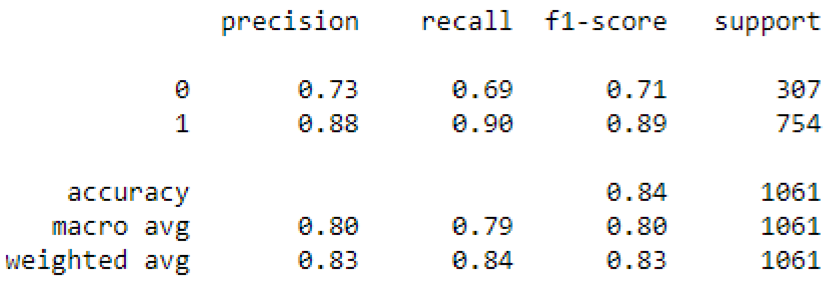
ROC Curve and AUC Score – Test

```
AUC Score for KNN SMOTE test data: 0.865
```



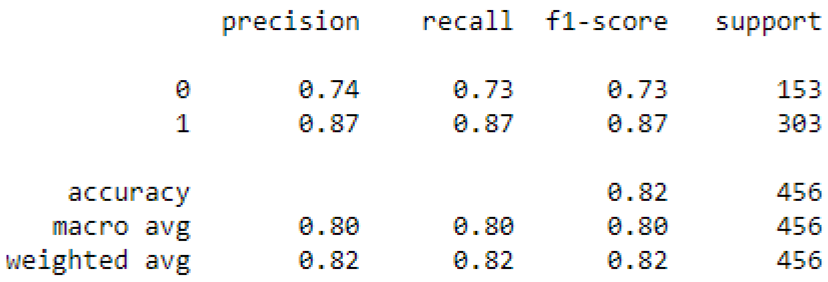
Model 4- Basic Naïve Bayes Model

Classification Report for Train Data



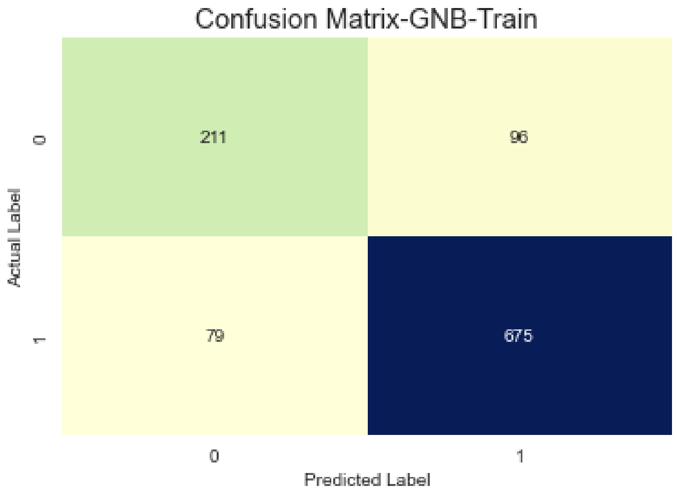
```
              precision    recall  f1-score   support

           0       0.73      0.69      0.71       307
           1       0.88      0.90      0.89       754

    accuracy                           0.84      1061
   macro avg       0.80      0.79      0.80      1061
weighted avg       0.83      0.84      0.83      1061
```

Classification Report for Test Data

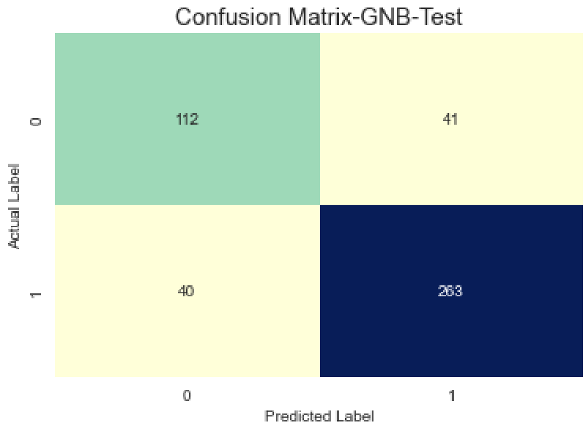


```
              precision    recall  f1-score   support

           0       0.74      0.73      0.73       153
           1       0.87      0.87      0.87       303

    accuracy                           0.82       456
   macro avg       0.80      0.80      0.80       456
weighted avg       0.82      0.82      0.82       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

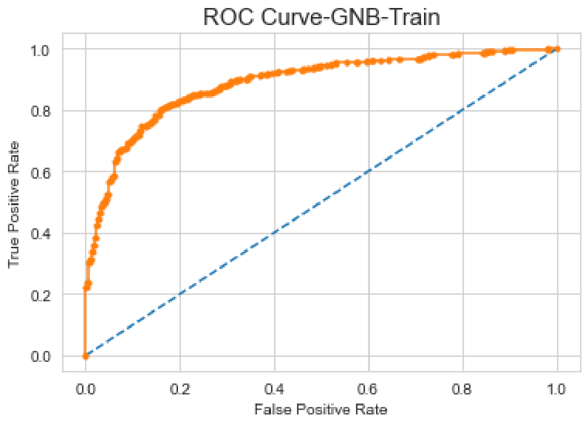


Accuracy Score- Train and Test

```
Accuracy score of GNB Trained data is  0.84
Accuracy score of GNB Tested data is  0.82
```

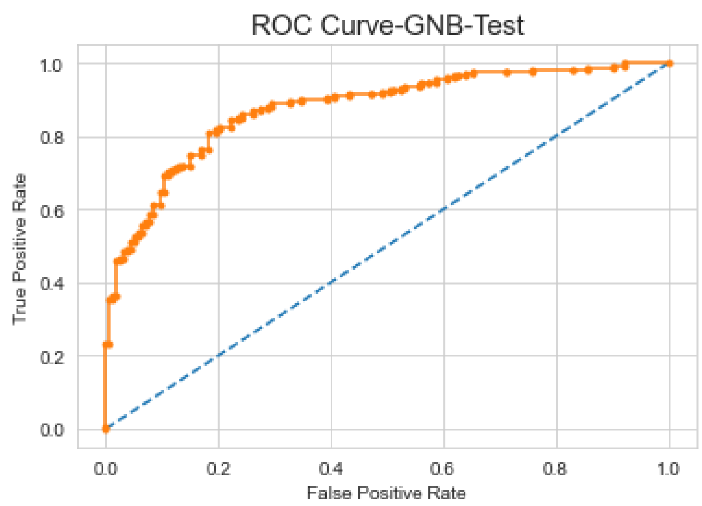
ROC Curve and AUC Score – Train

```
AUC Score for GNB train data: 0.888
```



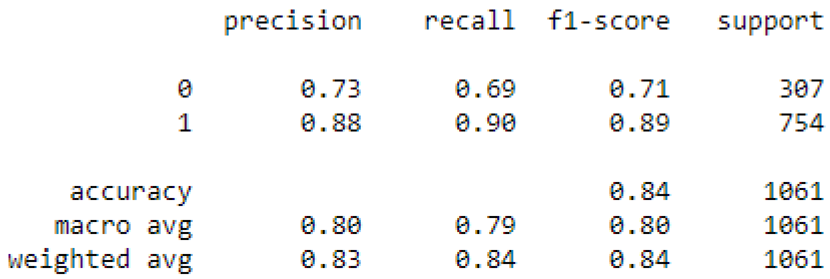
ROC Curve and AUC Score – Test

```
AUC Score for GNB test data: 0.876
```



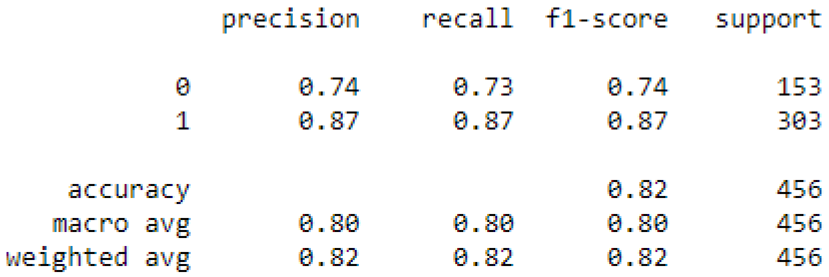
Model 11- GridSearchCV Naïve Bayes Model

Classification Report for Train Data



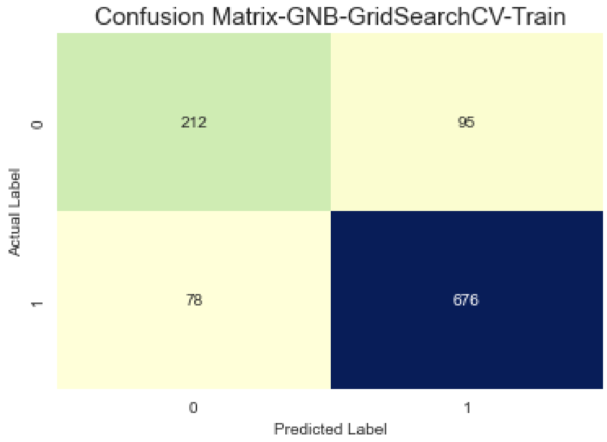
```
              precision    recall  f1-score   support

           0       0.73      0.69      0.71       307
           1       0.88      0.90      0.89       754

    accuracy                           0.84      1061
   macro avg       0.80      0.79      0.80      1061
weighted avg       0.83      0.84      0.84      1061
```

Classification Report for Test Data

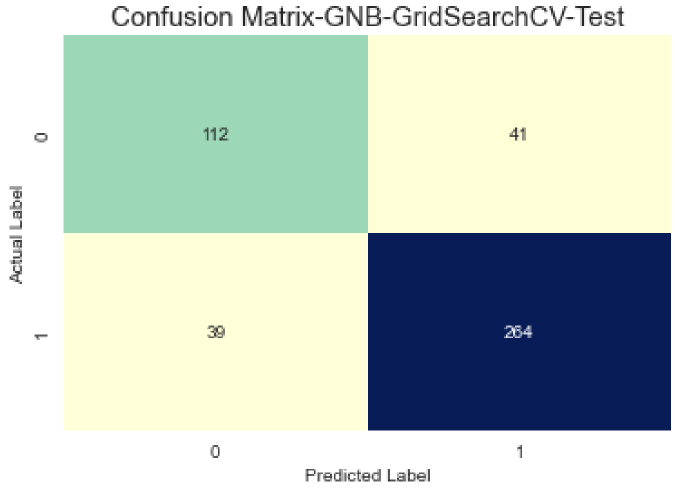


```
              precision    recall  f1-score   support

           0       0.74      0.73      0.74       153
           1       0.87      0.87      0.87       303

    accuracy                           0.82       456
   macro avg       0.80      0.80      0.80       456
weighted avg       0.82      0.82      0.82       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

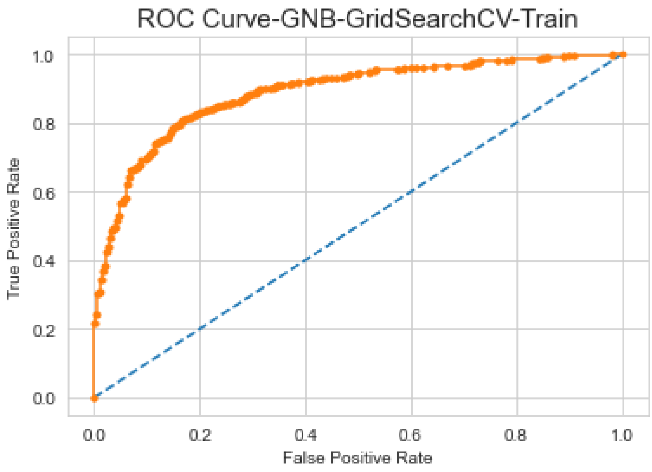


Accuracy Score- Train and Test

```
Accuracy score of GNB GridSearchCV Trained data is  0.84
Accuracy score of GNB GridSearchCV Tested data is  0.82
```

