# BUSINESS REPORT

# INDEX

Page no.

1.1 Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it. (3-5)

1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers. (5-15)

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). (16)

1.4 Apply Logistic Regression and LDA (linear discriminant analysis). (17-18)

1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results. (18-19)

1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting. (20-24)

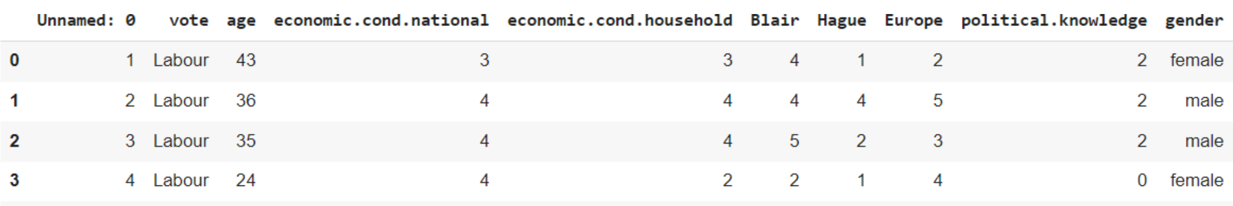
1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized. (24-31)

## 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.

### Overview of the Dataset

CNBE conducted a survey on 1525 voters to gather data on 9 different variables to figure out a pattern that would help decide a party's fate in an election. The goal of this data analysis is to construct a model from the gathered data that would predict whether a voter will vote for party A or party B. The variables in this dataset are:

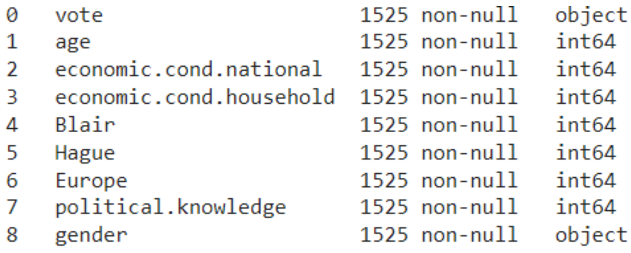
1. vote: Party choice: Conservative or Labour
2. age: in years
3. economic.cond.national: Assessment of current national economic conditions, rated from 1 to 5.
4. economic.cond.household: Assessment of current household economic conditions, rated from 1 to 5.
5. Blair: Assessment of the Labour leader, rated from 1 to 5.
6. Hague: Assessment of the Conservative leader, rated from 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, rated from 0 to 3.
9. gender: female or male.



| | 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 |

After reviewing the data, we have dropped the column "Unnamed: 0," as it is just a sequence of numbers from 0 to 1525. The shape of the data is 1525 rows and 9 columns.

### Data Types and Missing Values



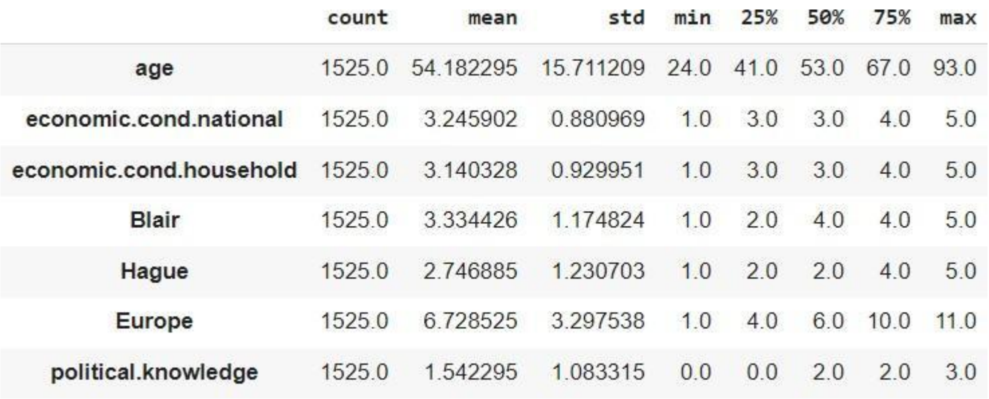
```
0   vote                     1525 non-null   object
1   age                      1525 non-null   int64
2   economic.cond.national   1525 non-null   int64
3   economic.cond.household  1525 non-null   int64
4   Blair                    1525 non-null   int64
5   Hague                    1525 non-null   int64
6   Europe                   1525 non-null   int64
7   political.knowledge      1525 non-null   int64
8   gender                   1525 non-null   object
```

Out of the 9 features in the dataset, 2 are object type, and the rest are integer type. The ordinal data types in this dataset are 'economic.cond.national,' 'economic.cond.household,' 'Blair,' 'Hague,' 'Europe,' and 'political.knowledge.'

We have checked for null values and found that there are none in the dataset.

### Summary Statistics and Skewness

Summary statistics, such as mean and standard deviation, can help identify the central tendency and spread of the dataset. The summary statistics for each variable are:



| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| age | 1525.0 | 54.182295 | 15.711209 | 24.0 | 41.0 | 53.0 | 67.0 | 93.0 |
| economic.cond.national | 1525.0 | 3.245902 | 0.880969 | 1.0 | 3.0 | 3.0 | 4.0 | 5.0 |
| economic.cond.household | 1525.0 | 3.140328 | 0.929951 | 1.0 | 3.0 | 3.0 | 4.0 | 5.0 |
| Blair | 1525.0 | 3.334426 | 1.174824 | 1.0 | 2.0 | 4.0 | 4.0 | 5.0 |
| Hague | 1525.0 | 2.746885 | 1.230703 | 1.0 | 2.0 | 2.0 | 4.0 | 5.0 |
| Europe | 1525.0 | 6.728525 | 3.297538 | 1.0 | 4.0 | 6.0 | 10.0 | 11.0 |
| political.knowledge | 1525.0 | 1.542295 | 1.083315 | 0.0 | 0.0 | 2.0 | 2.0 | 3.0 |

