Aug 2021 | PG-DSBA-Online

PROJECT REPORT
MACHINE LEARNING

SUBMITTED BY
DEV TRIPATHI

Contents

List of Figures

List of Tables

Problem 1

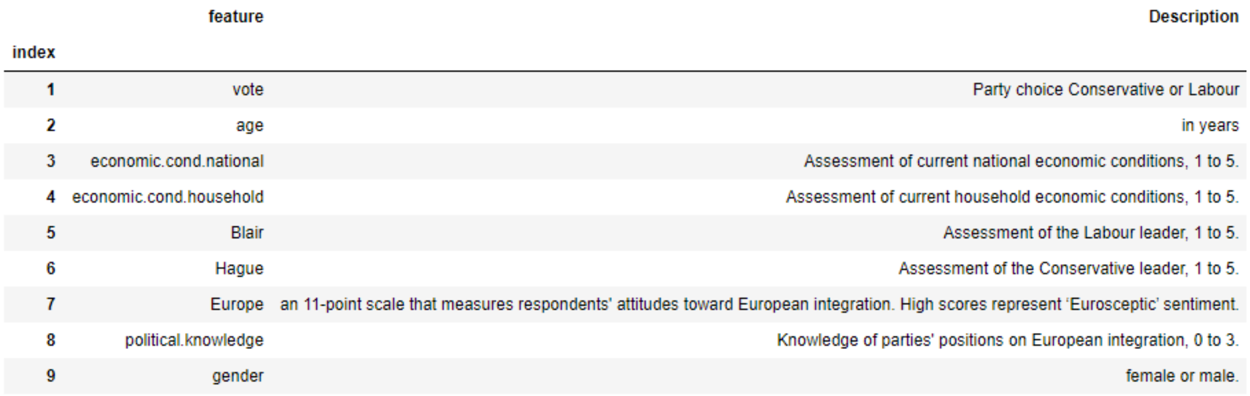
Dataset Analysis

Information about dataset

This dataset is provided by a leading news channel named CNBE, which wants to analyze the recent elections. This survey was conducted on 1525 voters with 9 variables. We have to build a model, to predict which party a voter will vote for based on the given information, to create an exit poll that will help in predicting the overall win and seats covered by a particular party.

Data Dictionary for Dataset

The dataset contains the following features and the information about these features is mentioned below:



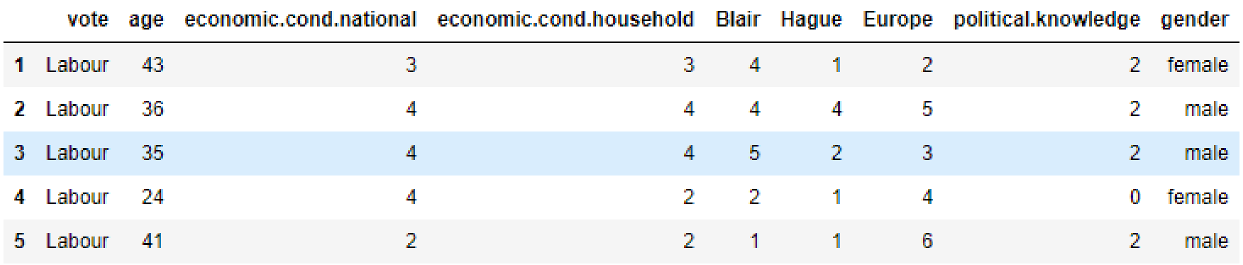
index	feature	Description
1	vote	Party choice Conservative or Labour
2	age	in years
3	economic.cond.national	Assessment of current national economic conditions, 1 to 5.
4	economic.cond.household	Assessment of current household economic conditions, 1 to 5.
5	Blair	Assessment of the Labour leader, 1 to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

Table 1: Data Dictionary for CNBE Dataset

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

EDA

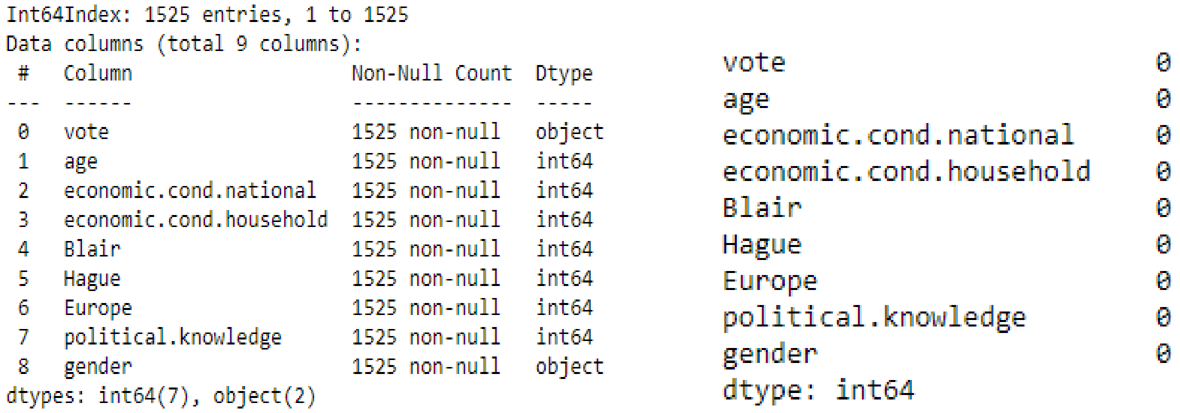
Sample of Dataset-



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

Figure 1: Sample of CNBE Dataset

Variable Information-



```
Int64Index: 1525 entries, 1 to 1525
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)
```

```
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
```

Figure 2: Feature Info and null values count of CNBE Dataset

- The dataset contains a total of 1525 entries and 8 features including our target variable which is **‘vote’**.
- As it is shown in the list, none of the variables contains null values.
- There are two columns i.e., ’vote’ and ‘gender’ which are object data types, which can be further converted to numerical data by simply assigning 0 and 1 for different categories.
- All other variables are of numeric (integer) data type.

1.2) Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also, check for outliers. Interpret the inferences for each.

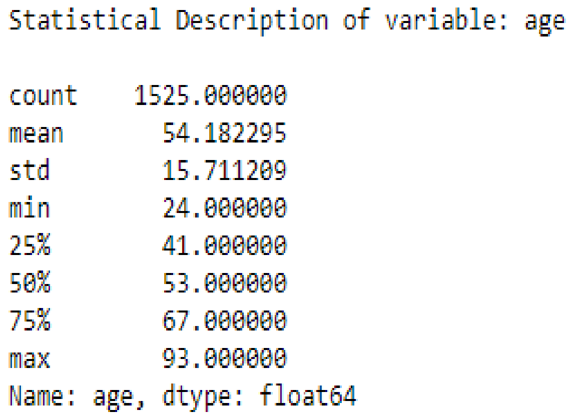
Sol:

As we have already performed null check, data types, and shape earlier we can proceed further with visualization of data.

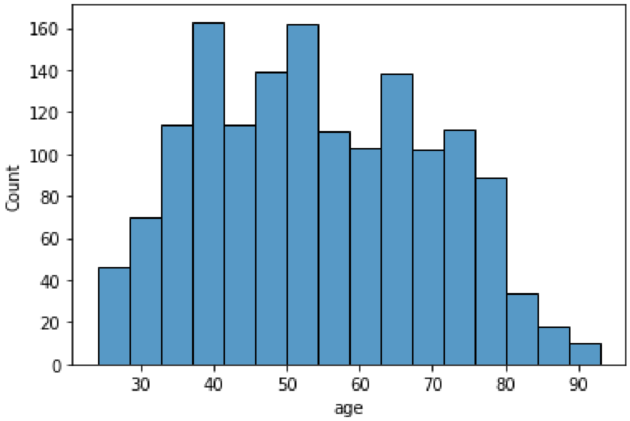
Univariate Analysis:

Numerical Variables: There is only a numeric type variable present in the dataset which is ‘age’.

1. Age:



```
Statistical Description of variable: age

count    1525.000000
mean       54.182295
std        15.711209
min        24.000000
25%        41.000000
50%        53.000000
75%        67.000000
max        93.000000
Name: age, dtype: float64
```



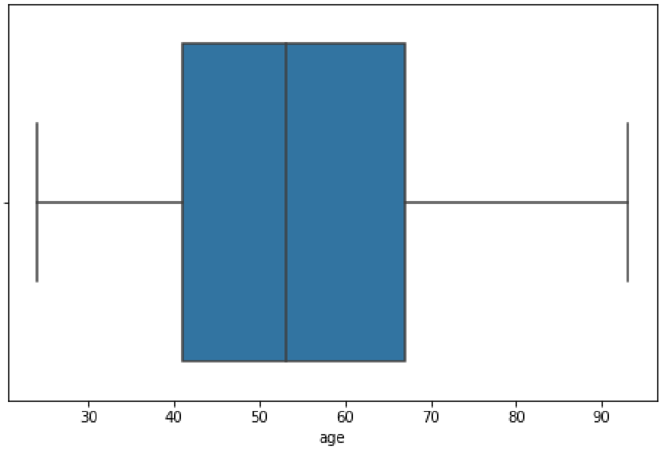
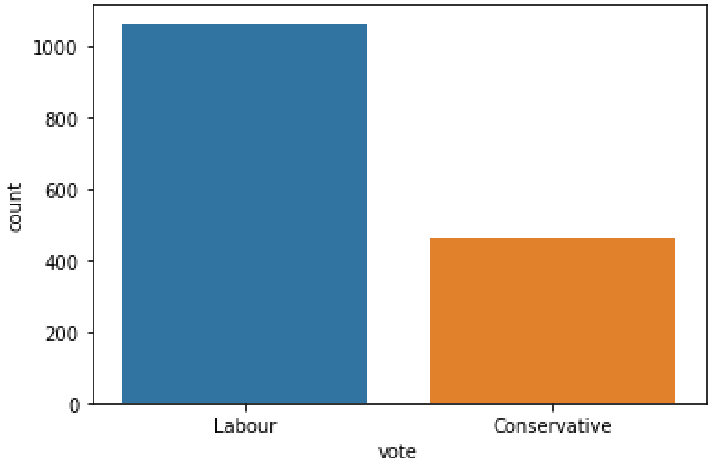


Figure 3: Univariate analysis for 'Age' Variable

As mentioned in the above figure, the mean value is greater than the median, we can say that the distribution is slightly skewed towards the right. Skewness towards the right shows that younger participants are slightly more than older ones. The minimum age of survey participants was 24 and the maximum was 93.

Categorical Variables:

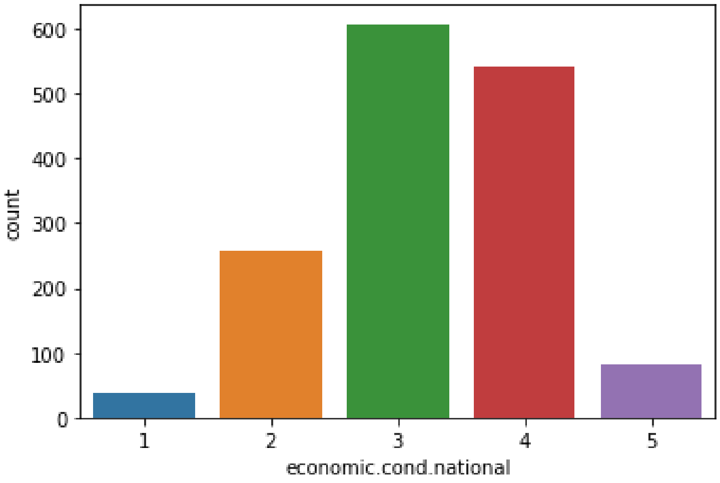
1. **Vote-**



As it is visible in the count plot of the vote variable, the number of people in favor of the 'Labour' party (1063) is more than the number of people in favor of the 'Conservative' party (462).