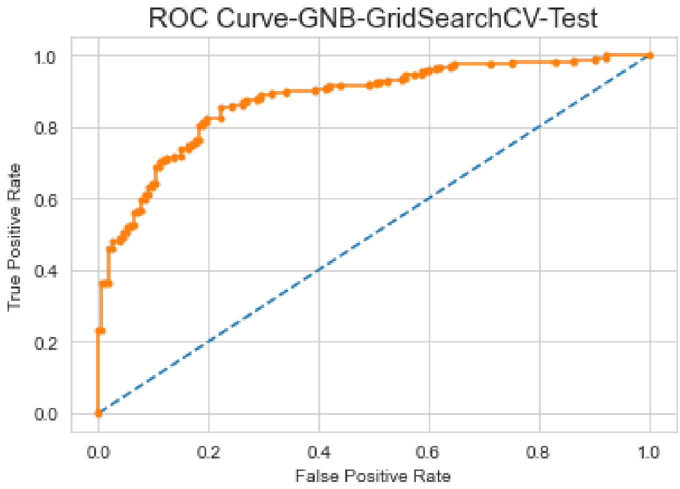
ROC Curve and AUC Score – Train

```
AUC Score for GNB GridSearchCV train data: 0.888
```



ROC Curve and AUC Score – Test

```
AUC Score for GNB GridSearchCV test data: 0.877
```

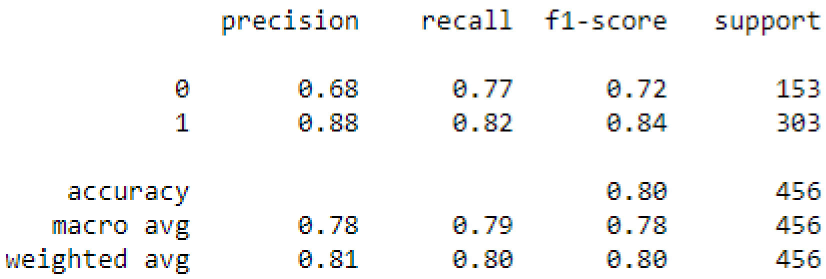


Model 12- SMOTE Naïve Bayes Model

Classification Report for Train Data

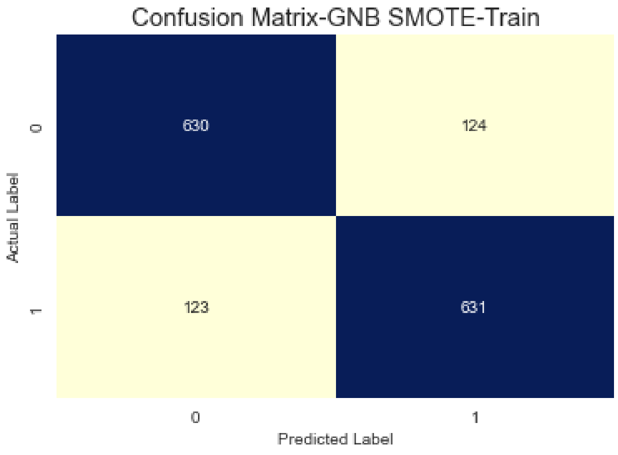
```
              precision    recall  f1-score   support

           0       0.84      0.84      0.84       754
           1       0.84      0.84      0.84       754

    accuracy                           0.84      1508
   macro avg       0.84      0.84      0.84      1508
weighted avg       0.84      0.84      0.84      1508
```

Classification Report for Test Data

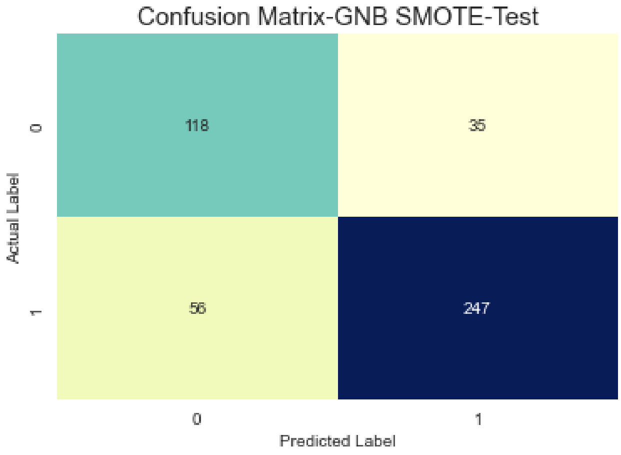


```
              precision    recall  f1-score   support

           0       0.68      0.77      0.72       153
           1       0.88      0.82      0.84       303

    accuracy                           0.80       456
   macro avg       0.78      0.79      0.78       456
weighted avg       0.81      0.80      0.80       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

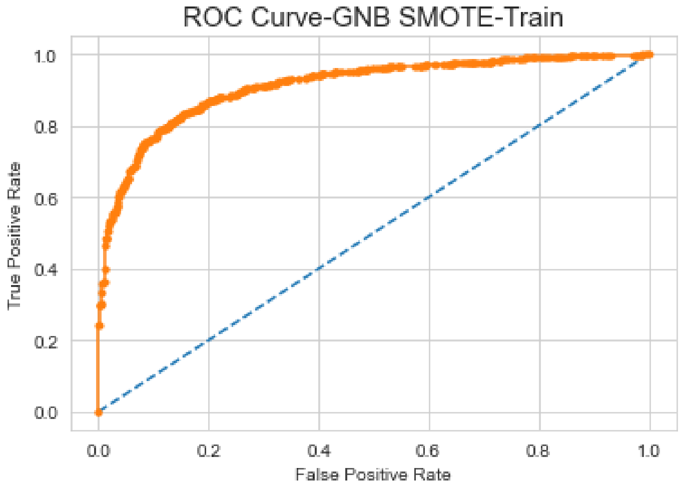


Accuracy Score- Train and Test

```
Accuracy score of GNB SMOTE Trained data is  0.84
Accuracy score of GNB SMOTE Tested data is  0.8
```

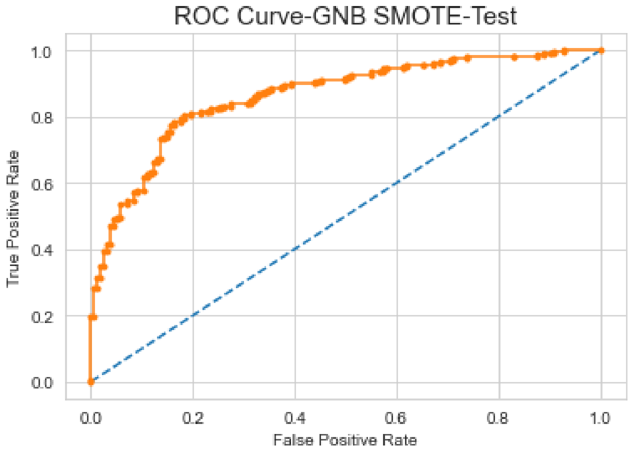
ROC Curve and AUC Score – Train

```
AUC Score for GNB SMOTE train data: 0.912
```



ROC Curve and AUC Score – Test

```
AUC Score for GNB SMOTE test data: 0.863
```

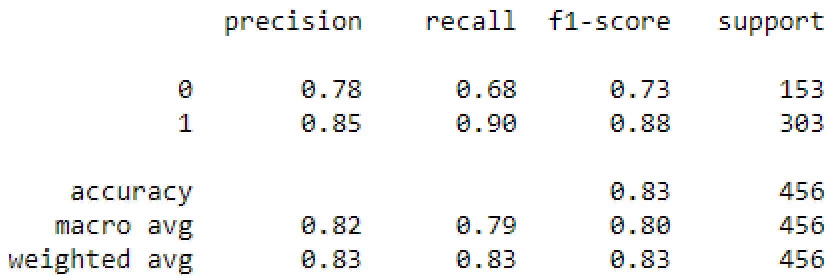


Model 13 – Basic Bagging Classifier Random Forest

Classification Report for Train Data

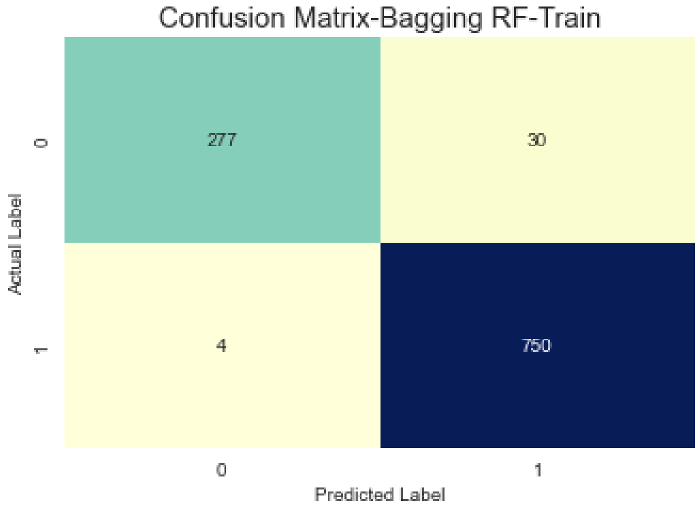
```
              precision    recall  f1-score   support

           0       0.99      0.90      0.94       307
           1       0.96      0.99      0.98       754

    accuracy                           0.97      1061
   macro avg       0.97      0.95      0.96      1061
weighted avg       0.97      0.97      0.97      1061
```

Classification Report for Test Data

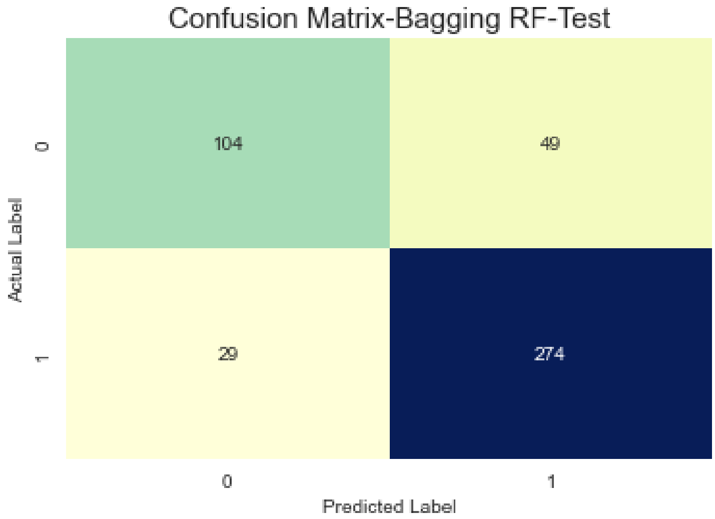


```
              precision    recall  f1-score   support

           0       0.78      0.68      0.73       153
           1       0.85      0.90      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.79      0.80       456
weighted avg       0.83      0.83      0.83       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