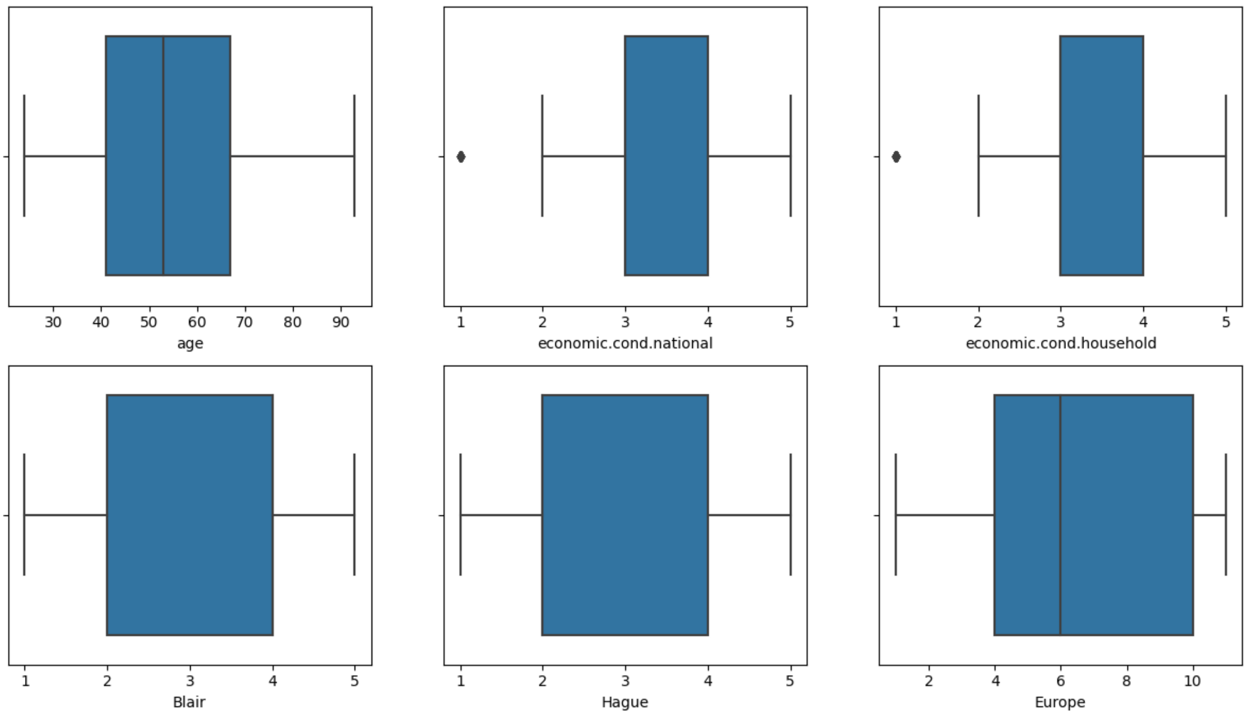
- Age: The mean age is 54.18 years, with a standard deviation of 15.71. The youngest participant is 24 years old, while the oldest is 93 years old.
- Economic condition of the nation: The mean score is 3.25, with a standard deviation of 0.88. The minimum score is 1 and the maximum is 5.
- Economic condition of the household: The mean score is 3.14, with a standard deviation of 0.93. The minimum score is 1 and the maximum is 5.
- Support for Tony Blair: The mean score is 3.33, with a standard deviation of 1.17. The minimum score is 1 and the maximum is 5.
- Support for William Hague: The mean score is 2.75, with a standard deviation of 1.23. The minimum score is 1 and the maximum is 5.
- Attitudes towards Europe: The mean score is 6.73, with a standard deviation of 3.30. The minimum score is 1 and the maximum is 11.
- Political knowledge: The mean score is 1.54, with a standard deviation of 1.08. The minimum score is 0 and the maximum is 3.

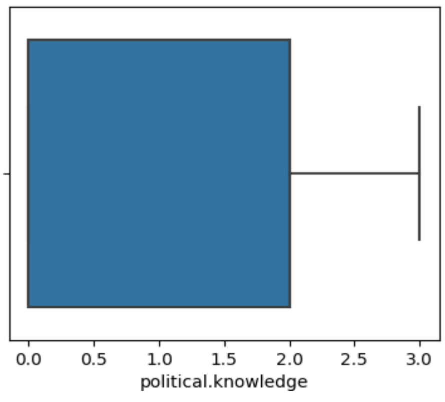
The skewness for each variable shows that the data is mostly symmetrical, with a few exceptions. The skewness for each variable is:

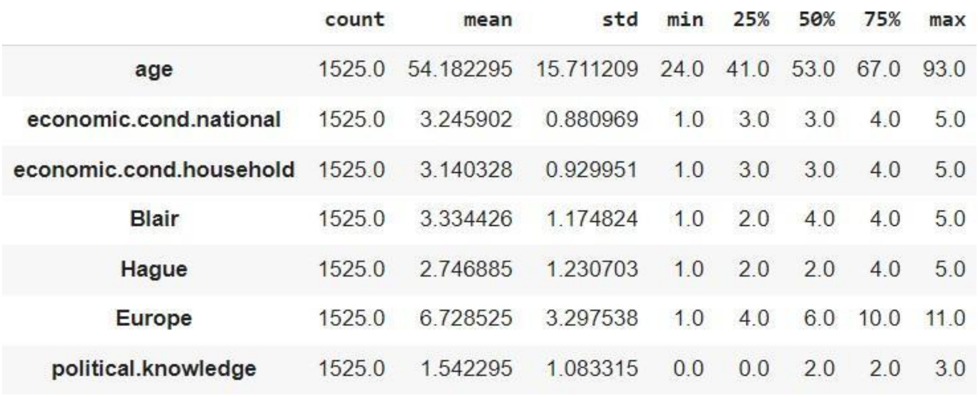
Overall, the dataset is well-organized and contains no missing values. The summary statistics show that the data is mostly symmetrical, but there are a few variables with a slight skewness. This information will be helpful in constructing a model to predict voter behaviour.

**1.2-Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also check for outliers (4 pts). Interpret the inferences for each (3 pts) Distribution plots(histogram) or similar plots for the continuous columns. Box plots. Appropriate plots for categorical variables. Inferences on each plot. Outliers proportion should be discussed, and inferences from above used plots should be there.**

**Univariate Analysis-**







| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| age | 1525.0 | 54.182295 | 15.711209 | 24.0 | 41.0 | 53.0 | 67.0 | 93.0 |
| economic.cond.national | 1525.0 | 3.245902 | 0.880969 | 1.0 | 3.0 | 3.0 | 4.0 | 5.0 |
| economic.cond.household | 1525.0 | 3.140328 | 0.929951 | 1.0 | 3.0 | 3.0 | 4.0 | 5.0 |
| Blair | 1525.0 | 3.334426 | 1.174824 | 1.0 | 2.0 | 4.0 | 4.0 | 5.0 |
| Hague | 1525.0 | 2.746885 | 1.230703 | 1.0 | 2.0 | 2.0 | 4.0 | 5.0 |
| Europe | 1525.0 | 6.728525 | 3.297538 | 1.0 | 4.0 | 6.0 | 10.0 | 11.0 |
| political.knowledge | 1525.0 | 1.542295 | 1.083315 | 0.0 | 0.0 | 2.0 | 2.0 | 3.0 |

From the above box plot and table above we can make below inferences-

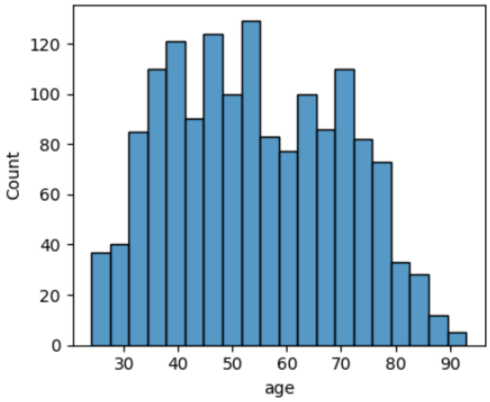
- Age: The middle 50% of the population has an age range from 41 to 67 years old.
- Economic condition (national): The middle 50% of the population rates the national economic condition between 3 and 4. Some outliers can also be seen.
- Economic condition (household): The middle 50% of the population rates their own household economic condition between 3 and 4. Some outliers can also be seen.
- Blair: The middle 50% of the population rates Blair between 2 and 4.
- Hague: The middle 50% of the population rates Hague between 2 and 4.
- Europe: The middle 50% of the population rates Europe between 4 and 10.
- Political knowledge: The middle 50% of the population has a political knowledge score between 0 and 2.