Figure 4: Count plot for Vote variable

2. Economic Condition National:



As one can see from the figure, most of the people assessed economic conditions to be just fine (for 3 total 607) to good (for 4 total 542). Very few people rated the national economic conditions as 2 (257) or 1 (37).

Figure 5: Count plot for National Economic Condition Assessment

3. <u>Economic Condition of Household</u>:

The number of people who assessed the household conditions to be just fine to very good (as 3,4 or 5) are far more (648,440 and 92 respectively) than people who assessed that to be bad or very bad (as 2 or 1).

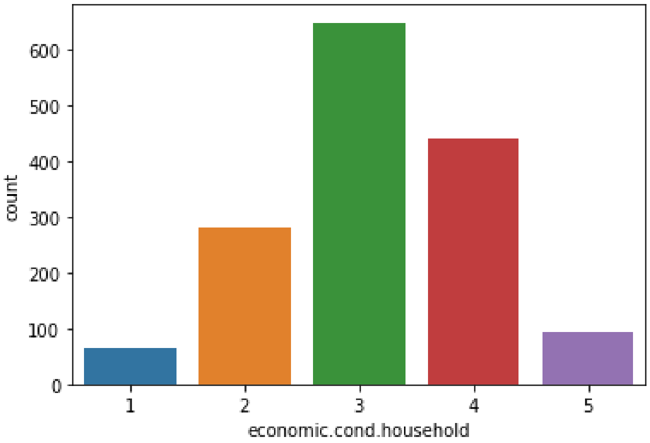


Figure 6: Count Plot for assessment of Economic Condition of Households

4. <u>Blair-</u>

The feature depicts the assessment of the Leader of the Labour party. As we can observe that most of the people (for 4 rating- 836 and 5 rating- 153) have assessed him/her to be on the better side. Though 438 people have rated him as 2 and 97 people as 1. Only 1 person has been rated as 3.

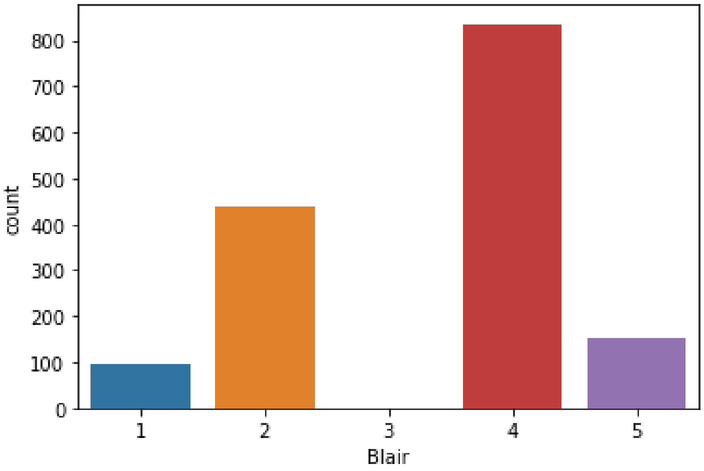


Figure 7: Count plot for assessment of Labour party leader

5. Hague:

This feature shows the assessment of the 'Conservative' party leader. As we can see, the rating is not so good for him/her. The number of people who rated as 2 is 624 and as 1 is 233 The number of people who rated as 4 is 558 and as 5 are 73. Neutral ratings are 37.

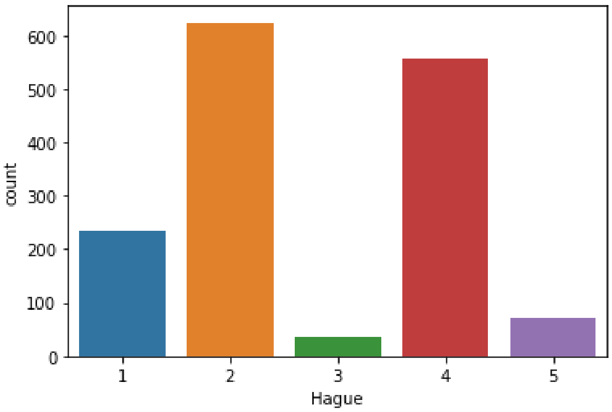
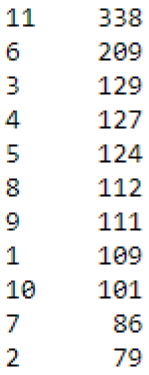


Figure 8: Count plot for assessment of Leader of Conservative party

6. Europe-

This feature represents the assessment of attitude towards European integration. As we can say by just looking at the plot that most of the people are not happy with European integration. Actual values for each rating are following:



```
11    338
6     209
3     129
4     127
5     124
8     112
9     111
1     109
10    101
7      86
2      79
```

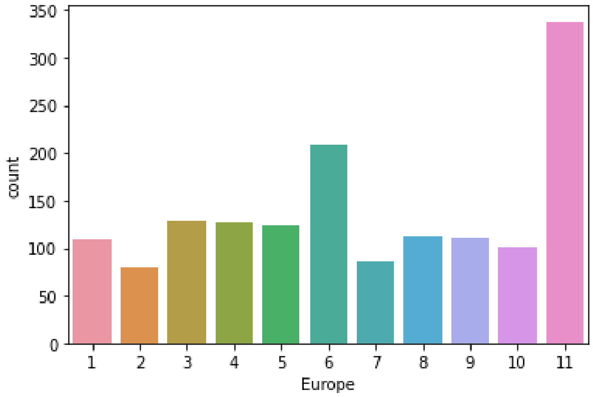


Figure 9: Count plot for Attitude towards European Integration

7. Political Knowledge-

This feature represents the assessment of knowledge of parties' attitudes towards European integration. As we can observe that most of the people have rated it as 2 (782) and 3 (250) which is a good sign for the survey. But few people have rated it to be as 0 (455) and 1 (38).

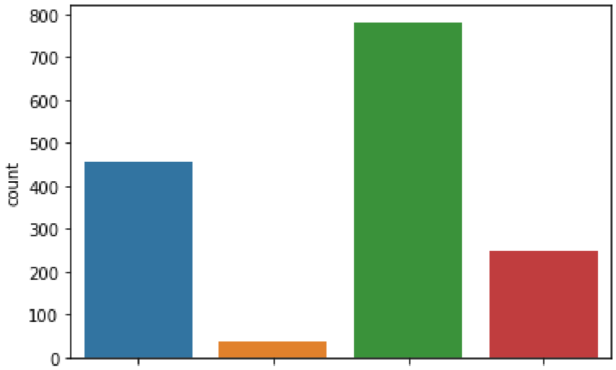
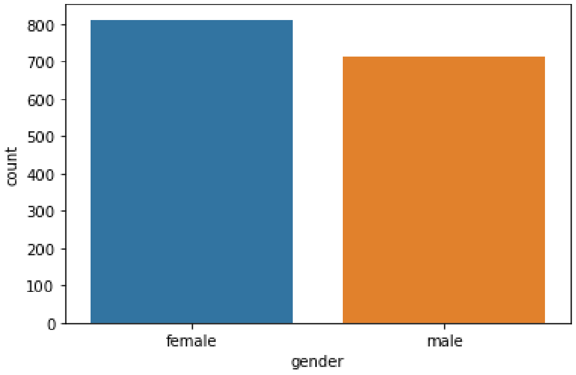


Figure 10: Count plot for political knowledge variable

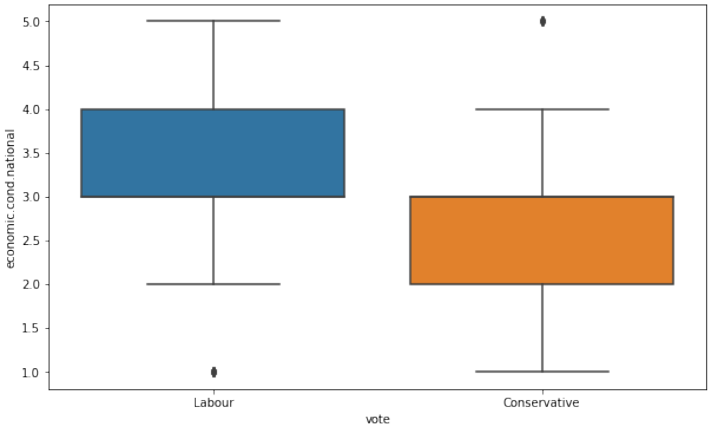


8. Gender-

In the survey, more female candidates (812) have participated than male candidates (713).

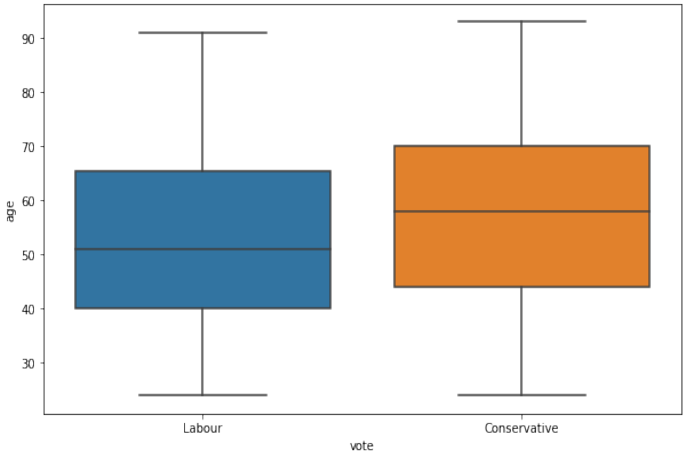
Figure 11: Count plot for gender variable

Bivariate Analysis



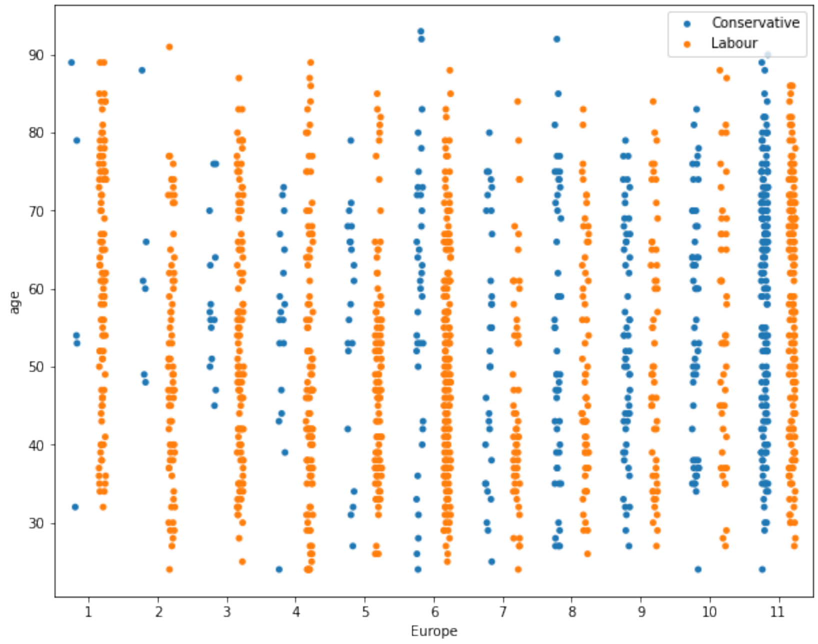
From this boxplot, we can see the people who assessed the economic condition (national) as 3 or higher are in favor of the labor party and the people who have assessed it to be 3 or lesser are in favor of the conservative party.