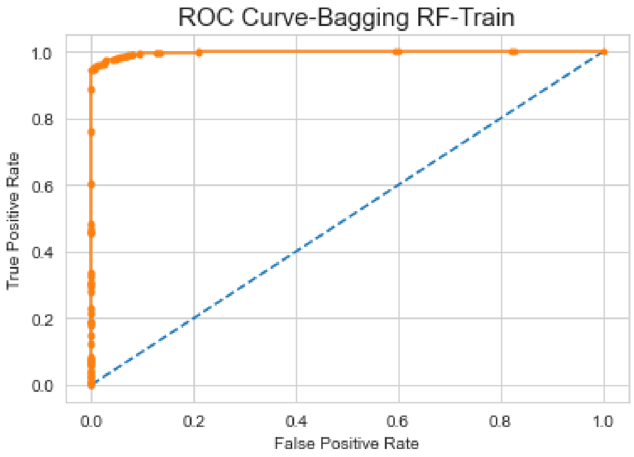


Accuracy Score- Train and Test

```
Accuracy score of Bagging RF Trained data is  0.97
Accuracy score of Bagging RF Tested data is  0.83
```

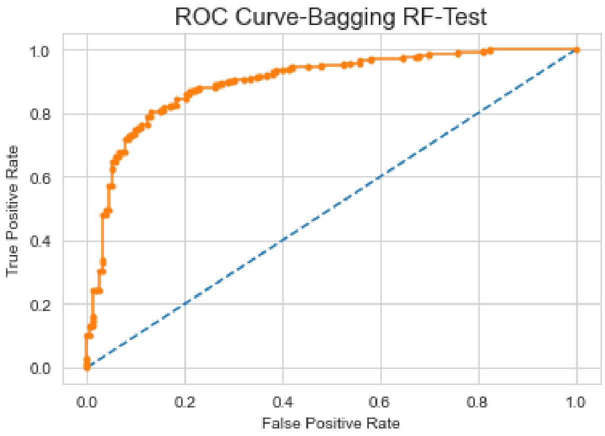
ROC Curve and AUC Score – Train

```
AUC Score for Bagging RF train data: 0.997
```



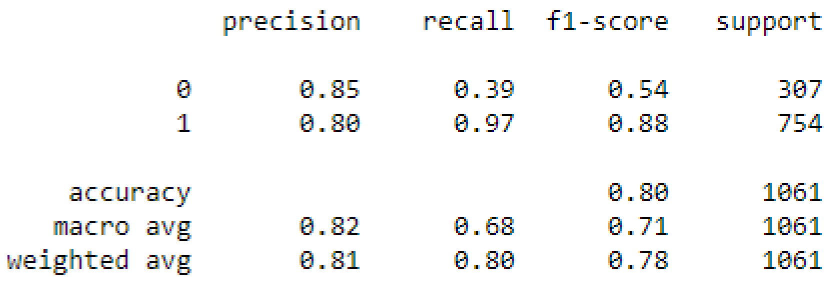
ROC Curve and AUC Score – Test

```
AUC Score for Bagging RF test data: 0.897
```



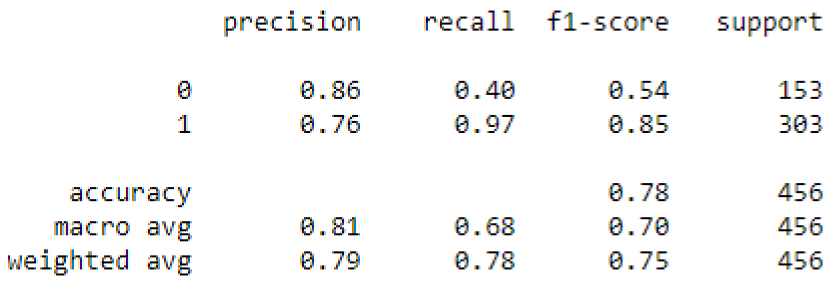
Model 14 - GridSeacrhCV Bagging RF Model

Classification Report for Train Data



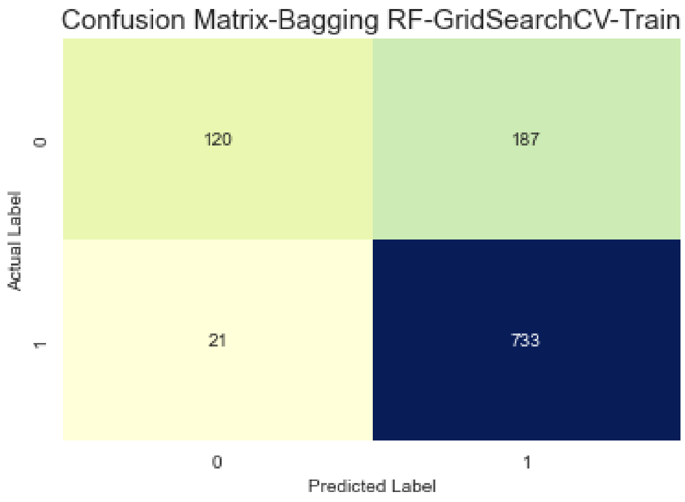
```
              precision    recall  f1-score   support

           0       0.85      0.39      0.54       307
           1       0.80      0.97      0.88       754

    accuracy                           0.80      1061
   macro avg       0.82      0.68      0.71      1061
weighted avg       0.81      0.80      0.78      1061
```

Classification Report for Test Data

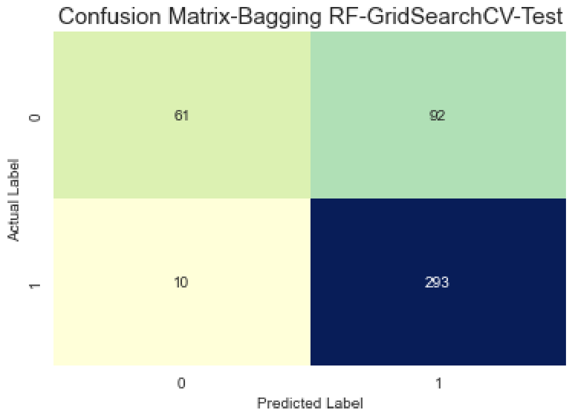


```
              precision    recall  f1-score   support

           0       0.86      0.40      0.54       153
           1       0.76      0.97      0.85       303

    accuracy                           0.78       456
   macro avg       0.81      0.68      0.70       456
weighted avg       0.79      0.78      0.75       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

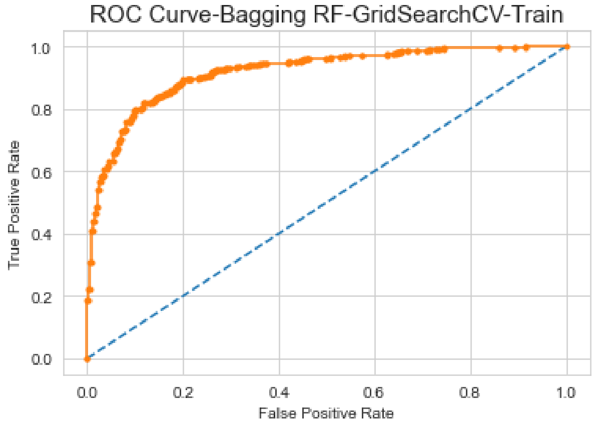


Accuracy Score- Train and Test

```
Accuracy score of Bagging RF GridSearchCV Trained data is  0.8
Accuracy score of Bagging RF GridSearchCV Tested data is  0.78
```

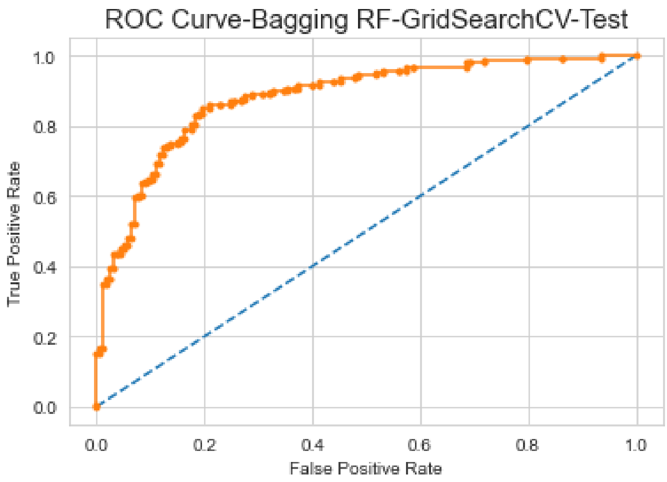
ROC Curve and AUC Score – Train

```
AUC Score for Bagging RF GridSearchCV train data: 0.918
```



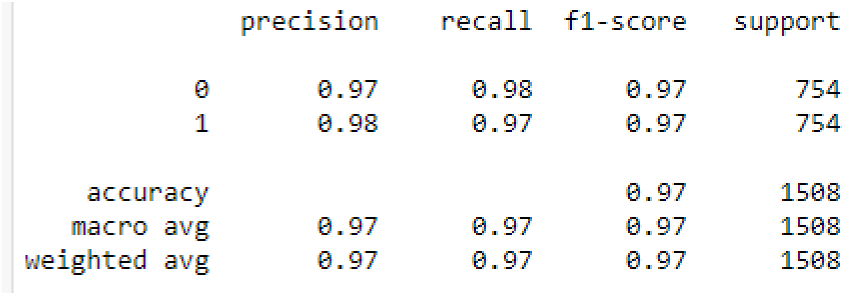
ROC Curve and AUC Score – Test

```
AUC Score for Bagging RF GridSearchCV test data: 0.881
```



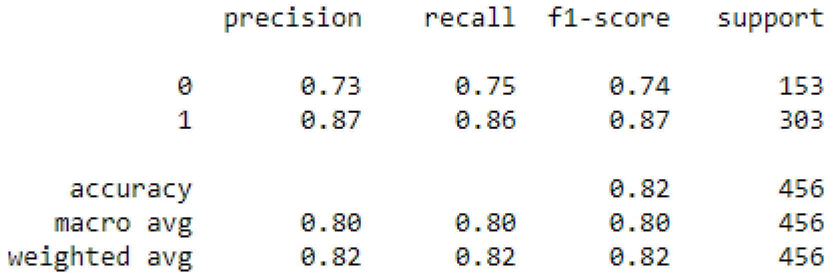
Model 15- SMOTE Bagging RF Model

Classification Report for Train Data



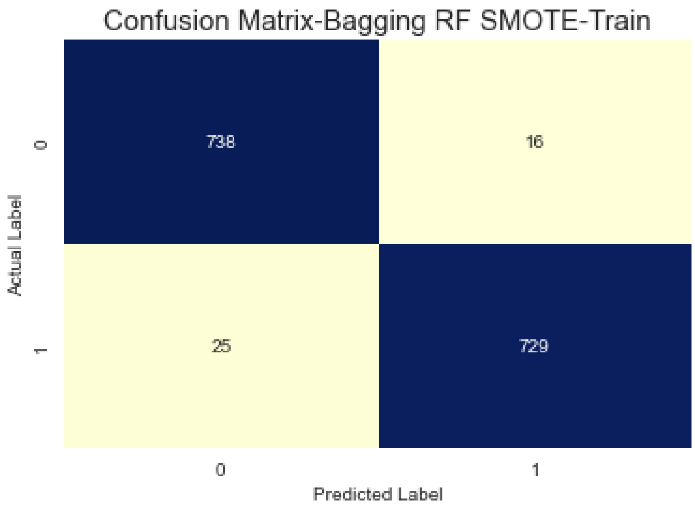
```
              precision    recall  f1-score   support