**Outliers -**

The columns for Economic.cond.national and Economic.cond.household have a few outliers, but the proportion of outliers is relatively low, at 2.42% and 4.26%, respectively.

These outliers are not going to be removed because they represent generous responses from voters and are important in maintaining the ordinal sequence of these features.

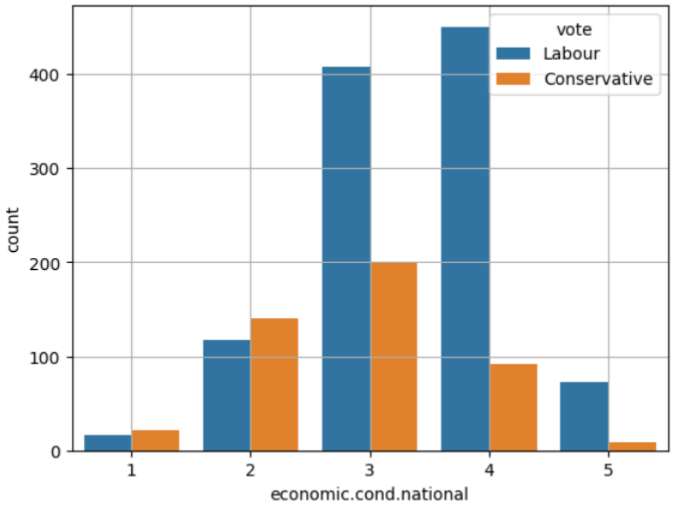
### Distribution plot (histogram) for Age column-

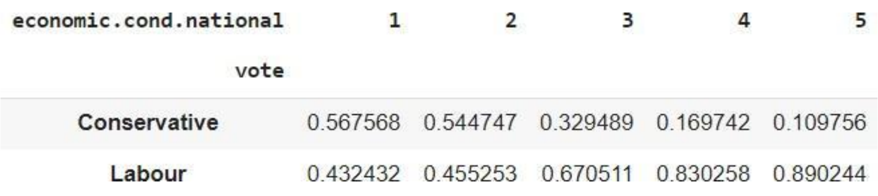


Distribution is almost normal for age column.

### Bivariate analysis-

Economic condition (national):





| economic.cond.national | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| vote | | | | | |
| Conservative | 0.567568 | 0.544747 | 0.329489 | 0.169742 | 0.109756 |
| Labour | 0.432432 | 0.455253 | 0.670511 | 0.830258 | 0.890244 |

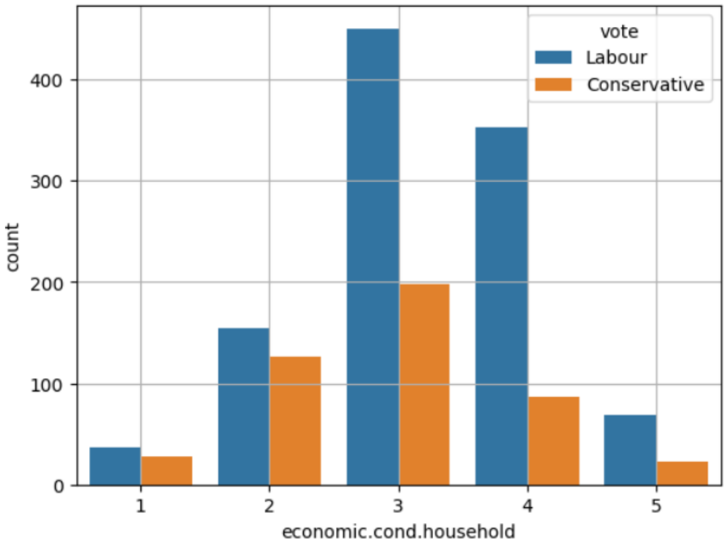
- Among people who thought the national economic conditions were poor (rated 1 or 2), a higher proportion voted for the Labour party (43.2% and 45.5%, respectively) compared to the Conservative party (56.8% and 54.5%, respectively).

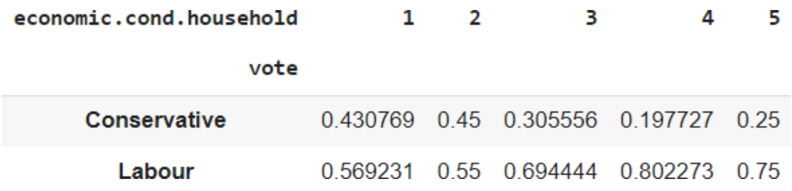
- Among people who thought the national economic conditions were good (rated 4 or 5), a higher proportion voted for the Conservative party (83.0% and 89.0%, respectively) compared to the Labour party (17.0% and 11.0%, respectively).

- Among people who thought the national economic conditions were average (rated 3), a higher proportion voted for the Labour party (67.1%) compared to the Conservative party (32.9%).

Therefore, we can infer that people's perception of the national economic conditions had a significant impact on their voting behaviour in the election. Those who believed that the economic conditions were poor were more likely to vote for the Labour party, while those who believed that the economic conditions were good were more likely to vote for the Conservative party.

**Economic condition Household-**





| economic.cond.household | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| vote | | | | | |
| Conservative | 0.430769 | 0.45 | 0.305556 | 0.197727 | 0.25 |
| Labour | 0.569231 | 0.55 | 0.694444 | 0.802273 | 0.75 |

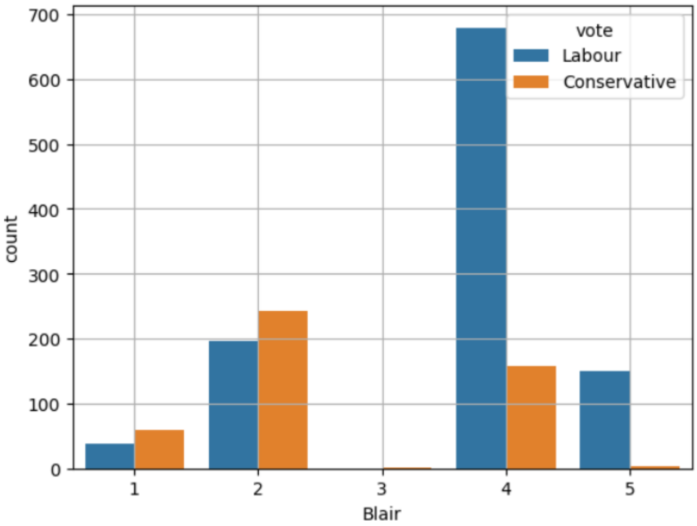
Inference-

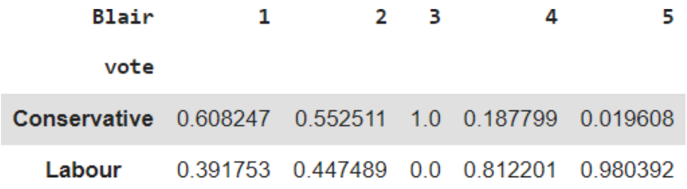
According to above table as the economic household condition improves the vote share of conservative decreases till rating 4.