Figure 12: Boxplot of Economic condition (national) assessment with hue of vote variable



This boxplot shows that the median age of people in favor of the Labour party is slightly less than the median age of people in favor of the Conservative party.

Figure 13: Box plot for Age variable with hue of Vote variable



Now, in this strip plot, if we look at the density of points closely, most people who support the 'Conservative' party from all age groups are not in favor of (with 6 or higher rating) European Integration. On the other hand, most of the people who are in favor of European Integration (6 or lower rating) are supporters of the Labour party.

Figure 14: Strip plot for Age vs Europe variable with hue of Vote variable

Now for obtaining the correlation plot we must encode the object data type variables. Here we have replaced the Conservative category as 0 and Labour as 1 for the **Vote** variable, and Female as 0 and Male as 1 for the **Gender** variable. After that we can obtain the following correlation plot:

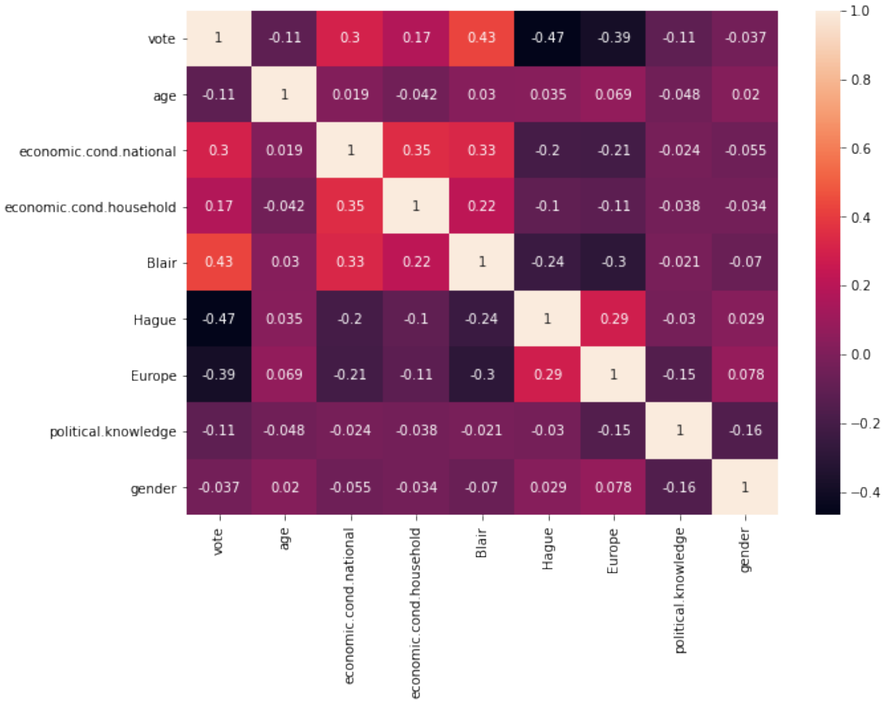
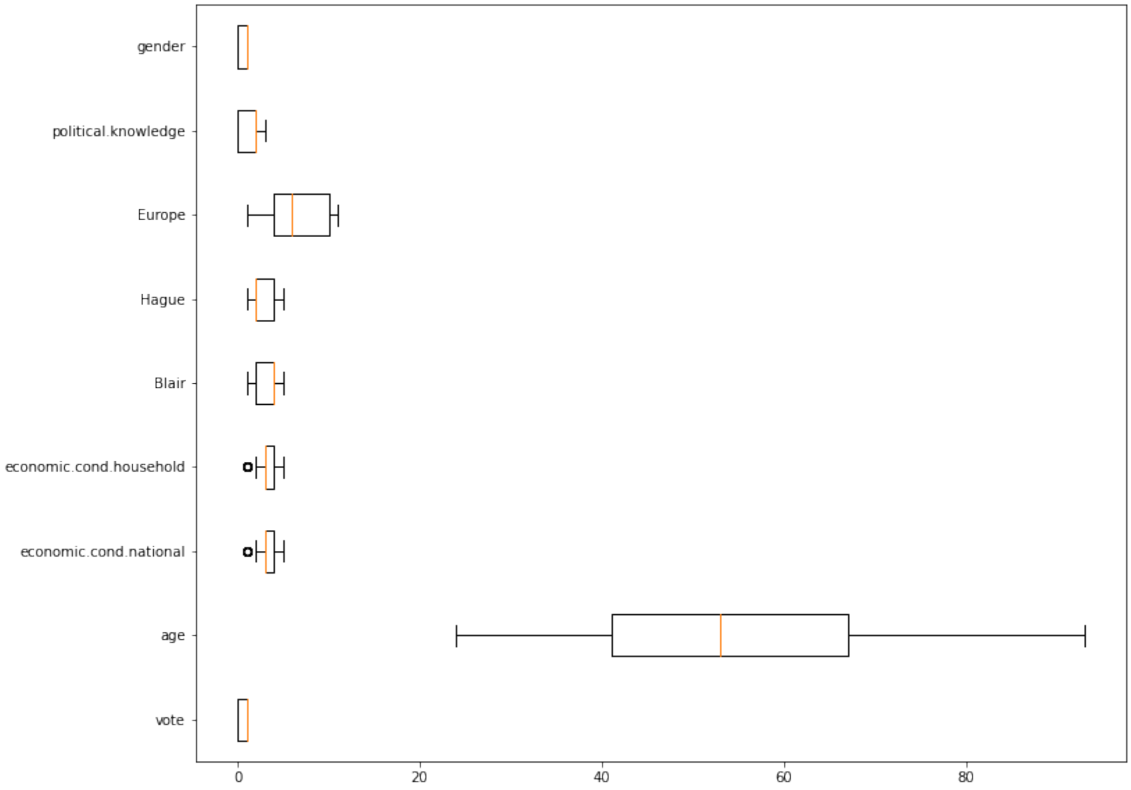


Figure 15: Correlation plot

- The correlation of target variable ‘Vote’ with ‘Hague’ and ‘Europe’ is negative which was expected since we have encoded the ‘Labour’ as 1 and ‘Hague’ is an assessment of leader of ‘Conservative’ party. Also, since, for the Europe variable higher rating means a negative attitude towards European Integration, we can say that voters in support of the ‘Labour’ party are in favor of the same.
- Similarly, a positive correlation with ‘Blair’ was expected as he/she is the leader of the ‘Labour’ party.
- The rest, of the features, have very low to nil correlation with each other (as the value are close to 0)

Now for modeling, we must look at outliers since some of the algorithms we are going to use are sensitive to outliers such as logistic regression, KNN. Hence, for further analysis, we must treat these outliers.

As we can see from the boxplots that only two variables namely (economic.cond.household and economic.cond.national) have outliers rest of the variables are free from outliers.



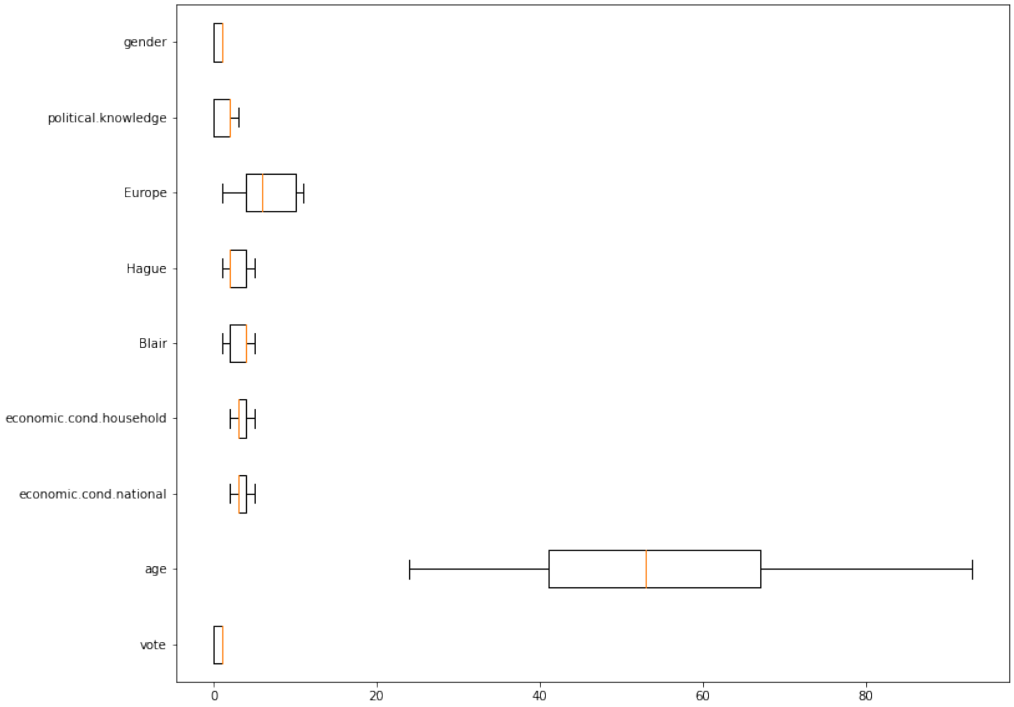


Figure 16: Boxplots before and after treating the outliers

In this analysis, we have simply replaced the outliers with the most frequently occurring value in the series.

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).

Sol:

As mentioned earlier the encoding was done by simply replacing the Conservative category as 0 and Labour as 1 for the **Vote** variable, and Female as 1 and Male as 0 for the **Gender** variable. Since we are using **distance-based methods** like KNN the scaling becomes a necessity here. In this analysis, we have used a Min-Max scaler. In the analysis, we have also scaled the target variable which is 'Vote' because the Min-Max scaler won't affect the values (because only binary values are there). Also, most of the features are ordinal with numerical. Hence the Min-Max scaler can be a better choice for the scaling while retaining the interpretability of the features.