           0       0.97      0.98      0.97       754
           1       0.98      0.97      0.97       754

    accuracy                           0.97      1508
   macro avg       0.97      0.97      0.97      1508
weighted avg       0.97      0.97      0.97      1508
```

Classification Report for Test Data

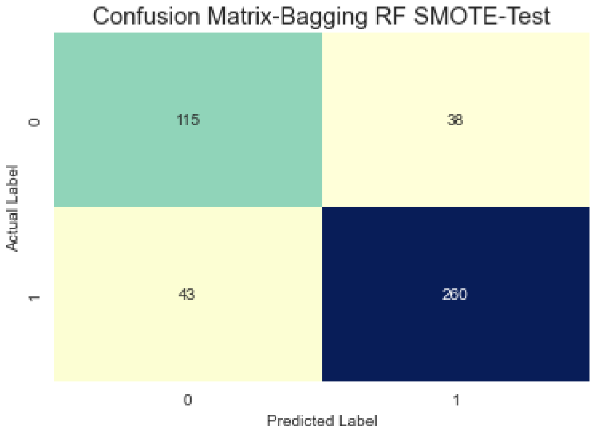


```
              precision    recall  f1-score   support

           0       0.73      0.75      0.74       153
           1       0.87      0.86      0.87       303

    accuracy                           0.82       456
   macro avg       0.80      0.80      0.80       456
weighted avg       0.82      0.82      0.82       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

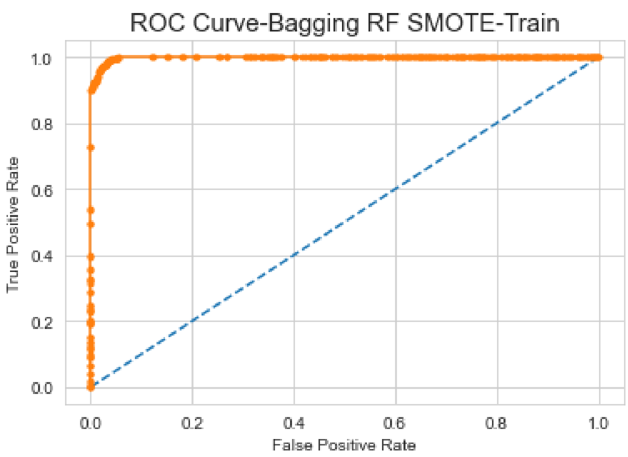


Accuracy Score- Train and Test

```
Accuracy score of Bagging RF SMOTE Trained data is  0.97
Accuracy score of Bagging RF SMOTE Tested data is  0.82
```

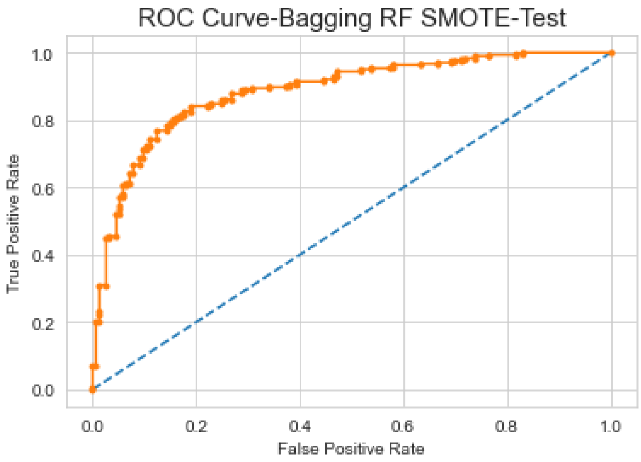
ROC Curve and AUC Score – Train

```
AUC Score for Bagging RF SMOTE train data: 0.998
```



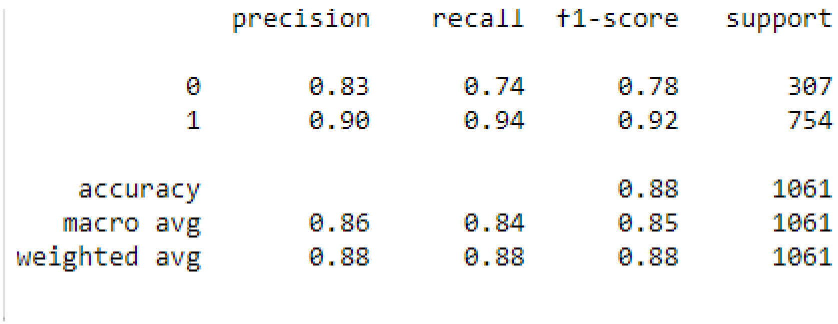
ROC Curve and AUC Score – Test

```
AUC Score for Bagging RF SMOTE test data: 0.886
```



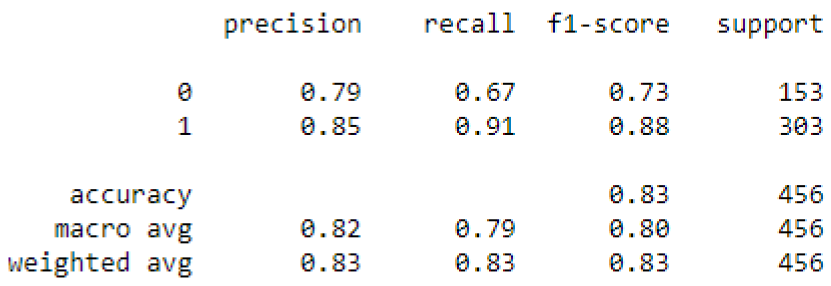
Model 16 - Basic Gradient Boosting

Classification Report for Train Data



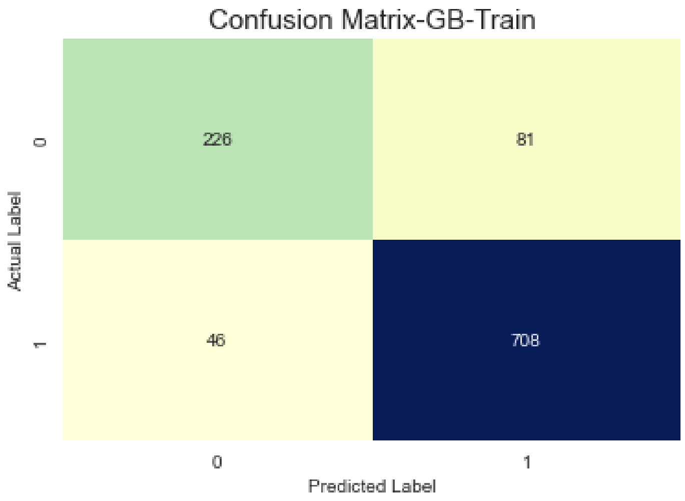
```
              precision    recall  f1-score   support

           0       0.83      0.74      0.78       307
           1       0.90      0.94      0.92       754

    accuracy                           0.88      1061
   macro avg       0.86      0.84      0.85      1061
weighted avg       0.88      0.88      0.88      1061
```

Classification Report for Test Data

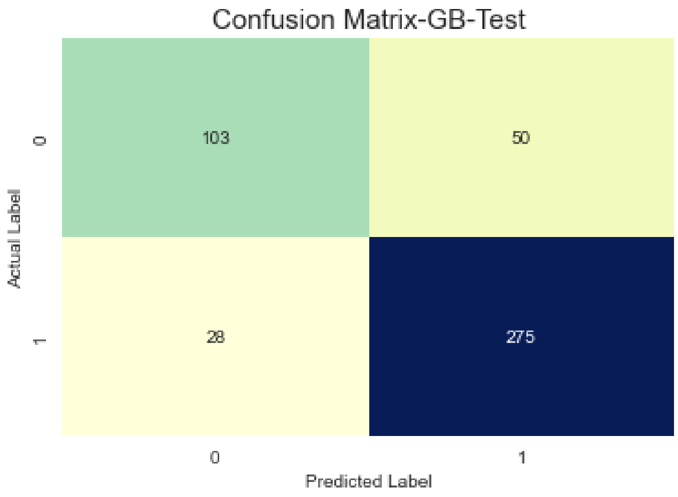


```
              precision    recall  f1-score   support

           0       0.79      0.67      0.73       153
           1       0.85      0.91      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.79      0.80       456
weighted avg       0.83      0.83      0.83       456
```

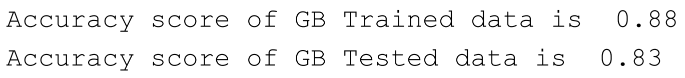
Confusion Matrix - Train Data



Confusion Matrix - Test Data



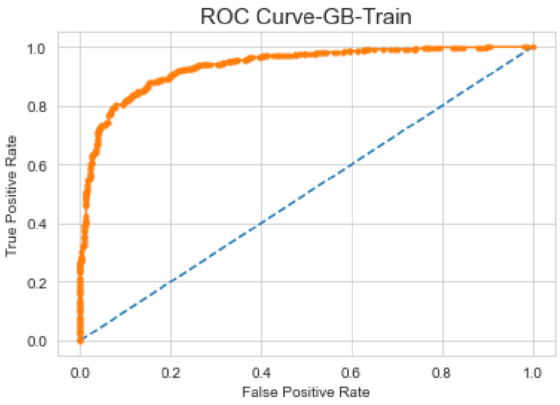
Accuracy Score- Train and Test



```
Accuracy score of GB Trained data is  0.88
Accuracy score of GB Tested data is  0.83
```

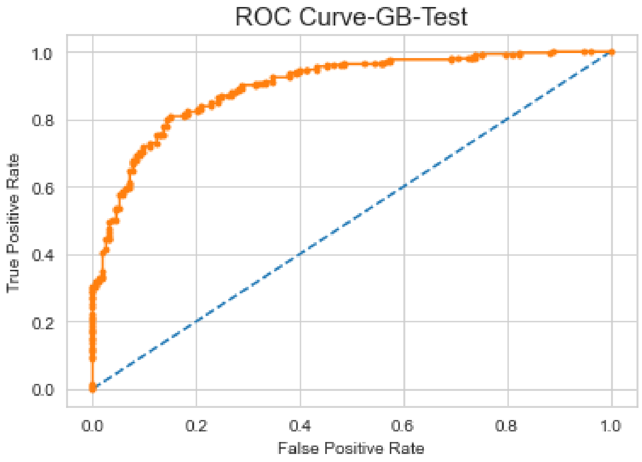
ROC Curve and AUC Score – Train

```
AUC Score for GB train data: 0.935
```



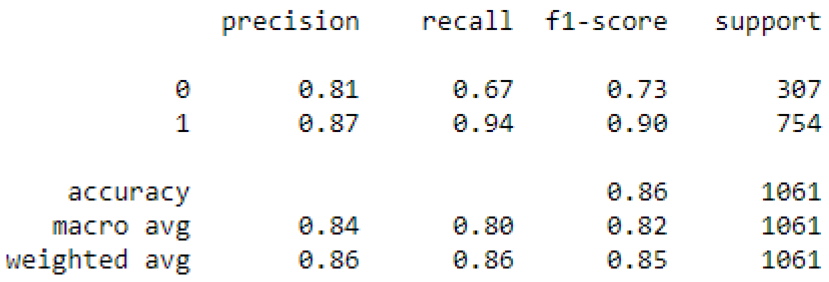
ROC Curve and AUC Score – Test

```
AUC Score for GB test data: 0.897
```



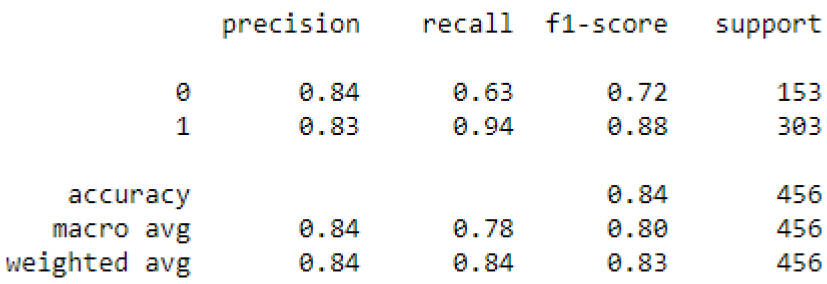
Model 17- GridSeacrhCV Gradient Boosting Model