As the economic household condition improves the vote share of labour party increases till rating 4.

Among people who thought the national economic household were high (rated 5), a higher proportion voted for the Labour party (75%) compared to the Conservative party (25%).

**Blair-**





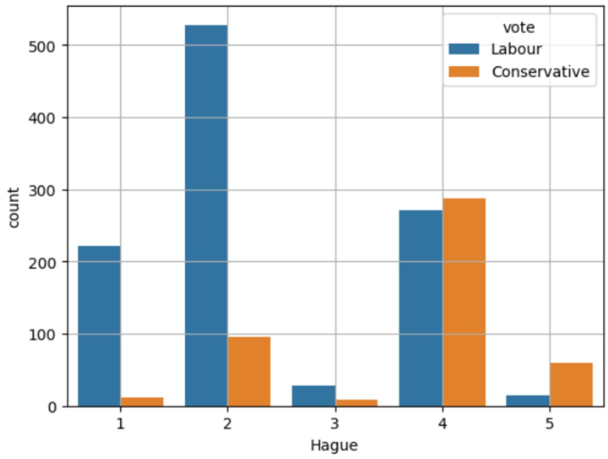
| Blair | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| vote | | | | | |
| Conservative | 0.608247 | 0.552511 | 1.0 | 0.187799 | 0.019608 |
| Labour | 0.391753 | 0.447489 | 0.0 | 0.812201 | 0.980392 |

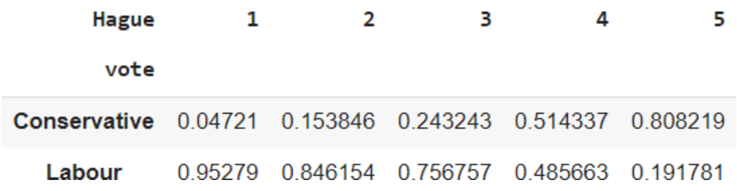
Here are some inferences we can draw from the data:

1. Respondents who rated Tony Blair positively (i.e. a rating of 4 or 5) were more likely to vote for the Labour party. Conversely, those who rated him negatively (i.e. a rating of 1 or 2) were more likely to vote for the Conservative party.

2. There is a clear difference in voting patterns between those who rate Blair positively and those who rate him negatively. For respondents who rated Blair a 1 or 2, a majority voted for the Conservative party. For those who rated Blair a 4 or 5, a majority voted for the Labour party.

**Hague-**





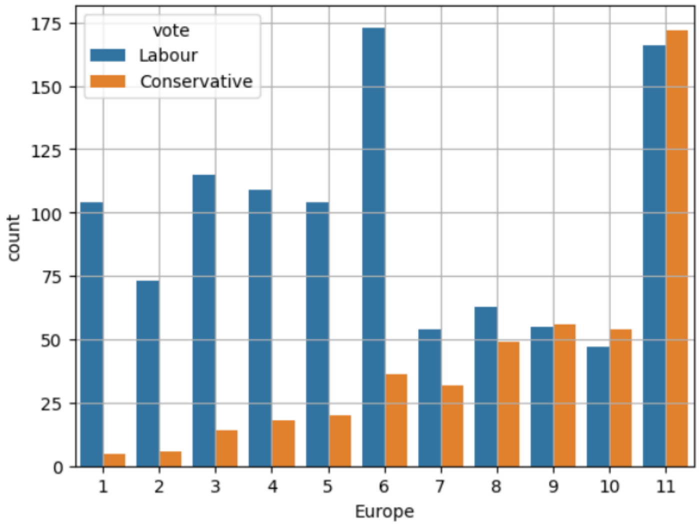
| Hague | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| vote | | | | | |
| Conservative | 0.04721 | 0.153846 | 0.243243 | 0.514337 | 0.808219 |
| Labour | 0.95279 | 0.846154 | 0.756757 | 0.485663 | 0.191781 |

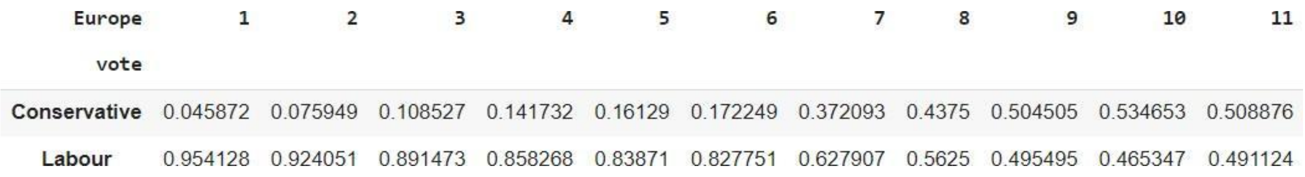
**Inference-**

As the rating of the Hague increased, the percentage of votes going to the Labour party decreased, while the percentage of votes going to the Conservative party increased.

Furthermore, the Conservative party was more successful in retaining a higher vote share among voters who gave the highest rating of 5 to The Hague.

**Europe-**





| Europe | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| vote | | | | | | | | | | | |
| Conservative | 0.045872 | 0.075949 | 0.108527 | 0.141732 | 0.16129 | 0.172249 | 0.372093 | 0.4375 | 0.504505 | 0.534653 | 0.508876 |
| Labour | 0.954128 | 0.924051 | 0.891473 | 0.858268 | 0.83871 | 0.827751 | 0.627907 | 0.5625 | 0.495495 | 0.465347 | 0.491124 |

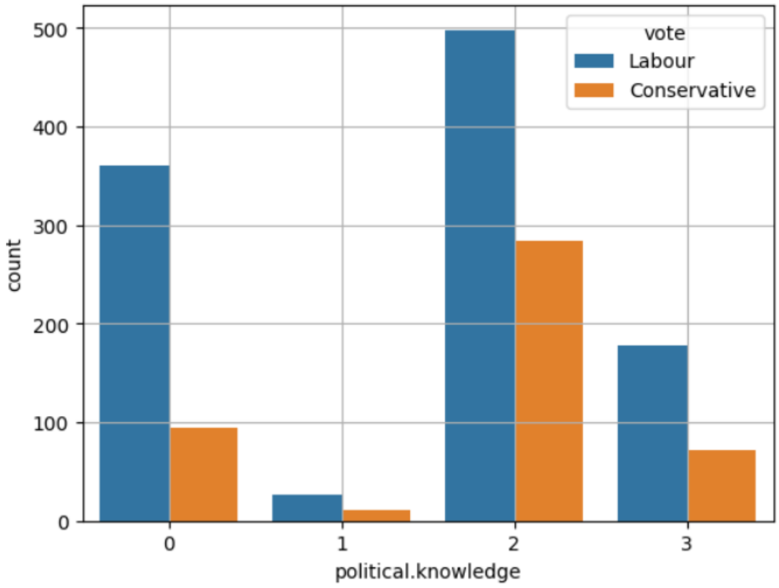
**Inference-**

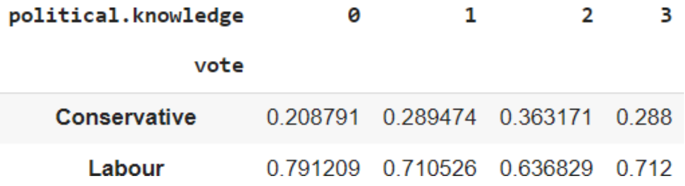
A score of 1 on this scale indicates a strong pro-European sentiment, while a score of 11 indicates a strong Eurosceptic sentiment.