Gender	**Label**
Female	1
Male	0

Vote	**Label**
Conservative	0
Labour	1

After scaling the box plot for all the features looks like:

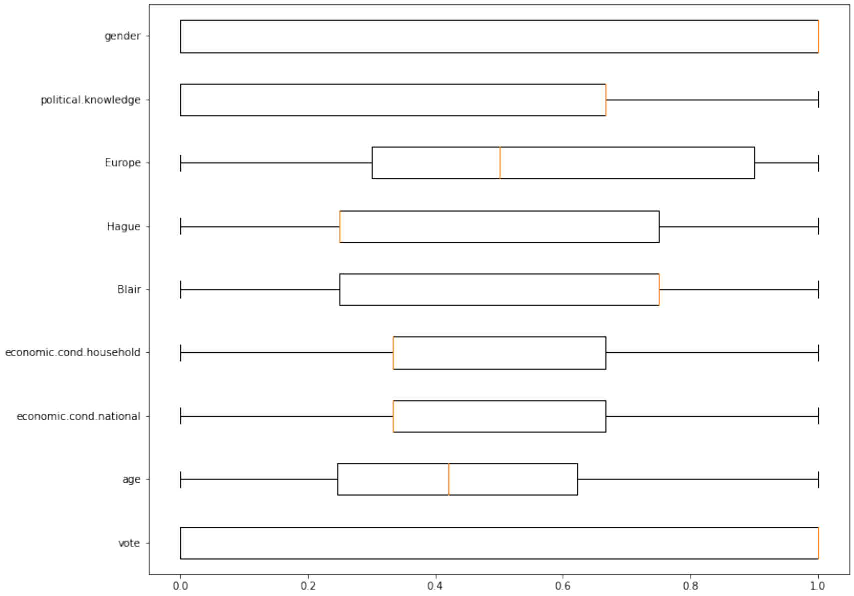
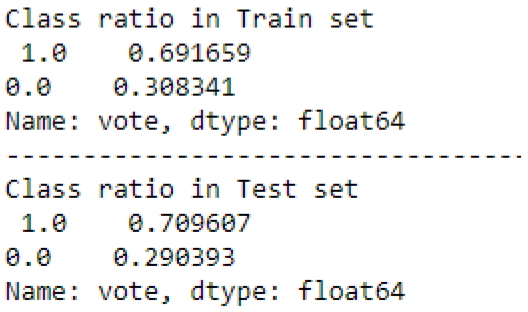


Figure 17: Box plots after scaling

As we can see the scaling worked well as all the features are between 0 and 1.

Now we can split the new data into Train and test sets. Here we have used 70 and 30 ratios (70 percent data for train set and 30 percent data for test set). We have to check the class ratio in each:



```
Class ratio in Train set
 1.0    0.691659
0.0    0.308341
Name: vote, dtype: float64
---------------------------------
Class ratio in Test set
 1.0    0.709607
0.0    0.290393
Name: vote, dtype: float64
```

Figure 18: Class ratio check for Train and Test set

As we can see that the class ratio for the train and test set is fairly equal.

1.4) Apply Logistic Regression and LDA (Linear Discriminant Analysis). Interpret the inferences of both models.

Sol-

Modeling

Logistic Regression Model:

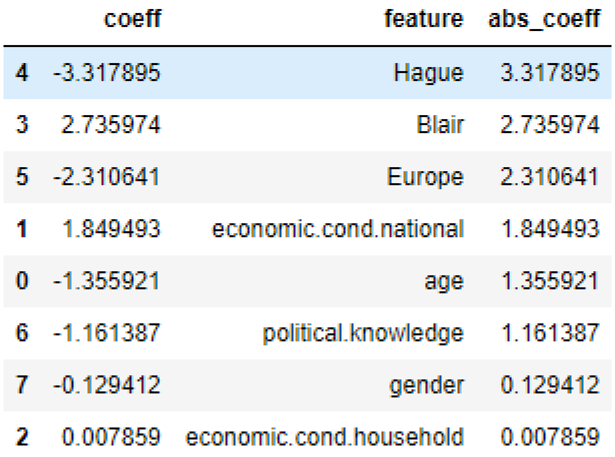
- The aim is to classify the target variable and hence the hyperparameters are chosen to get the desired outcome.
    - Penalty: This hyper-parameter is used to specify the type of normalization used. Few of the values for this hyper-parameter can be l1, l2, or none. The default value is l2.
    - The inverse of regularization: This hyper-parameter is denoted as C. Smaller values of this hyper-parameter indicate a stronger regularization. The default value is 1.0
    - Random state: random_state is the seed used by the random number generator. The default value is None.
    - Solver: This indicates which algorithm to use in the optimization problem. The default value is lbfgs. other possible values are newton-cg, liblinear, sag, saga.
    - Max iter: max_iter represents the maximum number of iterations taken for the solvers to converge a training process.
- Loss function or cross-entropy: We need to make sure our loss function which is nothing but the prediction error has to be the least, we need to find the model that gives the least loss function, this happens when there are fewer misclassifications.

In our analysis, we found that for the default value of maximum iterations the which is 100 the algorithm was unable to converge so we increased the number of iterations to 1000.

Now after training the model with default parameters the accuracy score obtained was 81 percent though the tuning of hyperparameters was tried for different values the results were comparable to default parameters. The parameters obtained after hyperparameter tuning are:

```
{'C': 30,
 'max_iter': 1000,
 'penalty': 'l2',
 'solver': 'liblinear'}
```

Coefficients obtained after fitting the train data are following:



	coeff	feature	abs_coeff
4	-3.317895	Hague	3.317895
3	2.735974	Blair	2.735974
5	-2.310641	Europe	2.310641
1	1.849493	economic.cond.national	1.849493
0	-1.355921	age	1.355921
6	-1.161387	political.knowledge	1.161387
7	-0.129412	gender	0.129412
2	0.007859	economic.cond.household	0.007859

As we can see that the value of the coefficient for ‘Hague’ is much higher than any other variable. This means this feature is the best among these for the prediction of ‘Vote’. All the other features are arranged in the level of importance for the prediction of ‘Vote’.

The accuracy obtained from this model on the train set was 85 percent while on the test set it was 81 percent.

Since the accuracy values for test data are slightly lesser than that on the train set, we can say the model is slightly overfitting.

Linear Discriminant Analysis:

Now, for tuning the LDA model following hyperparameters are considered:

- Solver: This indicates which algorithm to perform the optimization, the default is svd, other possible values are lsqr and eigen.
- Shrinkage: Shrinkage is a form of regularization. Range of values between 0 and 1.
- Tol: Absolute threshold for a singular value of X to be considered significant, used to estimate the rank of X. Dimensions whose singular values are non-significant are discarded. Only used if the solver is ‘svd’

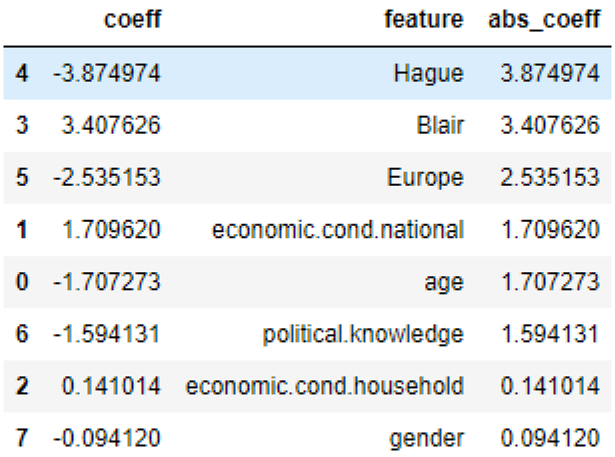
Though we tried to tune the parameter to using the Grid Search method unfortunately the results were found to be better using default parameters. Hence, the model with default values was considered for further proceedings.

1) The weights (or coefficients) are estimated so that the groups are separated as clearly as possible on the values of the discriminant functions.

2) LDA constructs an equation that minimizes the possibility of misclassifying cases into their respective classes.

Now, if we look at the coefficients obtained from LDA model, the order of feature importance is:

Hague > Blair > Europe > economic.cond.national > age > political.knowledge > economic.cond.household > gender



	coeff	feature	abs_coeff
4	-3.874974	Hague	3.874974
3	3.407626	Blair	3.407626
5	-2.535153	Europe	2.535153
1	1.709620	economic.cond.national	1.709620
0	-1.707273	age	1.707273
6	-1.594131	political.knowledge	1.594131
2	0.141014	economic.cond.household	0.141014
7	-0.094120	gender	0.094120

The accuracy score on the train set was found to be 84.34 percent and, on the test, set was 80.56 percent. Since the accuracy values for test data are slightly lesser than that on the train set, we can say the model is slightly overfitting.

1.5) Apply KNN Model and Naive Bayes Model. Interpret the inferences of each model.

Sol:

KNN-

K- Nearest Neighbors algorithm generally requires hyper-parameter tuning. Now, the thumb rule says k that is generally in the range of 1 to the square root of the number of data entries available which in our case comes out to be around 39 ($\sqrt{1525}$ = 39.05). One must keep in mind to avoid the even values of k since it can create ambiguity for the model which can result in poor performance. Hence, we have used grid search for the best value of k which in our case comes out to be 35.

Other than 'k' value, following parameters can also be looked for getting better performance:

- Weights: weight function used in prediction. Possible values:
  - 'uniform': uniform weights. All points in each neighborhood are weighted equally.
  - 'distance': weight points by the inverse of their distance. In this case, closer neighbors of a query point will have a greater influence than neighbors who are further away.
- Metric: The distance metric to use for the tree. The default metric is Minkowski, and p=2 is equivalent to the standard Euclidean metric. Other metrics are Euclidean and Manhattan.

After using grid search for the parameters, we get following results:

```
{'metric': 'manhattan', 'n_neighbors': 35, 'weights': 'uniform'}
```

Using these parameters, we have trained the model. The train set accuracy was found out to be 84.25 percent and the test set accuracy was found out to be 81.22 percent. Since the accuracies are fairly close on both train and test sets, we can say that the KNN model is doing well in terms of Over or Underfitting.

Naïve Bayes model-

Here we have used the Gaussian naïve Bayes model for the prediction of the target variable, in which likelihood probabilities are assumed to be Gaussian Normal distributed. Now, for the Naïve byes model, we can look for the smoothening parameter in grid search to tune our model-

- Var_smoothing: Portion of the largest variance of all features that are added to variances for calculation stability.

After applying grid search, we get the following results for best parameters:

```
{'var_smoothing': 0.1555676143930472}
```

Using this parameter, when we trained the model, the accuracy was improved slightly for both the train and test set. Finally, the model accuracy on the train set was found to be 84.34 percent and on the test set, 81.88 percent which seems fine. Since the values are pretty close, we can say that the model is fairly balanced in terms of over or underfitting.

1.6) Model Tuning, Bagging, and Boosting. Apply grid search on each model (include all models) and make models on best_params.

Sol:

Bagging Classifier model-

For the bagging classifier model which usually overfits on data without tuning of parameters, we have used the random forest as a base estimator. Now, for random forest the more is the number of trees in the forest the lesser will the overfitting. In our case, we tried the number of estimators as [100,200,300,400,500] for tuning of the random forest classifier. The maximum depth of each tree can also be treated as a tuning parameter. Similarly, the minimum number of samples required at leaf node, The minimum number of samples required to split an internal node, to do bootstrapping or not, and the maximum number of features used for training a tree, are some of the other parameters which were considered for tuning the random forest model.

```
{'bootstrap': True,
 'max_depth': 6,
 'max_features': 'auto',
 'min_samples_leaf': 3,
 'min_samples_split': 3,
 'n_estimators': 100,
 'random_state': 42}
```

Once this was complete, the best random forest we obtained was fed to the bagging classifier to use this estimator for training the bagging model and tuning it. After applying grid search for the number of estimators, the value obtained was

```
{'n_estimators': 100}
```

Now, after getting the best bagging model from grid search, we can train the model and get predictions from the model. The accuracy score obtained for the train set was 87.91 percent and for the test set was 81.22 percent. By looking at the accuracy scores we can say that model is still overfitting.