Classification Report for Train Data



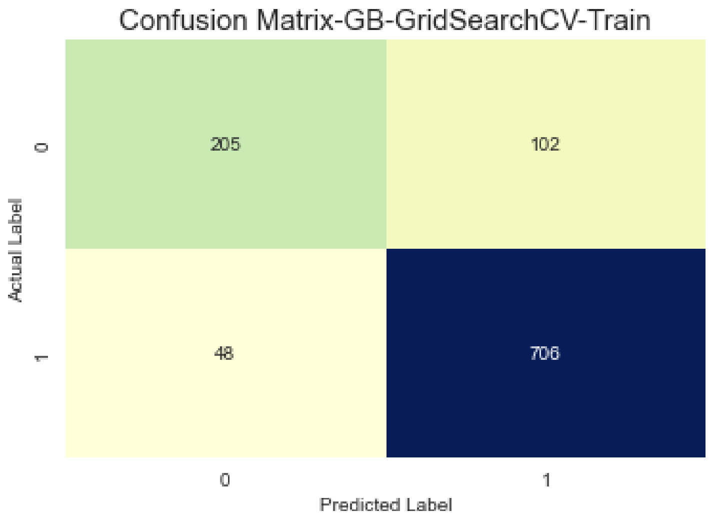
```
              precision    recall  f1-score   support

           0       0.81      0.67      0.73       307
           1       0.87      0.94      0.90       754

    accuracy                           0.86      1061
   macro avg       0.84      0.80      0.82      1061
weighted avg       0.86      0.86      0.85      1061
```

Classification Report for Test Data

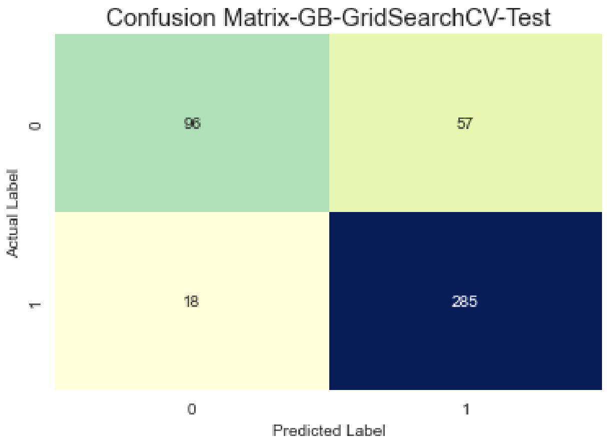


```
              precision    recall  f1-score   support

           0       0.84      0.63      0.72       153
           1       0.83      0.94      0.88       303

    accuracy                           0.84       456
   macro avg       0.84      0.78      0.80       456
weighted avg       0.84      0.84      0.83       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

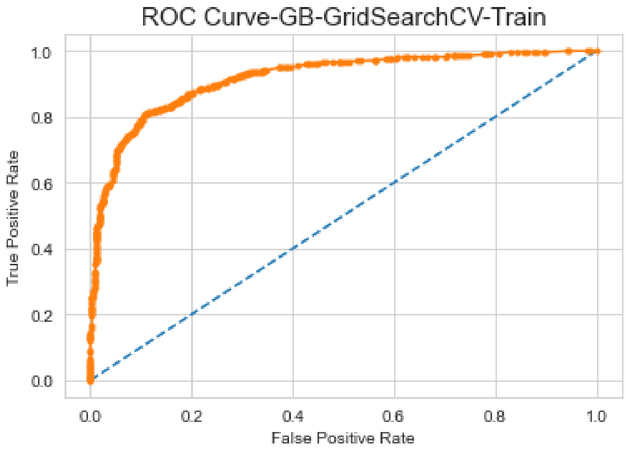


Accuracy Score- Train and Test

```
Accuracy score of GB GridSearchCV Trained data is  0.86
Accuracy score of GB GridSearchCV Tested data is  0.84
```

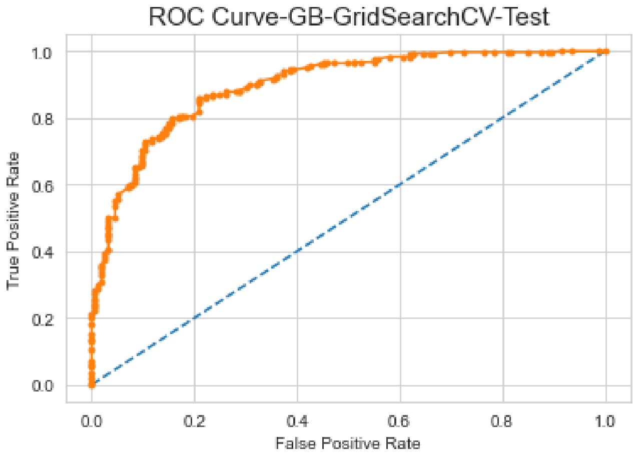
ROC Curve and AUC Score – Train

```
AUC Score for GB GridSearchCV train data: 0.918
```



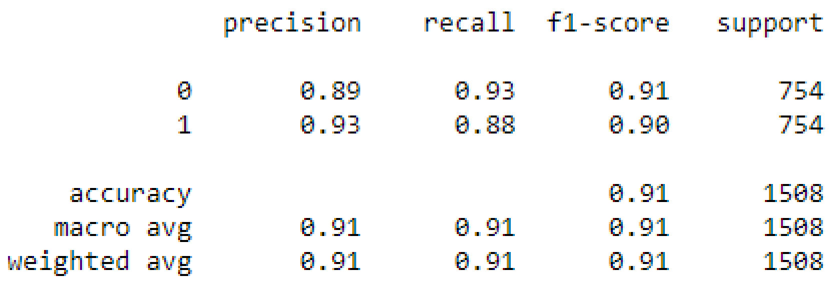
ROC Curve and AUC Score – Test

```
AUC Score for GB GridSearchCV test data: 0.897
```



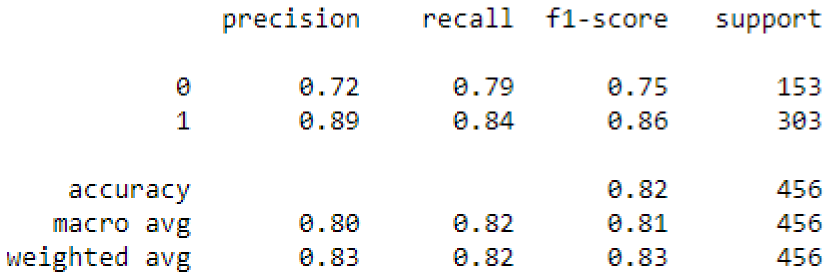
Model 18- SMOTE Gradient Boosting Model

Classification Report for Train Data



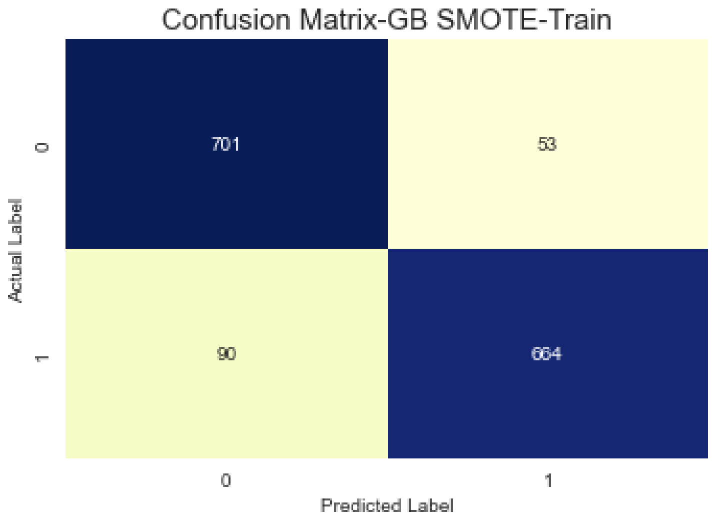
```
              precision    recall  f1-score   support

           0       0.89      0.93      0.91       754
           1       0.93      0.88      0.90       754

    accuracy                           0.91      1508
   macro avg       0.91      0.91      0.91      1508
weighted avg       0.91      0.91      0.91      1508
```

Classification Report for Test Data

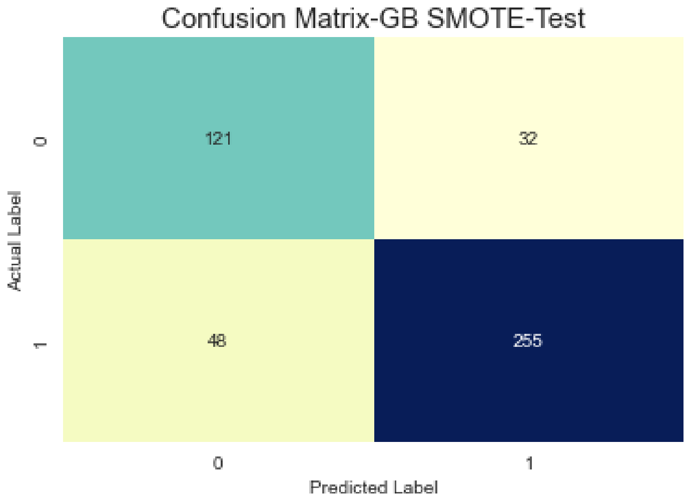


```
              precision    recall  f1-score   support

           0       0.72      0.79      0.75       153
           1       0.89      0.84      0.86       303

    accuracy                           0.82       456
   macro avg       0.80      0.82      0.81       456
weighted avg       0.83      0.82      0.83       456
```

Confusion Matrix - Train Data



Confusion Matrix - Test Data

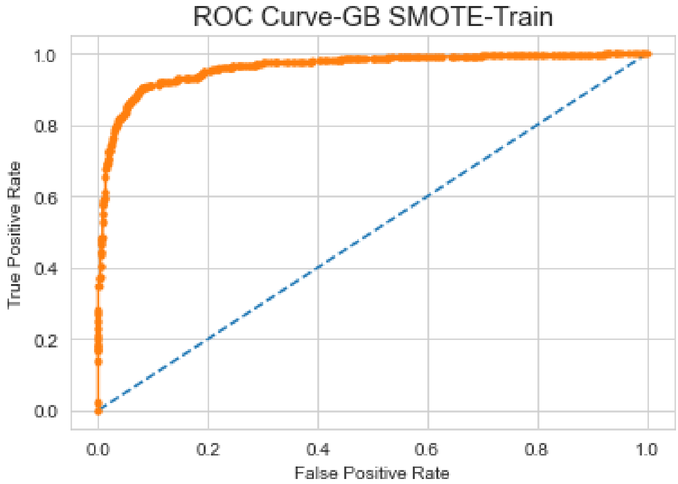


Accuracy Score- Train and Test

```
Accuracy score of GB SMOTE Trained data is  0.91
Accuracy score of GB SMOTE Tested data is  0.82
```

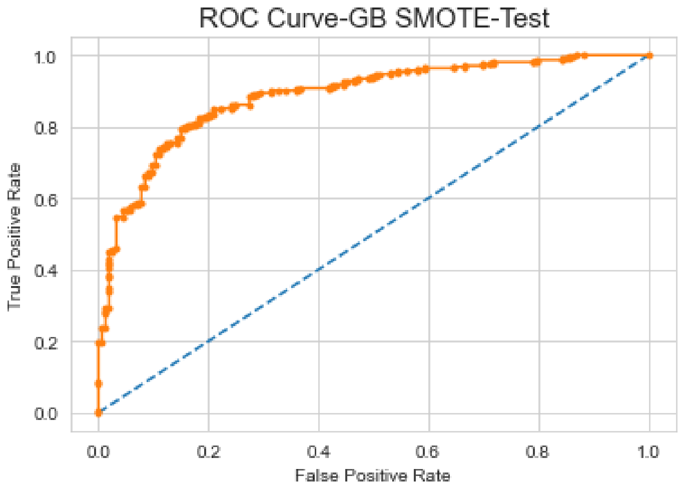
ROC Curve and AUC Score – Train