At the lower end of the scale (scores 1-4), both parties receive a relatively low percentage of votes, with the Conservatives receiving less than 15% of the vote and Labour receiving around 85-90% of the vote. As Eurosceptic sentiment increases (scores 5-11), the Conservative vote share increases steadily, with the party receiving over 50% of the vote at the highest levels of Euroscepticism. In contrast, the Labour vote share decreases as Eurosceptic sentiment increases, with the party receiving less than 50% of the vote at the highest levels of Euroscepticism.

Overall, this suggests that Eurosceptic sentiment is associated with higher levels of support for the Conservative party and lower levels of support for the Labour party.

**Political knowledge-**



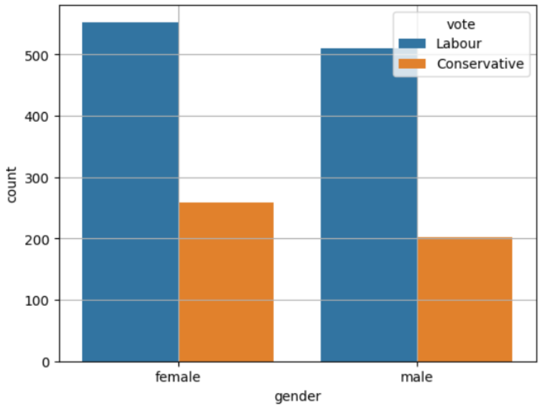


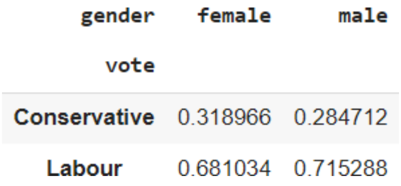
| political.knowledge | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| vote | | | | |
| Conservative | 0.208791 | 0.289474 | 0.363171 | 0.288 |
| Labour | 0.791209 | 0.710526 | 0.636829 | 0.712 |

**Inference-**

Political knowledge doesn’t seem to be a differentiating parameter while deciding whom would a person vote.

**Gender-**

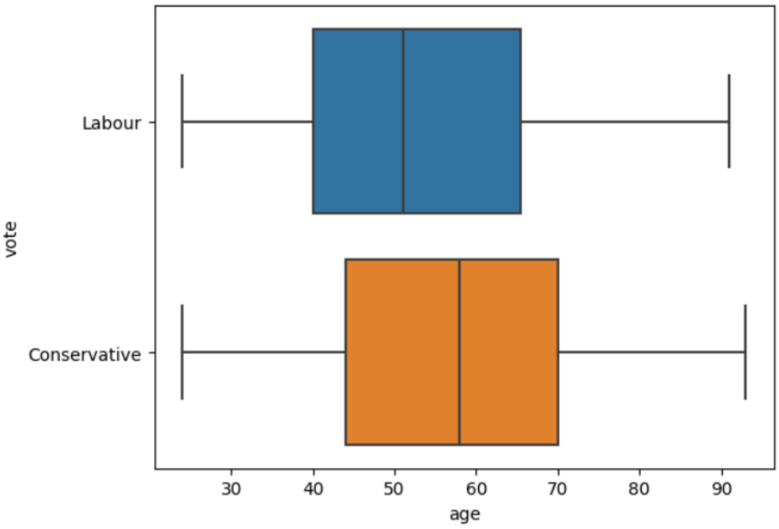


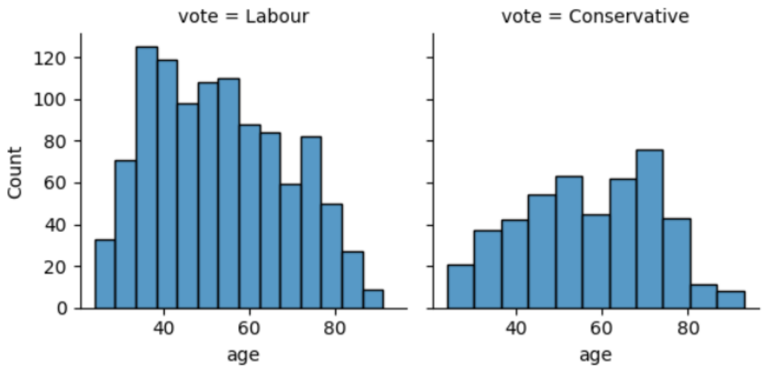


| gender | female | male |
|---|---|---|
| vote | | |
| Conservative | 0.318966 | 0.284712 |
| Labour | 0.681034 | 0.715288 |

**Inference**-Gender doesn’t make any differentiation, to whom a person will vote.

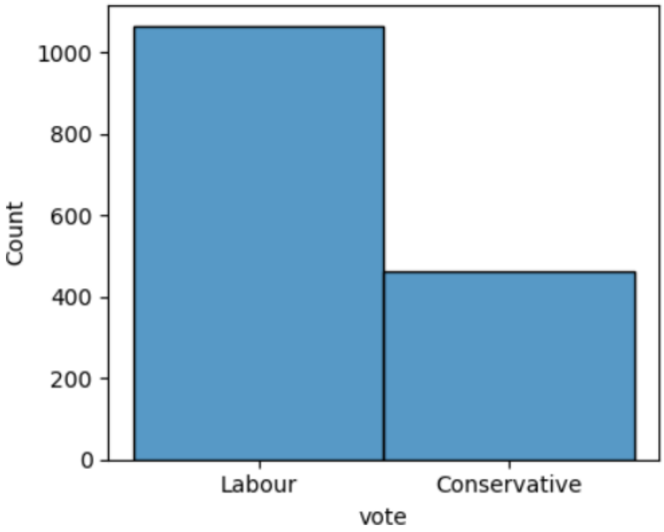
**Age-**





**Inference-** It's quite evident that people with a slight higher age group prefer to go for conservative party, and younger people like labour party more.