XGBOOST

XGBOOST is an extreme gradient boosting technique. It is gradient boosted decision trees with row and column sampling. The parameters which are there for getting better results are:

- N_estimators: Number of base learners
- Max_depth: maximum depth of base learners
- Learning_rate: represents gradient boosting learning rate
- Subsample: subsample ratio for training instance

In our analysis, after applying the grid search for the above-mentioned parameters we obtained the following results as best estimator:

```
{'learning_rate': 0.1,
 'max_depth': 3,
 'n_estimators': 100,
 'subsample': 0.5}
```

After training the model with these parameters the accuracy score improved by 3 percent. The accuracy values obtained for the train set were 88.28 percent and the test set was 81.004 percent. Since the accuracy on the train set is significantly greater than on the test set, we can say that the model is still overfitting.

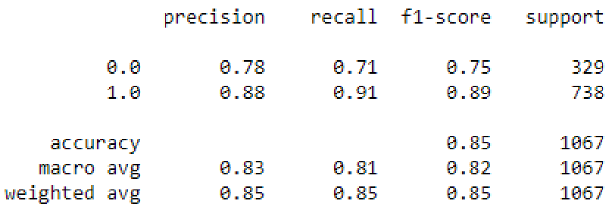
1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve, and get ROC. AUC score for each model, classification report.

Sol:

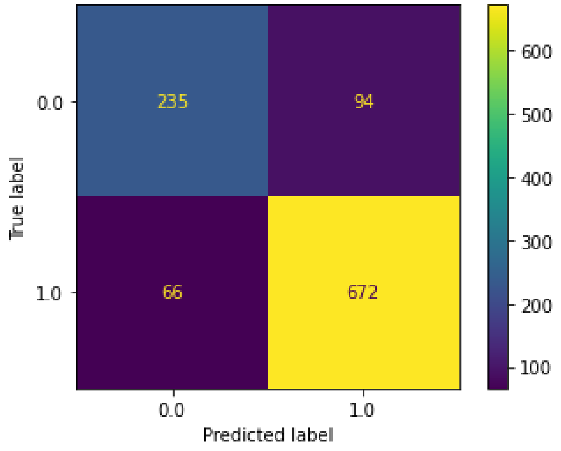
In our case, predicting both classes (i.e., 0 and 1) as accurately as possible is required because both classes are equally important. Hence, we must avoid misclassification in both classes. Therefore, the performance metric which becomes important in our case is **‘accuracy’ and ‘f1 score’.**

$$\textbf{Accuracy} = \frac{\textbf{(Correctly classified data points)}}{\textbf{(Total data points present in the dataset)}}$$

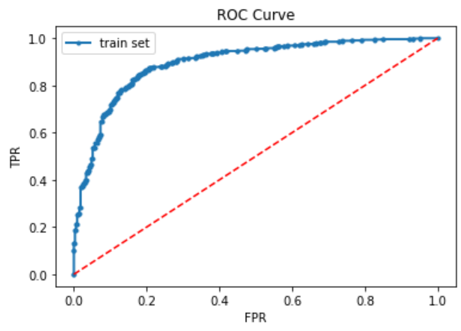
$$\textbf{F1 score} = \textbf{Harmonic mean of precision and recall} = \frac{\textbf{(2} * \textbf{precision} * \textbf{recall)}}{\textbf{(precision + recall)}}$$

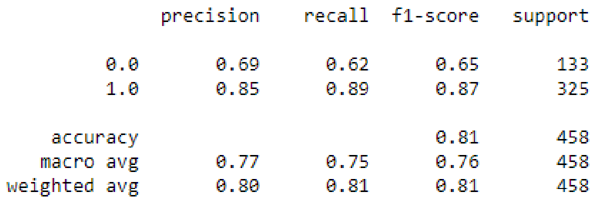


```
              precision    recall  f1-score   support

         0.0       0.78      0.71      0.75       329
         1.0       0.88      0.91      0.89       738

    accuracy                           0.85      1067
   macro avg       0.83      0.81      0.82      1067
weighted avg       0.85      0.85      0.85      1067
```

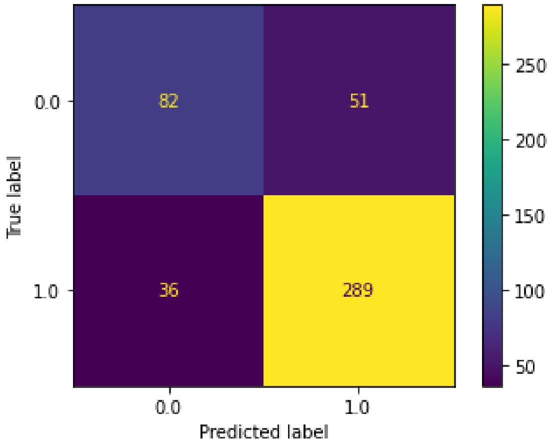


```
AUC score for Logistic regression model: 0.8962961590102224
```





```
              precision    recall  f1-score   support

         0.0       0.69      0.62      0.65       133
         1.0       0.85      0.89      0.87       325

    accuracy                           0.81       458
   macro avg       0.77      0.75      0.76       458
weighted avg       0.80      0.81      0.81       458
```



```
AUC score for Logistic regression model: 0.8720647773279352
```

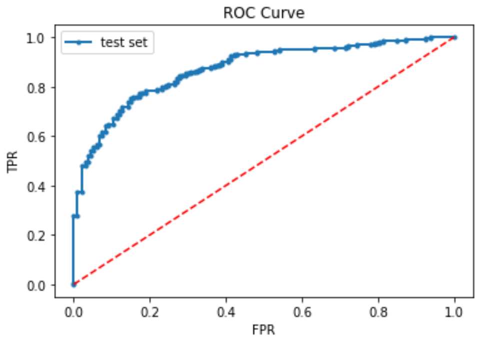
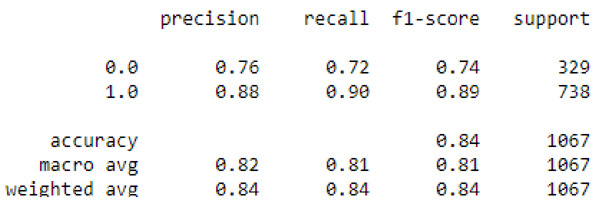
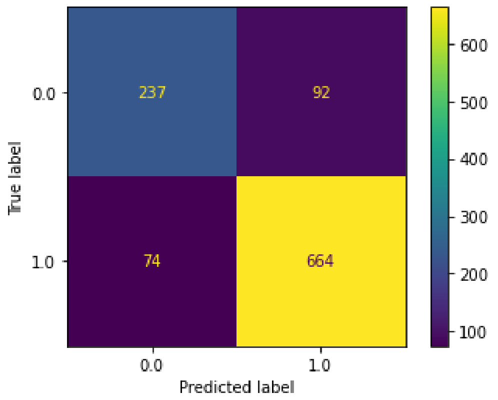


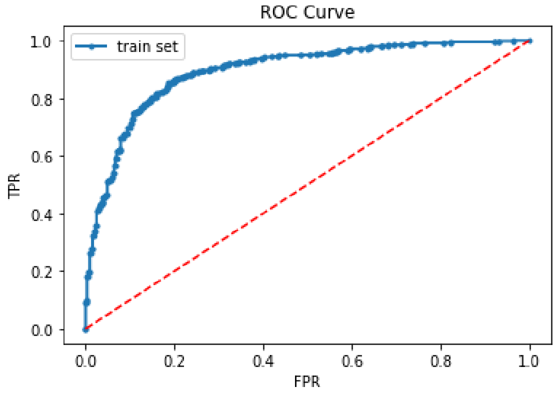
Figure 19: Performance Metrics for Logistic Regression on train set (right) and test set (left)

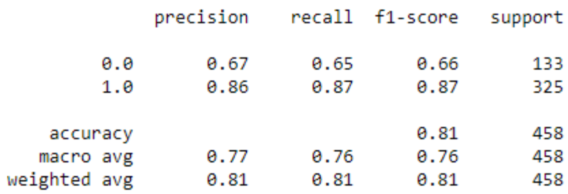


```
              precision    recall  f1-score   support

         0.0       0.76      0.72      0.74       329
         1.0       0.88      0.90      0.89       738

    accuracy                           0.84      1067
   macro avg       0.82      0.81      0.81      1067
weighted avg       0.84      0.84      0.84      1067
```

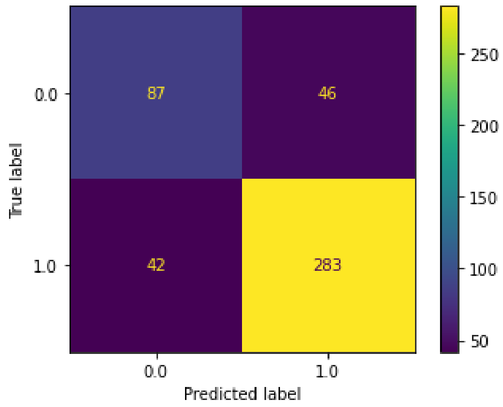


```
AUC score for LDA model: 0.895859589294981
```





```
              precision    recall  f1-score   support

         0.0       0.67      0.65      0.66       133
         1.0       0.86      0.87      0.87       325

    accuracy                           0.81       458
   macro avg       0.77      0.76      0.76       458
weighted avg       0.81      0.81      0.81       458
```



```
AUC score for LDA model: 0.8737073452862927
```