```
AUC Score for GB SMOTE train data: 0.962
```

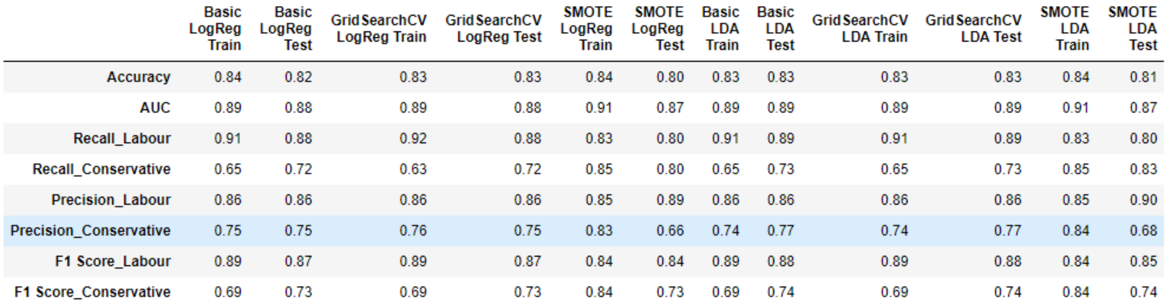


ROC Curve and AUC Score – Test

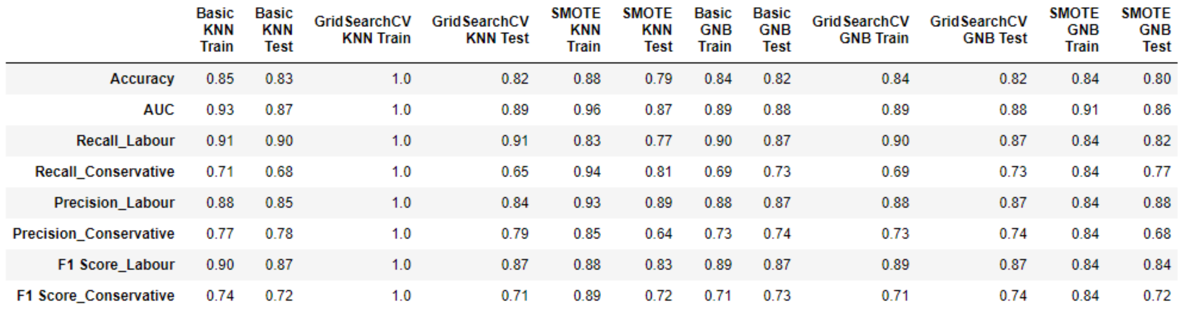
```
AUC Score for GB SMOTE test data: 0.886
```



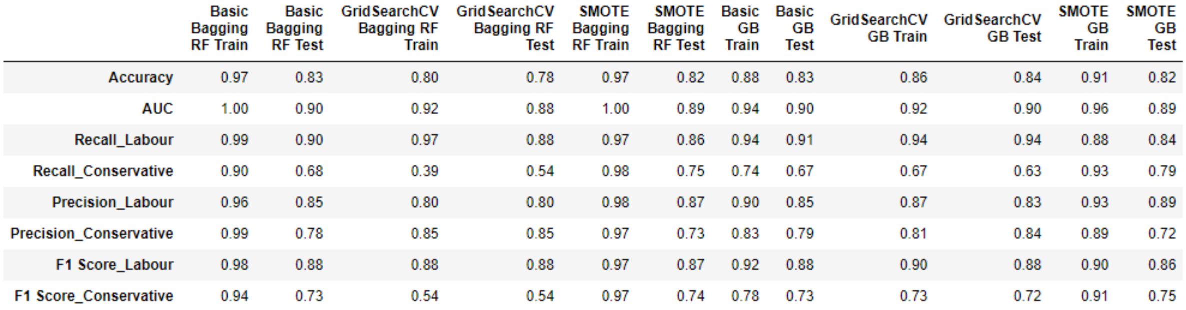
Comparison of all models



	Basic LogReg Train	Basic LogReg Test	GridSearchCV LogReg Train	GridSearchCV LogReg Test	SMOTE LogReg Train	SMOTE LogReg Test	Basic LDA Train	Basic LDA Test	GridSearchCV LDA Train	GridSearchCV LDA Test	SMOTE LDA Train	SMOTE LDA Test
Accuracy	0.84	0.82	0.83	0.83	0.84	0.80	0.83	0.83	0.83	0.83	0.84	0.81
AUC	0.89	0.88	0.89	0.88	0.91	0.87	0.89	0.89	0.89	0.89	0.91	0.87
Recall_Labour	0.91	0.88	0.92	0.88	0.83	0.80	0.91	0.89	0.91	0.89	0.83	0.80
Recall_Conservative	0.65	0.72	0.63	0.72	0.85	0.80	0.65	0.73	0.65	0.73	0.85	0.83
Precision_Labour	0.86	0.86	0.86	0.86	0.85	0.89	0.86	0.86	0.86	0.86	0.85	0.90
Precision_Conservative	0.75	0.75	0.76	0.75	0.83	0.66	0.74	0.77	0.74	0.77	0.84	0.68
F1 Score_Labour	0.89	0.87	0.89	0.87	0.84	0.84	0.89	0.88	0.89	0.88	0.84	0.85
F1 Score_Conservative	0.69	0.73	0.69	0.73	0.84	0.73	0.69	0.74	0.69	0.74	0.84	0.74



	Basic KNN Train	Basic KNN Test	GridSearchCV KNN Train	GridSearchCV KNN Test	SMOTE KNN Train	SMOTE KNN Test	Basic GNB Train	Basic GNB Test	GridSearchCV GNB Train	GridSearchCV GNB Test	SMOTE GNB Train	SMOTE GNB Test
Accuracy	0.85	0.83	1.0	0.82	0.88	0.79	0.84	0.82	0.84	0.82	0.84	0.80
AUC	0.93	0.87	1.0	0.89	0.96	0.87	0.89	0.88	0.89	0.88	0.91	0.86
Recall_Labour	0.91	0.90	1.0	0.91	0.83	0.77	0.90	0.87	0.90	0.87	0.84	0.82
Recall_Conservative	0.71	0.68	1.0	0.65	0.94	0.81	0.69	0.73	0.69	0.73	0.84	0.77
Precision_Labour	0.88	0.85	1.0	0.84	0.93	0.89	0.88	0.87	0.88	0.87	0.84	0.88
Precision_Conservative	0.77	0.78	1.0	0.79	0.85	0.64	0.73	0.74	0.73	0.74	0.84	0.68
F1 Score_Labour	0.90	0.87	1.0	0.87	0.88	0.83	0.89	0.87	0.89	0.87	0.84	0.84
F1 Score_Conservative	0.74	0.72	1.0	0.71	0.89	0.72	0.71	0.73	0.71	0.74	0.84	0.72



	Basic Bagging RF Train	Basic Bagging RF Test	GridSearchCV Bagging RF Train	GridSearchCV Bagging RF Test	SMOTE Bagging RF Train	SMOTE Bagging RF Test	Basic GB Train	Basic GB Test	GridSearchCV GB Train	GridSearchCV GB Test	SMOTE GB Train	SMOTE GB Test
Accuracy	0.97	0.83	0.80	0.78	0.97	0.82	0.88	0.83	0.86	0.84	0.91	0.82
AUC	1.00	0.90	0.92	0.88	1.00	0.89	0.94	0.90	0.92	0.90	0.96	0.89
Recall_Labour	0.99	0.90	0.97	0.88	0.97	0.86	0.94	0.91	0.94	0.94	0.88	0.84
Recall_Conservative	0.90	0.68	0.39	0.54	0.98	0.75	0.74	0.67	0.67	0.63	0.93	0.79
Precision_Labour	0.96	0.85	0.80	0.80	0.98	0.87	0.90	0.85	0.87	0.83	0.93	0.89
Precision_Conservative	0.99	0.78	0.85	0.85	0.97	0.73	0.83	0.79	0.81	0.84	0.89	0.72
F1 Score_Labour	0.98	0.88	0.88	0.88	0.97	0.87	0.92	0.88	0.90	0.88	0.90	0.86
F1 Score_Conservative	0.94	0.73	0.54	0.54	0.97	0.74	0.78	0.73	0.73	0.72	0.91	0.75

The best model among basic and Gridsearchcv models are found on the basis of F1 score as they are done based on unbalanced data. The best one is GridsearchCV LDA.

The best model among SMOTE is determined on the basis of accuracy as they are done based on balanced data. The best one is SMOTE LDA.

Hence LDA is the best model.

1.8 Based on these predictions, what are the insights?

Business Insights:

-> Most of the voters assessed the ratings of economic conditions and household conditions as good favouring to ‘Labour’ Party.

-> Most people having Eurosceptic sentiment favour Conservative property.

-> People older than 55 mostly favour Conservative party.

-> The number of females who has given a high score to national economic conditions is less than males.

-> More females are Eurosceptic than males.

-> The number of females who has given a low score to household economic conditions is higher than males.

Recommendations:

->Based on the sentiments of people like Eurosceptic sentiment the election campaign of a political party must be planned.

-> More data like whether the leader has faced any scandals in the past,manifesto of the parties can be collected to build better models.

-> Within a district voters belonging to different communites,groups can have reasons to vote for a particular party which can be studied.It plays a major role in vote swing(people would vote for a different party in the next election).

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:

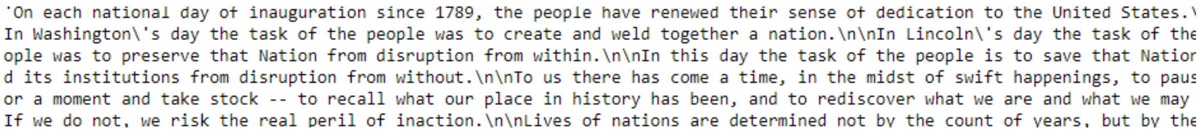
1. President Franklin D. Roosevelt in 1941
2. President John F. Kennedy in 1961
3. President Richard Nixon in 1973.

Introduction

The purpose of this whole exercise is to apply pre-processing techniques on the speeches of the 3 Presidents individually to prepare the data for further text analysis. Text analysis is done to find the most common words in the 3 speeches and also plot the Word Cloud.

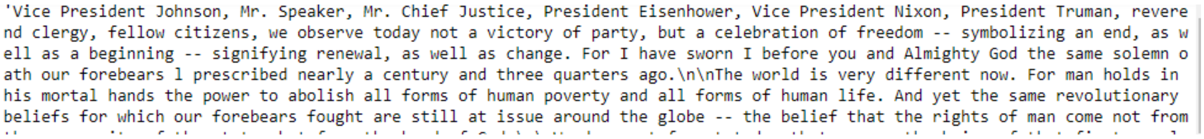
Data Description

1941-Roosevelt's Speech Sample



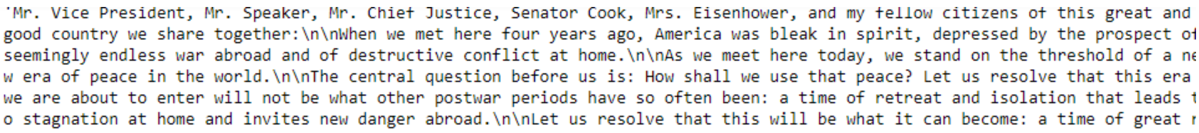
```
'On each national day of inauguration since 1789, the people have renewed their sense of dedication to the United States.\
In Washington\'s day the task of the people was to create and weld together a nation.\n\nIn Lincoln\'s day the task of the
ople was to preserve that Nation from disruption from within.\n\nIn this day the task of the people is to save that Natior
d its institutions from disruption from without.\n\nTo us there has come a time, in the midst of swift happenings, to paus
or a moment and take stock -- to recall what our place in history has been, and to rediscover what we are and what we may
If we do not, we risk the real peril of inaction.\n\nLives of nations are determined not by the count of years, but by the
```

1961-Kennedy's Speech Sample



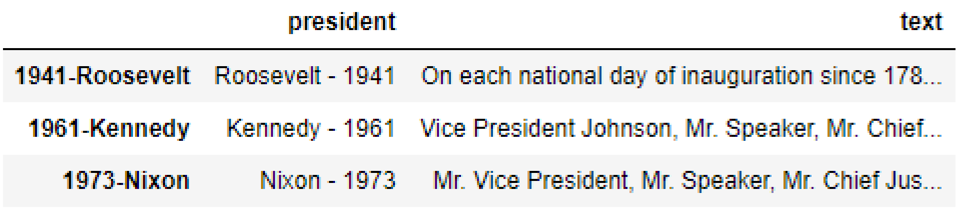
```
'Vice President Johnson, Mr. Speaker, Mr. Chief Justice, President Eisenhower, Vice President Nixon, President Truman, revere
nd clergy, fellow citizens, we observe today not a victory of party, but a celebration of freedom -- symbolizing an end, as w
ell as a beginning -- signifying renewal, as well as change. For I have sworn I before you and Almighty God the same solemn o
ath our forebears l prescribed nearly a century and three quarters ago.\n\nThe world is very different now. For man holds in
his mortal hands the power to abolish all forms of human poverty and all forms of human life. And yet the same revolutionary
beliefs for which our forebears fought are still at issue around the globe -- the belief that the rights of man come not from
```

1973-Nixon's Speech Sample



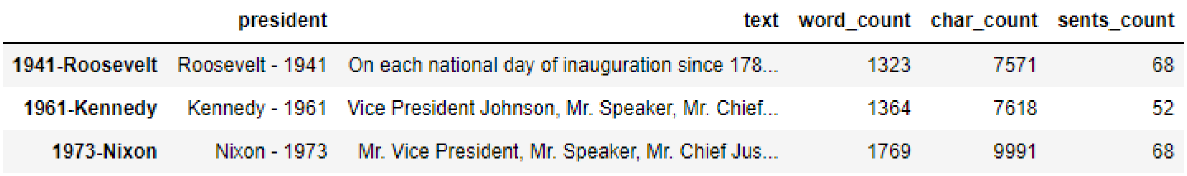
```
'Mr. Vice President, Mr. Speaker, Mr. Chief Justice, Senator Cook, Mrs. Eisenhower, and my fellow citizens of this great and
good country we share together:\n\nWhen we met here four years ago, America was bleak in spirit, depressed by the prospect of
seemingly endless war abroad and of destructive conflict at home.\n\nAs we meet here today, we stand on the threshold of a ne
w era of peace in the world.\n\nThe central question before us is: How shall we use that peace? Let us resolve that this era
we are about to enter will not be what other postwar periods have so often been: a time of retreat and isolation that leads t
o stagnation at home and invites new danger abroad.\n\nLet us resolve that this will be what it can become: a time of great r
```

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



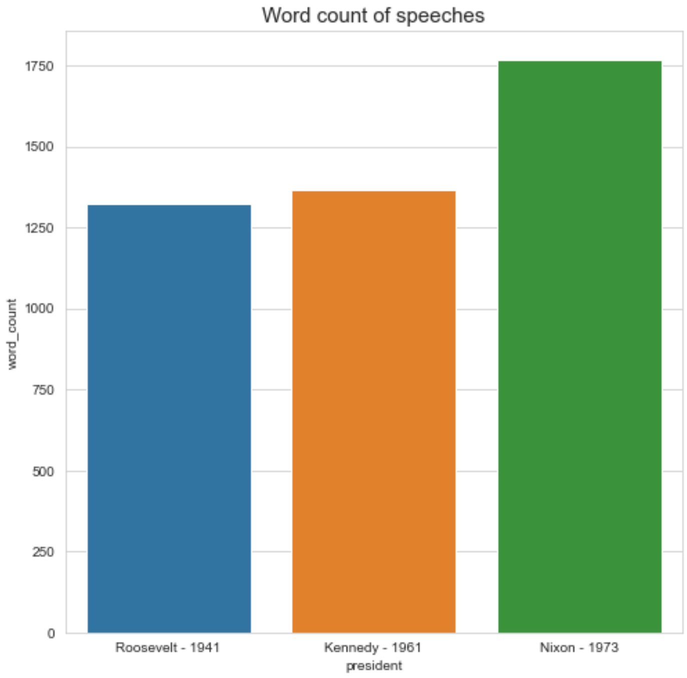
	president	text
1941-Roosevelt	Roosevelt - 1941	On each national day of inauguration since 178...
1961-Kennedy	Kennedy - 1961	Vice President Johnson, Mr. Speaker, Mr. Chief...
1973-Nixon	Nixon - 1973	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...

The speeches are inserted into the dataframe under the column text .The name of the President along with the year is given as column ‘president’ and it is also indexed separately.

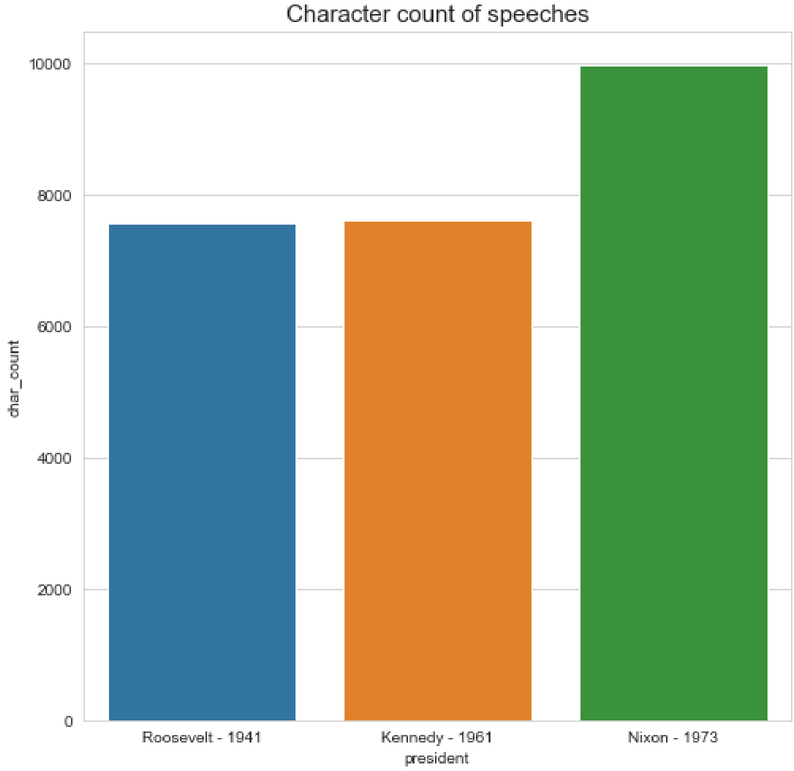


	president	text	word_count	char_count	sents_count
1941-Roosevelt	Roosevelt - 1941	On each national day of inauguration since 178...	1323	7571	68
1961-Kennedy	Kennedy - 1961	Vice President Johnson, Mr. Speaker, Mr. Chief...	1364	7618	52
1973-Nixon	Nixon - 1973	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...	1769	9991	68

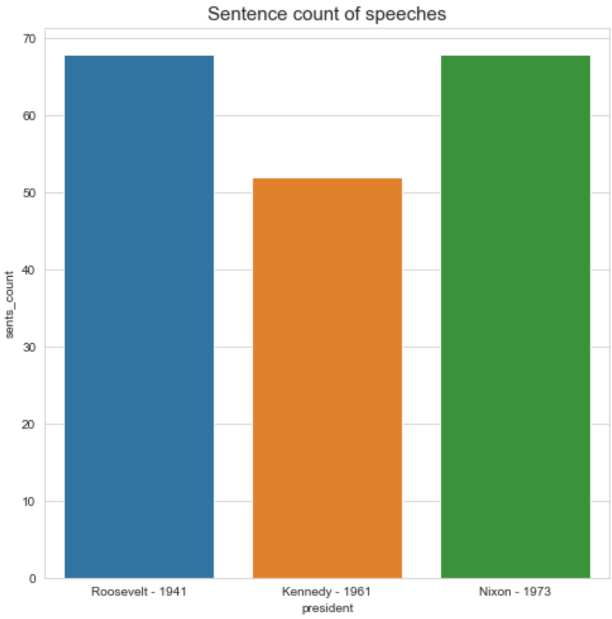
Word Count- Split function splits the word separately (default separator is any whitespace) and using length function the words are counted. Roosevelt’s speech has 1323 words. Kennedy’s speech has 1364 words. Nixon’s speech has 1769 words.



Character Count – It is counted by converting the speech into string and applying length function on it. Roosevelt's speech has 7571 characters. Kennedy's speech has 7618 characters. Nixon's speech has 9991 characters.



Sentence Count – It is done by applying sent_tokenize function on the speech to split data into sentence. The length function is applied on output from this function to count the sentences. Roosevelt's speech has 68 sentences. Kennedy's speech has 52 sentences. Nixon's speech has 68 sentences.

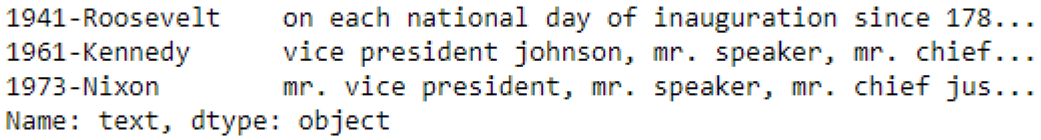


2.2 Remove all the stopwords from all three speeches.

Word count before removing stopwords are given as below:
1941-Roosevelt → 1323
1961-Kennedy → 1364
1973-Nixon → 1769

The following pre-processing techniques are applied on speech data:

Converting into Lowercase
The lower function is applied on the data to give the following output:



```
1941-Roosevelt    on each national day of inauguration since 178...
1961-Kennedy      vice president johnson, mr. speaker, mr. chief...
1973-Nixon        mr. vice president, mr. speaker, mr. chief jus...
Name: text, dtype: object
```

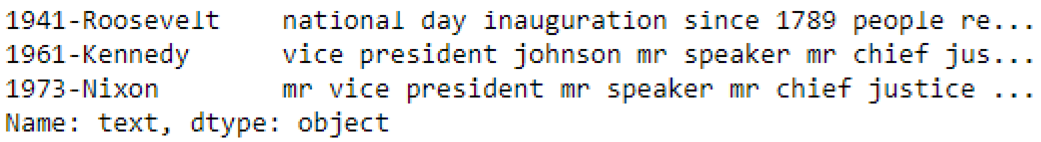
Removing Stopwords and Punctuation
The stopwords module is imported from nltk.corpus package from which English stopwords are extracted using words function. The stopwords along with punctuation marks are stored in the list variable 'stopwords'. The words not present in the list is stored back in the text column.

Word count after removing stopwords are given as below:
1941-Roosevelt → 615
1961-Kennedy → 668
1973-Nixon → 786

Some of the words in Roosevelt's speech are 'each', 'of', 'the', 'have', 'their'.
Some of the words in Kennedy speech are 'we','not','a','of','but'.
Some of the words in Nixon speech are 'and','my','of','this', 'we'.

Removal of special characters

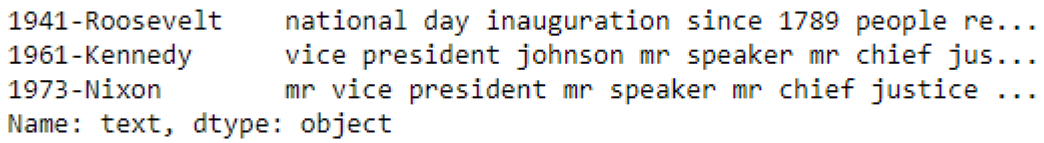
\s is the regular expression for whitespcaes, \w is the regular expression for wo rd characters, ^ denotes not.
The special characters are removed using the regular expression [^\w\s] and t he output is as follows:



```
1941-Roosevelt    national day inauguration since 1789 people re...
1961-Kennedy      vice president johnson mr speaker mr chief jus...
1973-Nixon        mr vice president mr speaker mr chief justice ...
Name: text, dtype: object
```

Lemmatization

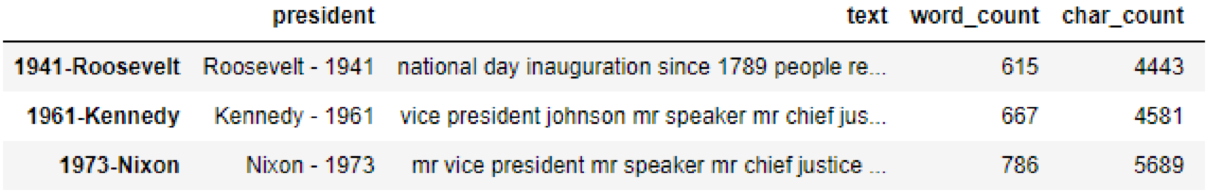
WordNetLemmatizer is used to perform Lemmatization. Lemmatization refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.



```
1941-Roosevelt    national day inauguration since 1789 people re...
1961-Kennedy      vice president johnson mr speaker mr chief jus...
1973-Nixon        mr vice president mr speaker mr chief justice ...
Name: text, dtype: object
```

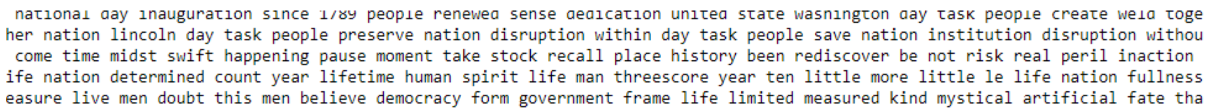
Count of words, characters after preprocessing

Roosevelt's speech has 615 words. Kennedy's speech has 667 words. Nixon's speech has 786 words. Roosevelt's speech has 4443 characters. Kennedy's speech has 4581 characters. Nixon's speech has 5689 characters.



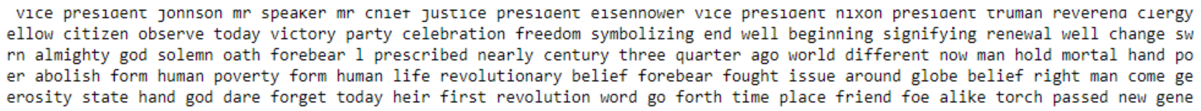
	president	text	word_count	char_count
1941-Roosevelt	Roosevelt - 1941	national day inauguration since 1789 people re...	615	4443
1961-Kennedy	Kennedy - 1961	vice president johnson mr speaker mr chief jus...	667	4581
1973-Nixon	Nixon - 1973	mr vice president mr speaker mr chief justice ...	786	5689

Roosevelt's speech sample after pre-processing:



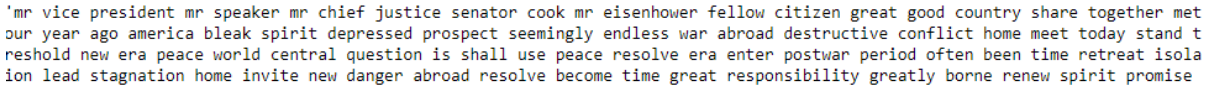
```
 national day inauguration since 1789 people renewed sense dedication united state washington day task people create weld toge
her nation lincoln day task people preserve nation disruption within day task people save nation institution disruption withou
 come time midst swift happening pause moment take stock recall place history been rediscover be not risk real peril inaction
ife nation determined count year lifetime human spirit life man threescore year ten little more little le life nation fullness
easure live men doubt this men believe democracy form government frame life limited measured kind mystical artificial fate tha
```

Kennedy's speech sample after pre-processing:



```
 vice president johnson mr speaker mr chief justice president eisenhower vice president nixon president truman reverend clergy
ellow citizen observe today victory party celebration freedom symbolizing end well beginning signifying renewal well change sw
rn almighty god solemn oath forebear l prescribed nearly century three quarter ago world different now man hold mortal hand po
er abolish form human poverty form human life revolutionary belief forebear fought issue around globe belief right man come ge
erosity state hand god dare forget today heir first revolution word go forth time place friend foe alike torch passed new gene
```

Nixon's speech sample after pre-processing:



```
'mr vice president mr speaker mr chief justice senator cook mr eisenhower fellow citizen great good country share together met
our year ago america bleak spirit depressed prospect seemingly endless war abroad destructive conflict home meet today stand t
reshold new era peace world central question is shall use peace resolve era enter postwar period often been time retreat isola
ion lead stagnation home invite new danger abroad resolve become time great responsibility greatly borne renew spirit promise
```

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

Top 3 words from Roosevelt's Speech along with their frequencies are:
nation → 15
life → 11
democracy → 9

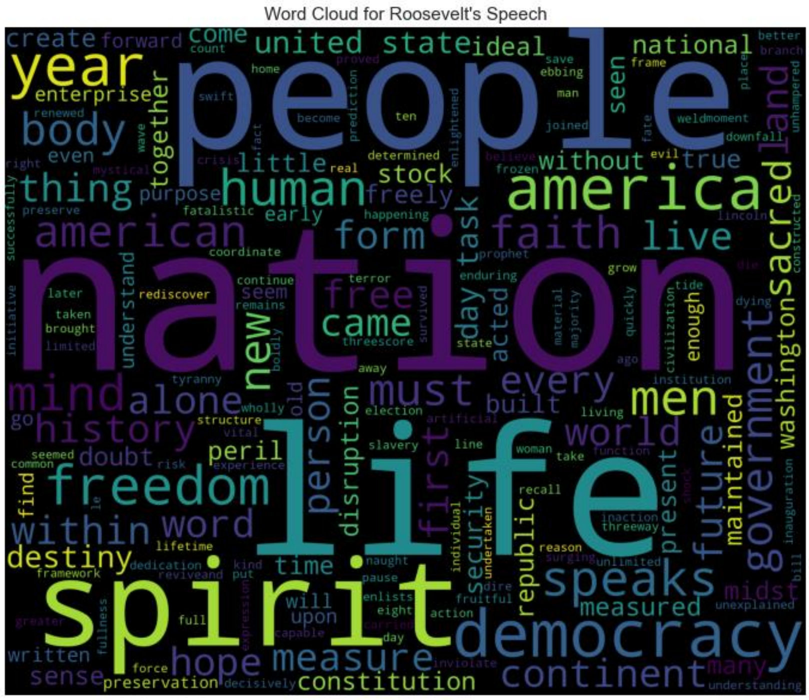
Top 3 words from Kennedy’s Speech along with their frequencies are:
side → 8
world → 8
nation → 7

Top 3 words from Nixon’s Speech along with their frequencies are:
america → 21
peace → 19
world → 18

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

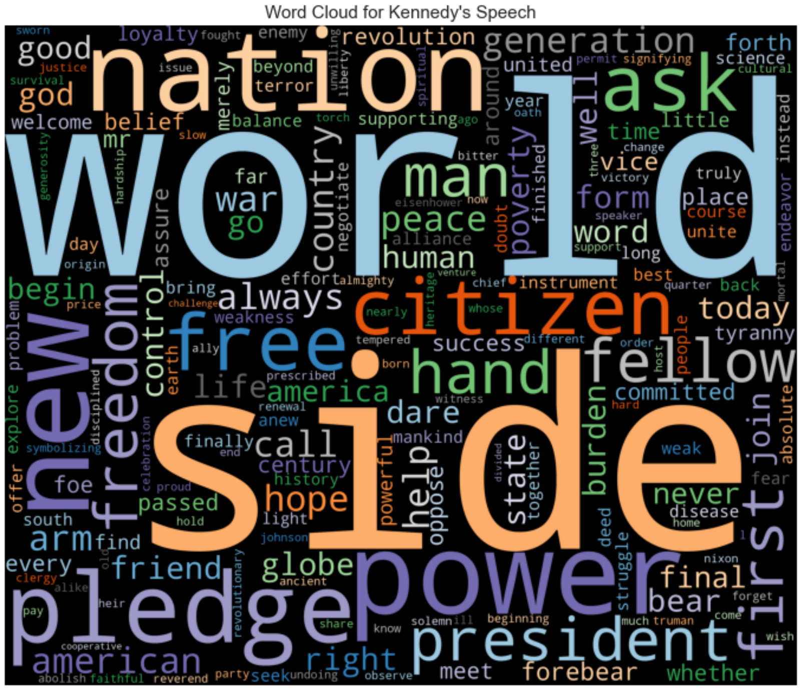
Word clouds are visual representations of word frequency that give greater prominence to words that appear more frequently in a source text. The larger the word in the visual the more common the word was in the documents.

Word Cloud for Roosevelt's Speech



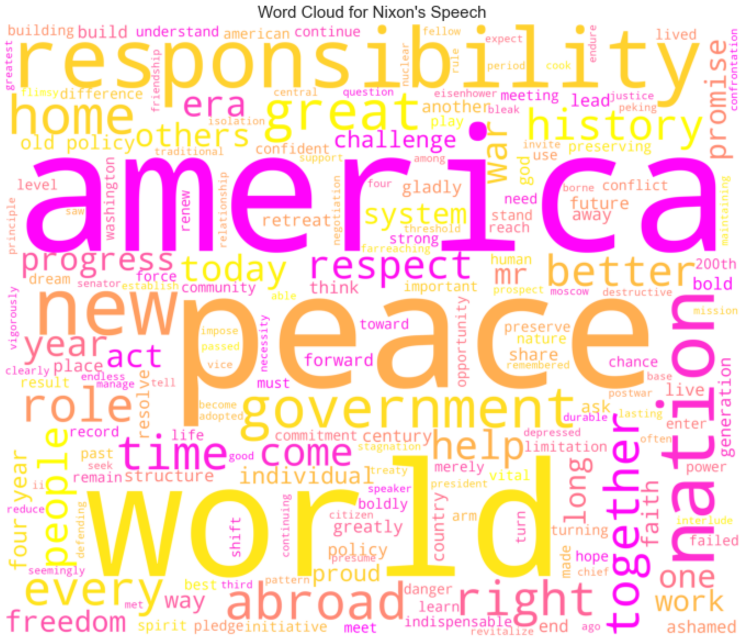
From this we can infer that some of the most common words are nation, life, people, democracy, spirit.

Word Cloud for Kennedy's Speech



From this we can infer that some of the most common words are world, side, nation, power, pledge.

Word Cloud for Nixon's Speech



From this we can infer that some of the most common words are america, peace, world, government, responsibility.

--X-------X-------X--