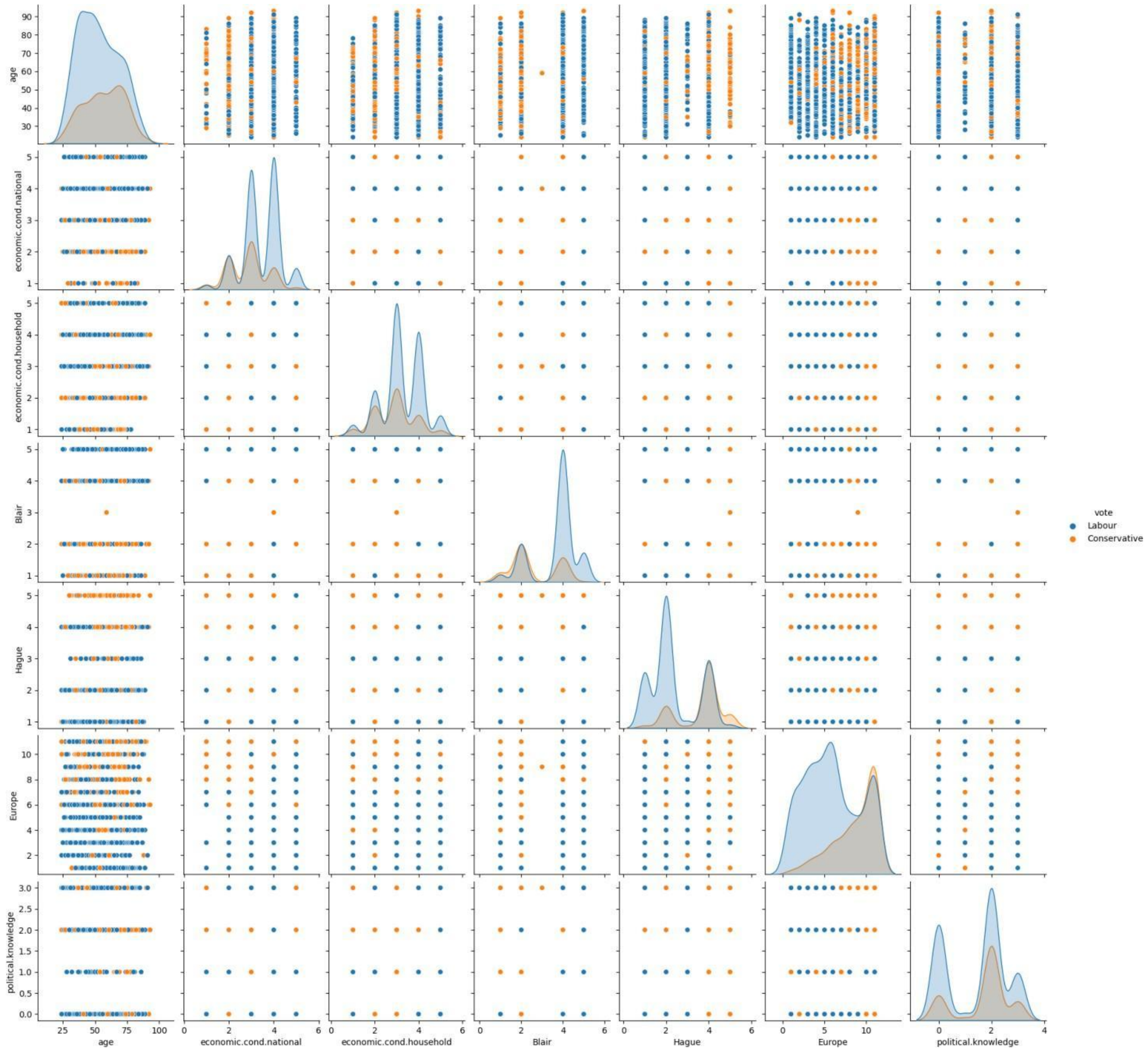
**Vote-**



**Percentage vote share is as below-**

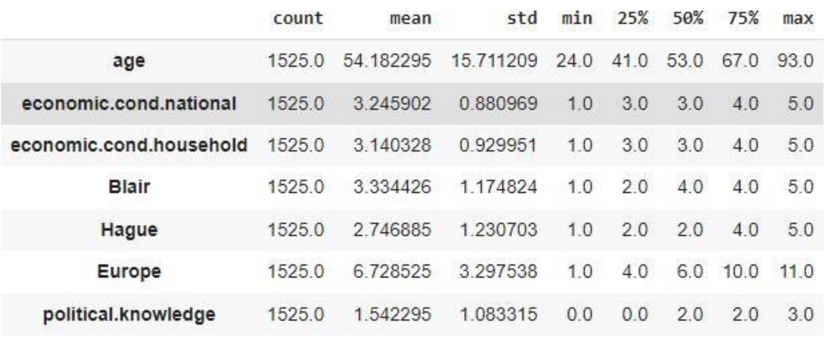
Labour - 69.71%

Conservative 30.29%



We can clearly see a difference in distribution for ‘Europe’, ‘Hague’, ‘Blair’ and ‘Age’ columns.

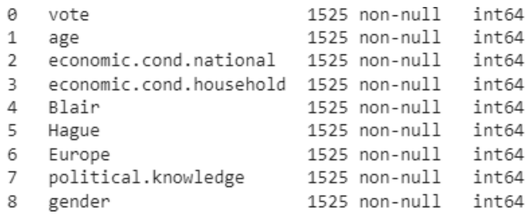
**1.3-Encode the data (having string values) for Modelling. Is Scaling necessary here or not?( 2 pts), Data Split: Split the data into train and test (70:30) (2 pts). The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for example std dev, variance, etc. Should justify whether there is a necessity for scaling. Object data should be converted into categorical/numerical data to fit in the models.**



| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **age** | 1525.0 | 54.182295 | 15.711209 | 24.0 | 41.0 | 53.0 | 67.0 | 93.0 |
| **economic.cond.national** | 1525.0 | 3.245902 | 0.880969 | 1.0 | 3.0 | 3.0 | 4.0 | 5.0 |
| **economic.cond.household** | 1525.0 | 3.140328 | 0.929951 | 1.0 | 3.0 | 3.0 | 4.0 | 5.0 |
| **Blair** | 1525.0 | 3.334426 | 1.174824 | 1.0 | 2.0 | 4.0 | 4.0 | 5.0 |
| **Hague** | 1525.0 | 2.746885 | 1.230703 | 1.0 | 2.0 | 2.0 | 4.0 | 5.0 |
| **Europe** | 1525.0 | 6.728525 | 3.297538 | 1.0 | 4.0 | 6.0 | 10.0 | 11.0 |
| **political.knowledge** | 1525.0 | 1.542295 | 1.083315 | 0.0 | 0.0 | 2.0 | 2.0 | 3.0 |

Scaling is necessary for features with a larger range or a high standard deviation, such as Age and Europe. However, features such as economic.cond.national, Blair, Hague and economic.cond.household may not require scaling as they are already on a similar scale.
So we are going to scale the data. We will use min-max scaler.