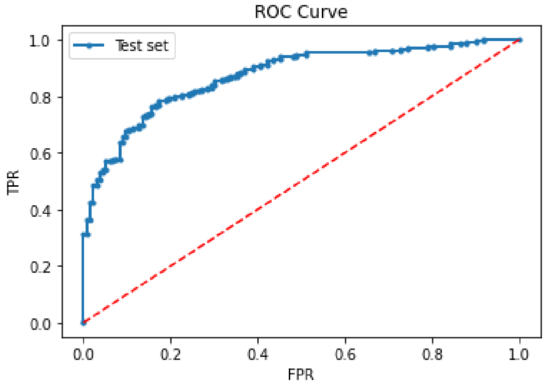
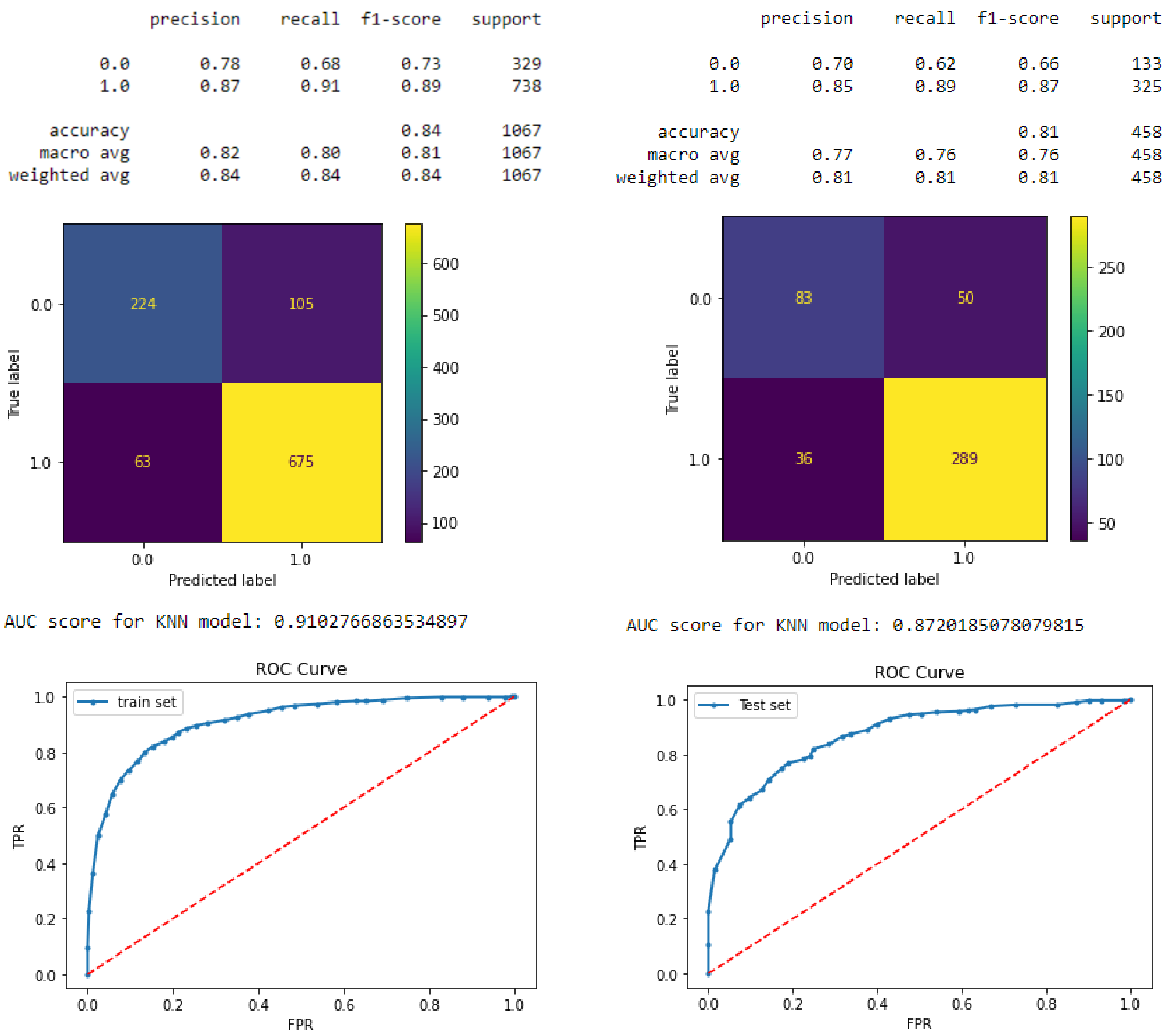


Figure 20: Performance Metrics for LDA model on train set (right) and test set (left)



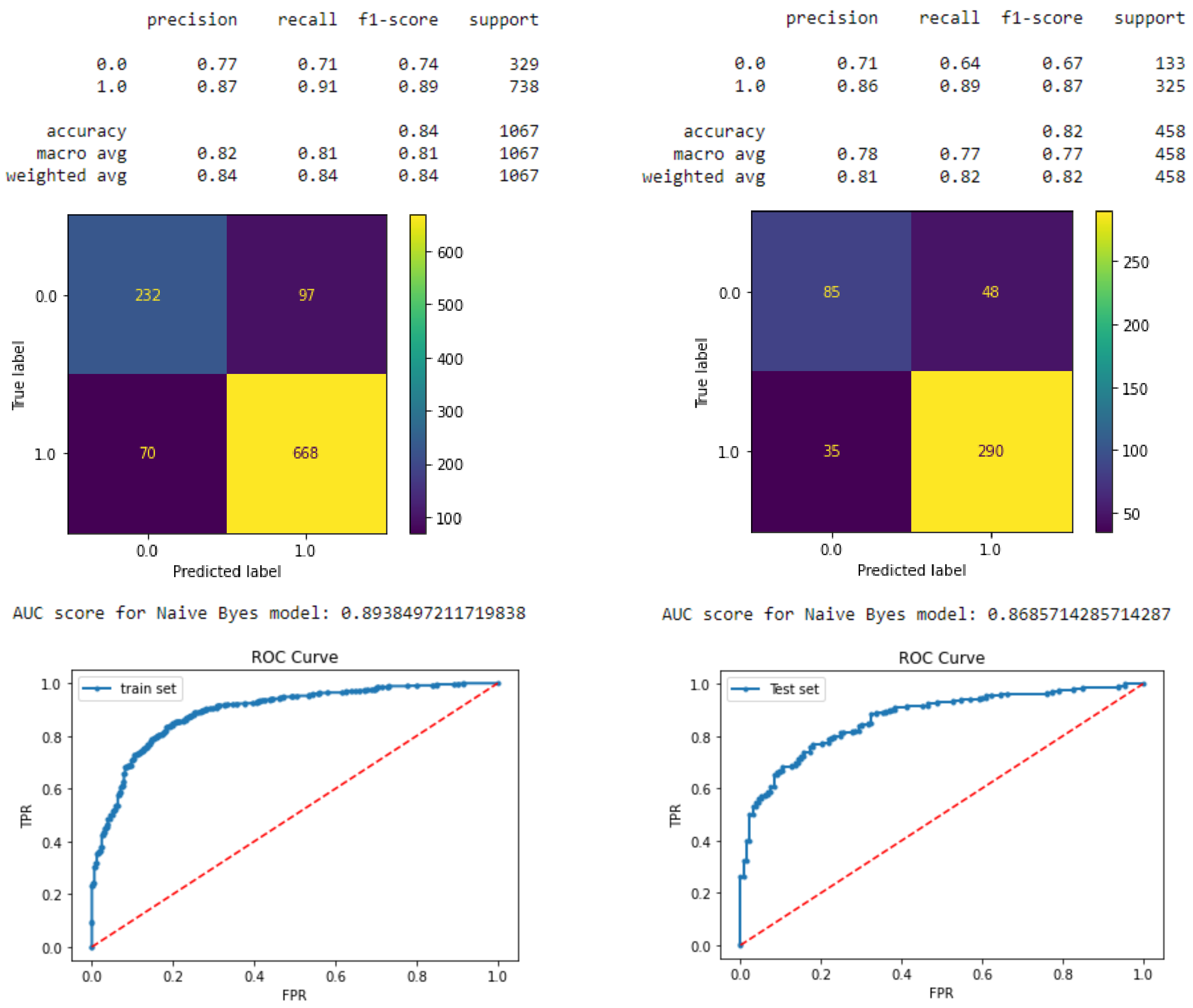
```
              precision    recall  f1-score   support

         0.0       0.78      0.68      0.73       329
         1.0       0.87      0.91      0.89       738

    accuracy                           0.84      1067
   macro avg       0.82      0.80      0.81      1067
weighted avg       0.84      0.84      0.84      1067
```

```
AUC score for KNN model: 0.9102766863534897
```

```
              precision    recall  f1-score   support

         0.0       0.70      0.62      0.66       133
         1.0       0.85      0.89      0.87       325

    accuracy                           0.81       458
   macro avg       0.77      0.76      0.76       458
weighted avg       0.81      0.81      0.81       458
```

```
AUC score for KNN model: 0.8720185078079815
```

Figure 21: Performance Metrics for KNN model on train set (left) and test set (right)



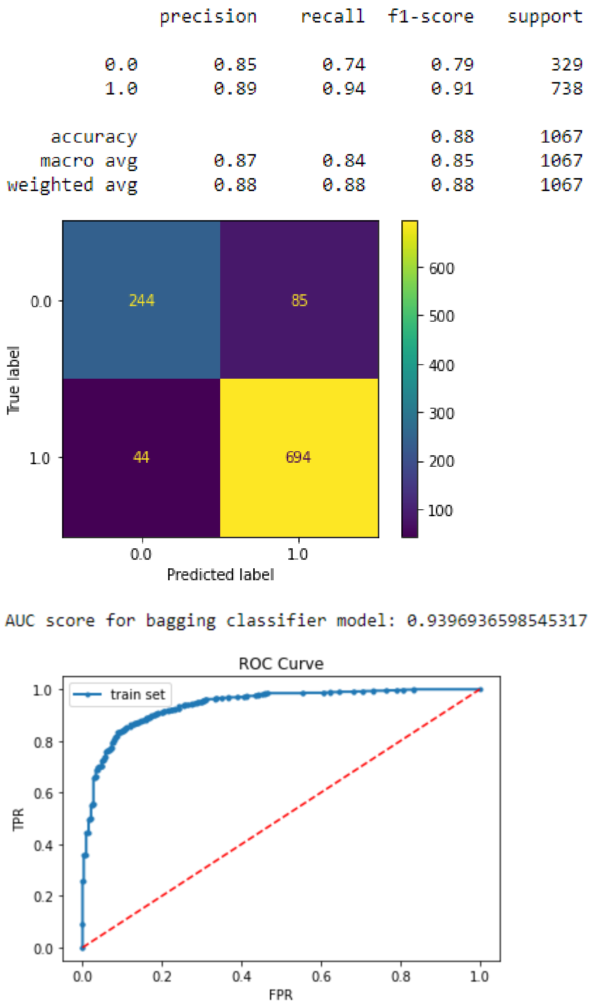
```
              precision    recall  f1-score   support

         0.0       0.77      0.71      0.74       329
         1.0       0.87      0.91      0.89       738

    accuracy                           0.84      1067
   macro avg       0.82      0.81      0.81      1067
weighted avg       0.84      0.84      0.84      1067
```

```
AUC score for Naive Byes model: 0.8938497211719838
```

```
              precision    recall  f1-score   support

         0.0       0.71      0.64      0.67       133
         1.0       0.86      0.89      0.87       325

    accuracy                           0.82       458
   macro avg       0.78      0.77      0.77       458
weighted avg       0.81      0.82      0.82       458
```

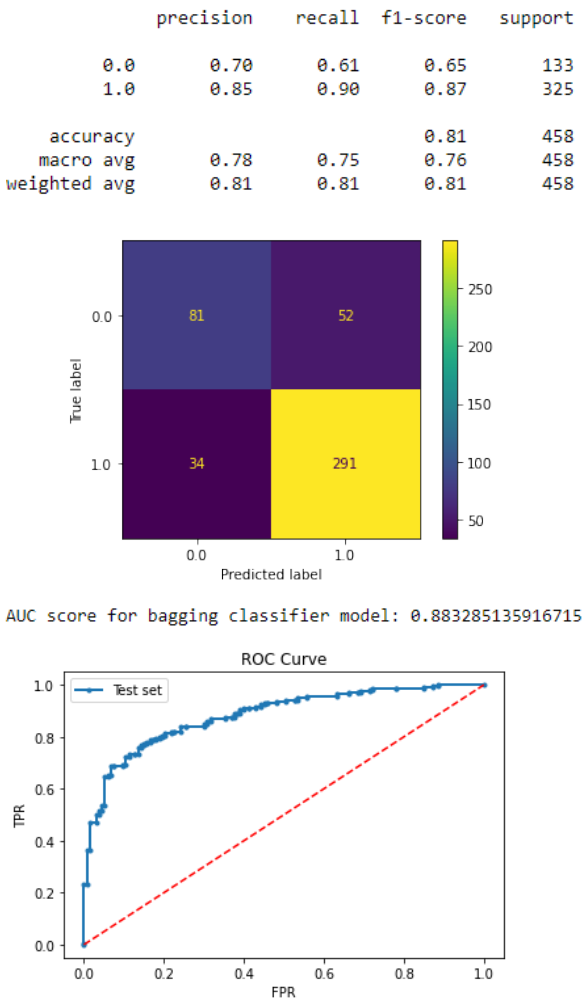
```
AUC score for Naive Byes model: 0.8685714285714287
```