We encoded the gender and vote feature, say for gender – (female-0, Male-1)
For Vote feature – (Labour-0, Conservative-1)



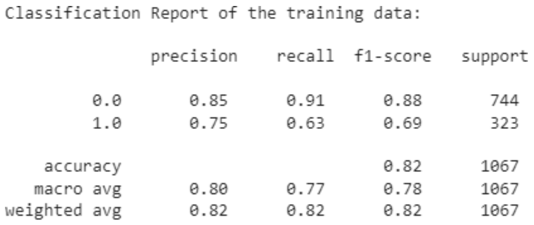
```
0   vote                     1525 non-null   int64
1   age                      1525 non-null   int64
2   economic.cond.national   1525 non-null   int64
3   economic.cond.household  1525 non-null   int64
4   Blair                    1525 non-null   int64
5   Hague                    1525 non-null   int64
6   Europe                   1525 non-null   int64
7   political.knowledge      1525 non-null   int64
8   gender                   1525 non-null   int64
```

Both gender and vote are now integer type-

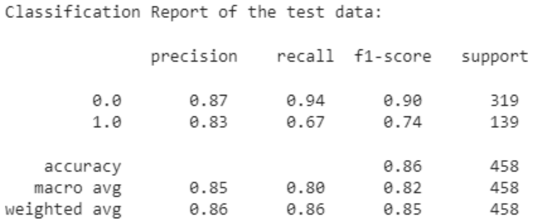
We now split the data into test and train data with ratio (70:30).
We get shape of train data and test data as-
(1067, 8)
(458, 8)

**1.4-Apply Logistic Regression and LDA (Linear Discriminant Analysis) (2 pts). Interpret the inferences of both model s (2 pts). Successful implementation of each model. Logical reason should be shared if any custom changes are made to the parameters while building the model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)**

### Logistic regression-

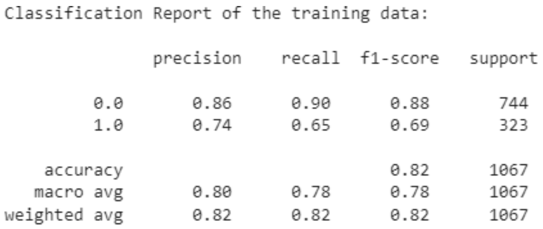


```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.85      0.91      0.88       744
         1.0       0.75      0.63      0.69       323

    accuracy                           0.82      1067
   macro avg       0.80      0.77      0.78      1067
weighted avg       0.82      0.82      0.82      1067
```

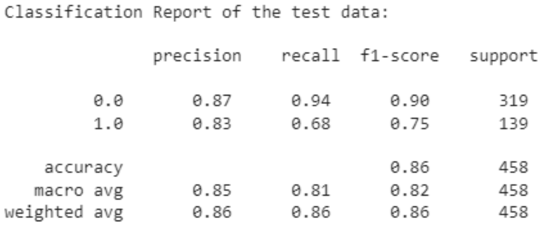


```
Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.87      0.94      0.90       319
         1.0       0.83      0.67      0.74       139

    accuracy                           0.86       458
   macro avg       0.85      0.80      0.82       458
weighted avg       0.86      0.86      0.85       458
```

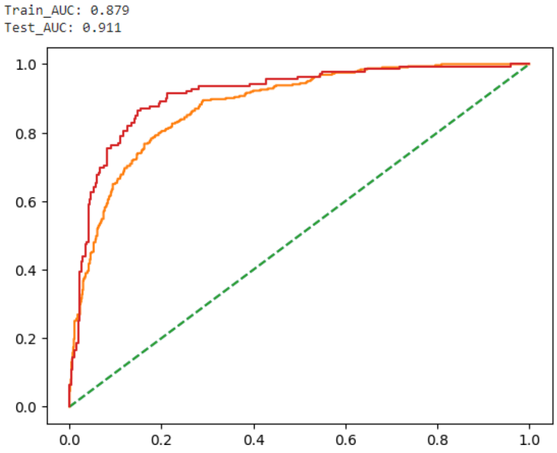
### LDA



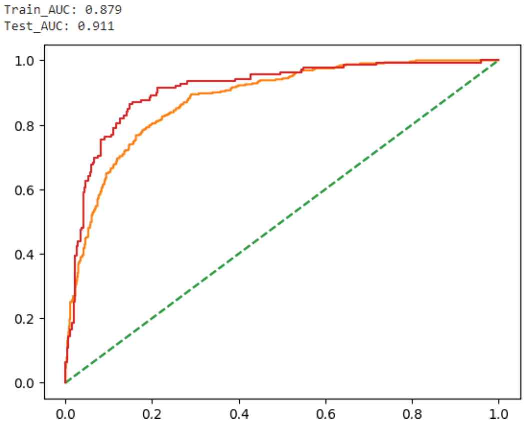
```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.86      0.90      0.88       744
         1.0       0.74      0.65      0.69       323

    accuracy                           0.82      1067
   macro avg       0.80      0.78      0.78      1067
weighted avg       0.82      0.82      0.82      1067
```



```
Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.87      0.94      0.90       319
         1.0       0.83      0.68      0.75       139

    accuracy                           0.86       458
   macro avg       0.85      0.81      0.82       458
weighted avg       0.86      0.86      0.86       458
```



```
Train_AUC: 0.879
Test_AUC: 0.911
```



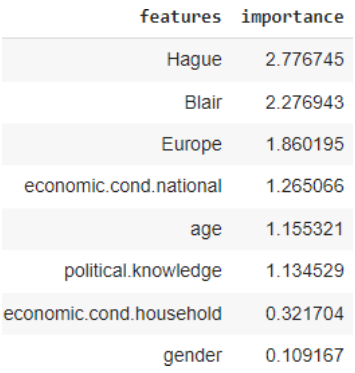
```
Train_AUC: 0.879
Test_AUC: 0.911
```

It could clearly be seen that both training and testing accuracy for both the models is 82% and 86% respectively.
This model is under-fit; it is fully capturing the patterns available in training data.

AUC score for both models is also same, and curves are identical too.

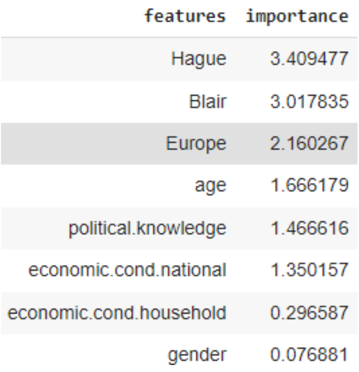
Let’s check feature importance for both the models-

Logistic Regression



| features | importance |
|---|---|
| Hague | 2.776745 |
| Blair | 2.276943 |
| Europe | 1.860195 |
| economic.cond.national | 1.265066 |
| age | 1.155321 |
| political.knowledge | 1.134529 |
| economic.cond.household | 0.321704 |
| gender | 0.109167 |