Figure 22: Performance Metrics for Naive Bayes model on train set (left) and test set (right)



```
              precision    recall  f1-score   support

         0.0       0.85      0.74      0.79       329
         1.0       0.89      0.94      0.91       738

    accuracy                           0.88      1067
   macro avg       0.87      0.84      0.85      1067
weighted avg       0.88      0.88      0.88      1067
```

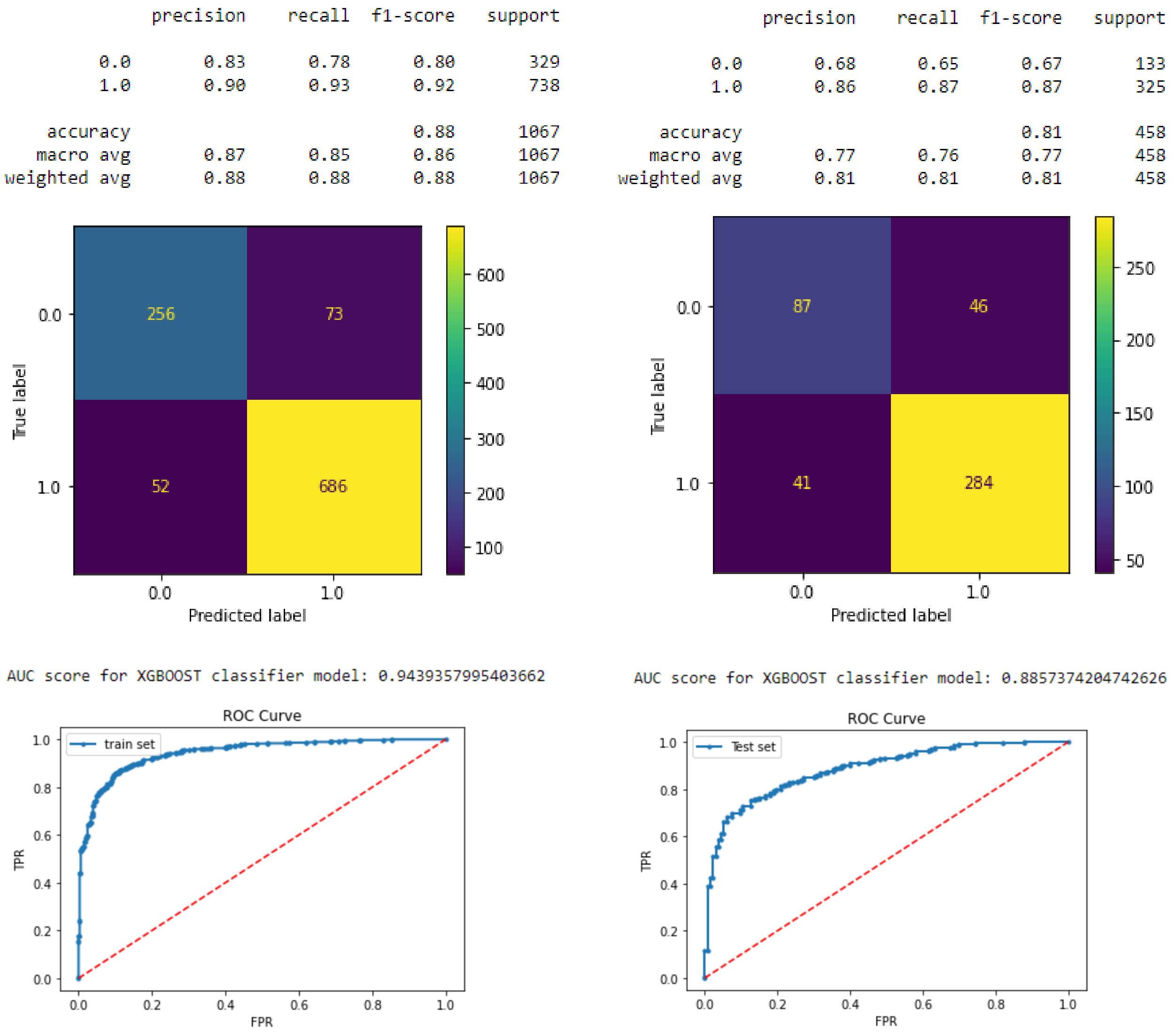
```
AUC score for bagging classifier model: 0.9396936598545317
```



```
              precision    recall  f1-score   support

         0.0       0.70      0.61      0.65       133
         1.0       0.85      0.90      0.87       325

    accuracy                           0.81       458
   macro avg       0.78      0.75      0.76       458
weighted avg       0.81      0.81      0.81       458
```

```
AUC score for bagging classifier model: 0.883285135916715
```

Figure 23: Performance Metrics for Bagging model on train set (left) and test set (right)



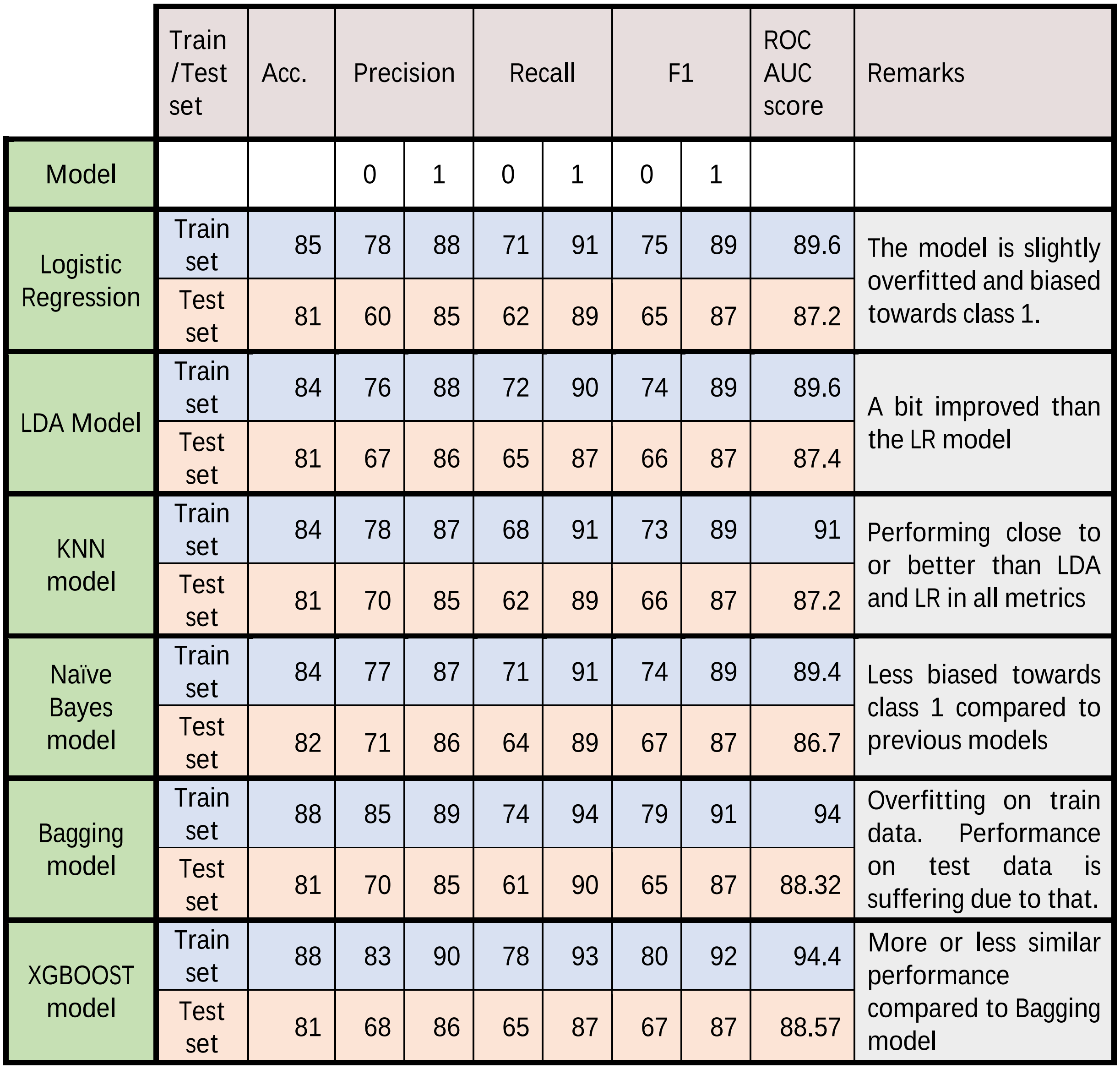
```
              precision    recall  f1-score   support

         0.0       0.83      0.78      0.80       329
         1.0       0.90      0.93      0.92       738

    accuracy                           0.88      1067
   macro avg       0.87      0.85      0.86      1067
weighted avg       0.88      0.88      0.88      1067
```

```
              precision    recall  f1-score   support

         0.0       0.68      0.65      0.67       133
         1.0       0.86      0.87      0.87       325

    accuracy                           0.81       458
   macro avg       0.77      0.76      0.77       458
weighted avg       0.81      0.81      0.81       458
```

```
AUC score for XGBOOST classifier model: 0.9439357995403662
```

```
AUC score for XGBOOST classifier model: 0.8857374204742626
```

Figure 24: Performance Metrics for XGBOOST model on train set (left) and test set (right)



	Train /Test set	Acc.	Precision		Recall		F1		ROC AUC score	Remarks
Model			0	1	0	1	0	1		
Logistic Regression	Train set	85	78	88	71	91	75	89	89.6	The model is slightly overfitted and biased towards class 1.
	Test set	81	60	85	62	89	65	87	87.2	
LDA Model	Train set	84	76	88	72	90	74	89	89.6	A bit improved than the LR model
	Test set	81	67	86	65	87	66	87	87.4	
KNN model	Train set	84	78	87	68	91	73	89	91	Performing close to or better than LDA and LR in all metrics
	Test set	81	70	85	62	89	66	87	87.2	
Naïve Bayes model	Train set	84	77	87	71	91	74	89	89.4	Less biased towards class 1 compared to previous models
	Test set	82	71	86	64	89	67	87	86.7	
Bagging model	Train set	88	85	89	74	94	79	91	94	Overfitting on train data. Performance on test data is suffering due to that.
	Test set	81	70	85	61	90	65	87	88.32	
XGBOOST model	Train set	88	83	90	78	93	80	92	94.4	More or less similar performance compared to Bagging model
	Test set	81	68	86	65	87	67	87	88.57	

Table 2: Performance Metrics for all the models and comparison

From the analysis performed and results mentioned above table following insights can be drawn:

- All the values mentioned in the above table are in percentage.
- Even after using cross-validation and sampling of features, the ensemble methods suffer to predict class 0 (Conservative party supporters) accurately.

- Since the accuracy values are a bit more or less the same in every model but XGBOOST model was able to predict class 0 a bit more accurately as the recall value is slightly higher in the case of XGBOOST model.
- After comparing all the metrics, we can conclude that **Logistic Regression model and LDA** is performing well in every aspect considering the **simplicity of the model with almost similar performance when compared to complex models** like bagging and XGBOOST model which are similar performance at the cost of high complexity.

1.8) Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective.

Sol:

Inferences and Recommendations

- Feature like assessment of Economic condition (national) was found to be more important for predicting the exit polls.
- From the analysis, it was found the political knowledge assessment wasn't really helpful in predicting the target variables. Hence, news channel can look for some other features to add in data.
- After visualization of data, it was found that the median age of people who are supporters of the Labour party, is slightly less than the median age of people who are supporters of the Conservative party. The channel can look for the reasons why is it so.
- From data visualization, it was also found that the most people who support the Conservative party, tend to have negative attitude towards European Integration. And Europe variable was also found to be important in prediction of target variable. Hence, more features like this (assessment on political issues) can be added in survey questionnaire for getting better insights on data and accurate exit polls.
- Based on this analysis, **Logistic Regression** or **LDA model** are recommended models for generalization purposes.

Problem 2

Inaugural Corpus Analysis

Information about the Corpus

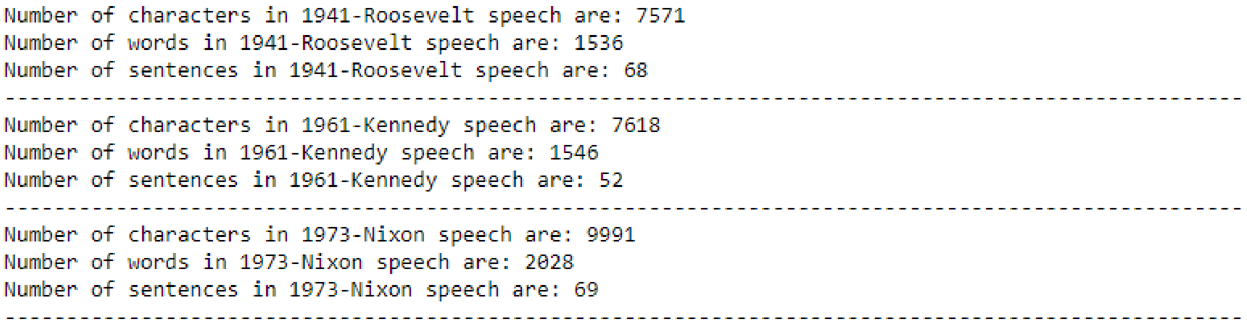
The corpus contains speeches known as US Presidential Inaugural Addresses. For our problem, we have to analyze three speeches from the corpus:

- President Franklin D. Roosevelt in 1941
- President John F. Kennedy in 1961
- President Richard Nixon in 1973

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

Sol:

For this problem, we have considered the whitespaces as characters too. Following results were obtained after analysis:



```
Number of characters in 1941-Roosevelt speech are: 7571
Number of words in 1941-Roosevelt speech are: 1536
Number of sentences in 1941-Roosevelt speech are: 68
----------------------------------------------------------------------------------------------------
Number of characters in 1961-Kennedy speech are: 7618
Number of words in 1961-Kennedy speech are: 1546
Number of sentences in 1961-Kennedy speech are: 52
----------------------------------------------------------------------------------------------------
Number of characters in 1973-Nixon speech are: 9991
Number of words in 1973-Nixon speech are: 2028
Number of sentences in 1973-Nixon speech are: 69
----------------------------------------------------------------------------------------------------
```

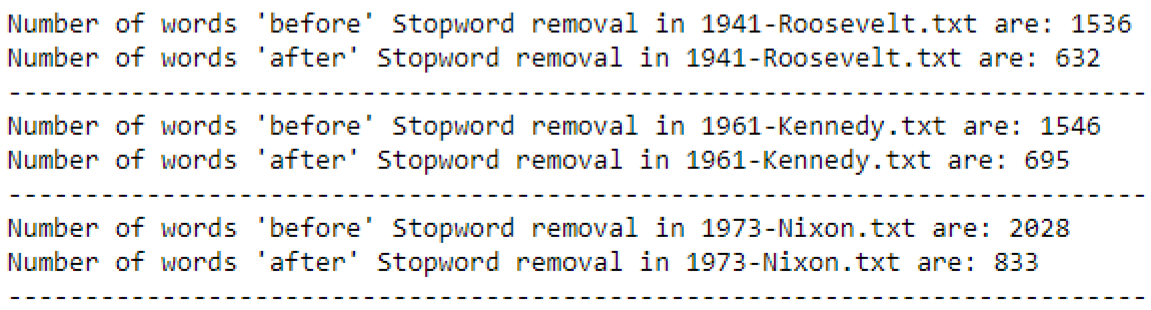
Figure 25: Number of Characters, words, and sentences in each speech

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

Sol:

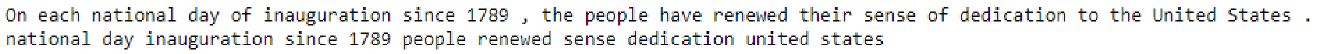
In this analysis, it was found that the text data contains additional stopwords like '—' and 'mr' which are not very informative, hence these were removed from the cleaned words. For general stopwords the stopwords provided in nltk library were used along with punctuations.

Now, the word count before and after the stopwords removal is shown below:



```
Number of words 'before' Stopword removal in 1941-Roosevelt.txt are: 1536
Number of words 'after' Stopword removal in 1941-Roosevelt.txt are: 632
--------------------------------------------------------------------------
Number of words 'before' Stopword removal in 1961-Kennedy.txt are: 1546
Number of words 'after' Stopword removal in 1961-Kennedy.txt are: 695
--------------------------------------------------------------------------
Number of words 'before' Stopword removal in 1973-Nixon.txt are: 2028
Number of words 'after' Stopword removal in 1973-Nixon.txt are: 833
--------------------------------------------------------------------------
```

Figure 26: Number of words before and after removal of stopwords



```
On each national day of inauguration since 1789 , the people have renewed their sense of dedication to the United States .
national day inauguration since 1789 people renewed sense dedication united states
```

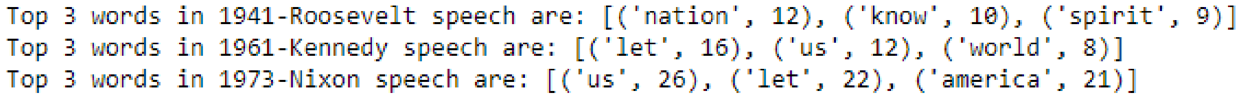
Figure 27: Sample of a sentence before (1st line) and after removal (2nd line) of stopwords

<u>Note:</u> Stemming has been avoided in this analysis since it is rule-based and some of the words make no sense at all after stemming like 'people' becomes 'peopl' which is not a word in the dictionary.

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

<u>Sol:</u>

After obtaining the frequency distribution of the bag of 'cleaned words' the following words were found to be most frequent in each speech:



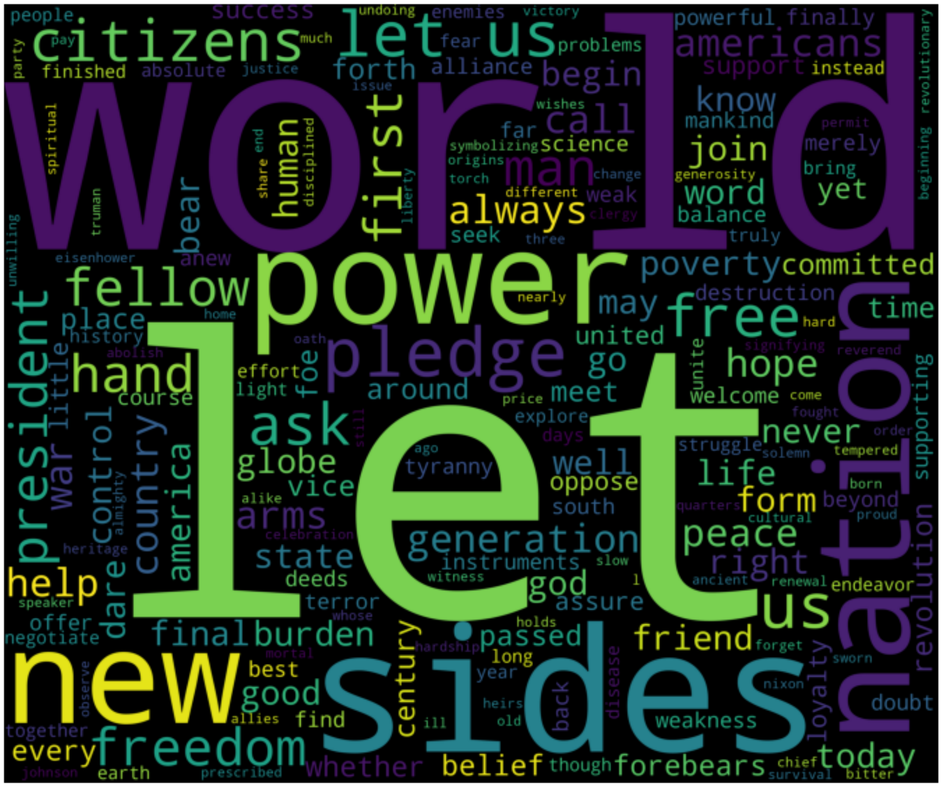
```
Top 3 words in 1941-Roosevelt speech are: [('nation', 12), ('know', 10), ('spirit', 9)]
Top 3 words in 1961-Kennedy speech are: [('let', 16), ('us', 12), ('world', 8)]
Top 3 words in 1973-Nixon speech are: [('us', 26), ('let', 22), ('america', 21)]
```

Figure 28: Top 3 words in each speech (after removing the stopwords)

2.4) Plot the word cloud of each of the three speeches, (after removing the stop words)

<u>Sol:</u> Word Cloud for each of the three speeches:

Wordcloud for speech: 1961-Kennedy


world
let
power
sides
new
nation
citizens
let us
americans
pledge
fellow
free
president
hand
ask
man
first
always
join
hope
peace
generation
freedom
us
friend
burden
final
help
arms
globe
country
control
war
america
state
people
success
undoing
enemies
victory
powerful
finally
pay
much
fear
problems
party
finished
absolute
justice
forth
alliance
support
instead
revolutionary
issue
begin
know
mankind
call
spiritual
wishes
human
merely
symbolizing
science
share
end
disciplined
origins
bring
torch
change
generosity
liberty
word
yet
truman
different
weak
balance
clergy
seek
three
truly
unwilling
beginning
eisenhower
anew
bear
poverty
committed
may
destruction
nearly
place
little
time
home
united
history
oath
hard
abolish
go
unite
signifying
reverend
effort
light
foe
around
welcome
come
course
price
meet
fought
explore
supporting
still
days
order
never
struggle
ago
solemn
tempered
tyranny
well
life
vice
oppose
born
alike
almighty
form
beyond
quarters
south
proud
cultural
celebration
heritage
instruments
right
slow
witness
deeds
dare
god
ancient
renewal
endeavor
revolution
terror
assure
speaker
offer
whose
negotiate
holds
passed
century
loyalty
mortal
hardship
forget
long
sworn
best
observe
good
year
disease
nixon
doubt
heirs
back
find
allies
ill
old
weakness
together
every
belief
though
forebears
chief
today
johnson
earth
prescribed
whether
survival
bitter