LDA

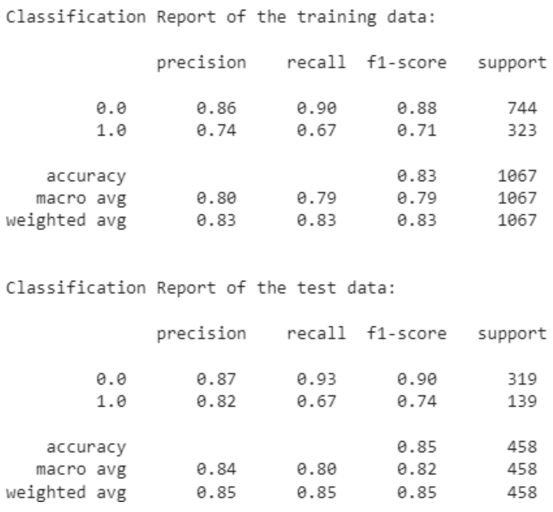


| features | importance |
|---|---|
| Hague | 3.409477 |
| Blair | 3.017835 |
| Europe | 2.160267 |
| age | 1.666179 |
| political.knowledge | 1.466616 |
| economic.cond.national | 1.350157 |
| economic.cond.household | 0.296587 |
| gender | 0.076881 |

According to both the models Hague, Blair and Europe are the top 3 best features in prediction.

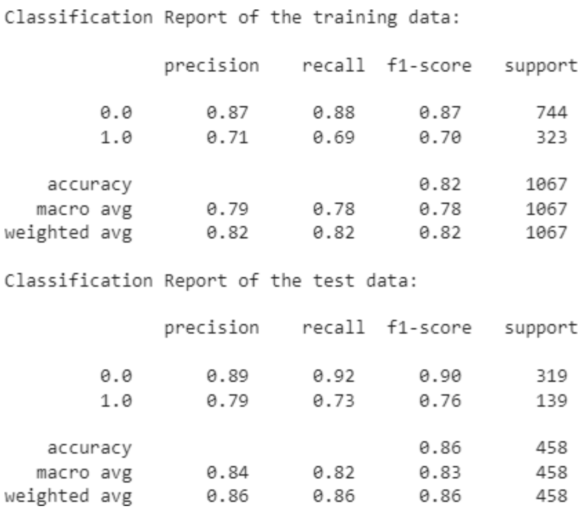
## 1.5-Apply KNN Model and Naïve Bayes Model (2pts). Interpret the inferences of each model (2 pts). Successful implementation of each model. Logical reason should be shared if any custom changes are made to the parameters while building the model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)

### KNN



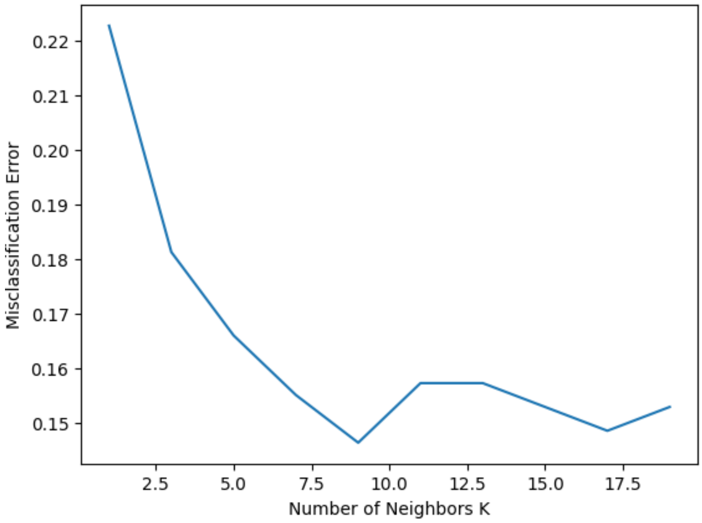
```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.86      0.90      0.88       744
         1.0       0.74      0.67      0.71       323

    accuracy                           0.83      1067
   macro avg       0.80      0.79      0.79      1067
weighted avg       0.83      0.83      0.83      1067

Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.87      0.93      0.90       319
         1.0       0.82      0.67      0.74       139

    accuracy                           0.85       458
   macro avg       0.84      0.80      0.82       458
weighted avg       0.85      0.85      0.85       458
```

### Naïve Bayes



```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.87      0.88      0.87       744
         1.0       0.71      0.69      0.70       323

    accuracy                           0.82      1067
   macro avg       0.79      0.78      0.78      1067
weighted avg       0.82      0.82      0.82      1067

Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.89      0.92      0.90       319
         1.0       0.79      0.73      0.76       139

    accuracy                           0.86       458
   macro avg       0.84      0.82      0.83       458
weighted avg       0.86      0.86      0.86       458
```

For KNN we have taken K Nearest neighbour as 9. According to below graph-



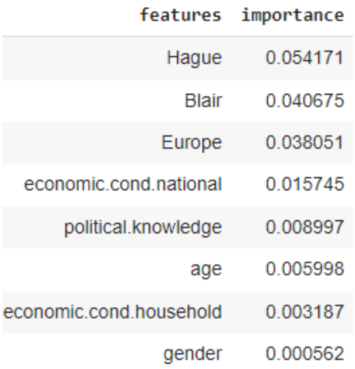
Rest for both the models the train and test accuracies are-

| Model | Train accuracy | Test accuracy |
|---|---|---|
| KNN | 83 | 85 |
| Naïve Bayes | 82 | 86 |

Both these models are under-fit, again the test accuracy is grater then train accuracy.

Taking 9 points in vicinity and using the Euclidian distance method we are able to get 83 and 85 percent train and test accuracies.

For Naïve Bayes below is the feature importance-



| features | importance |
|---|---|
| Hague | 0.054171 |
| Blair | 0.040675 |
| Europe | 0.038051 |
| economic.cond.national | 0.015745 |
| political.knowledge | 0.008997 |
| age | 0.005998 |
| economic.cond.household | 0.003187 |
| gender | 0.000562 |

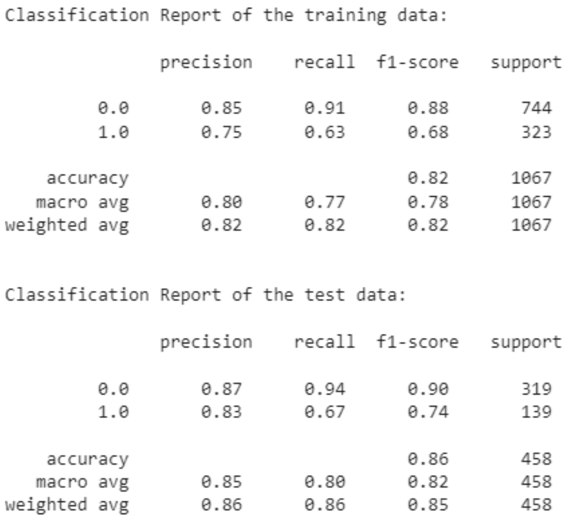
Again Hague, Blair and Europe are top 3 features in prediction for naïve Bayes.

**1.6-Model Tuning (4 pts) , Bagging ( 1.5 pts) and Boosting (1.5 pts). Apply grid search on each model (include all models) and make models on best_params. Compare and comment on performances of all. Comment on feature importance if applicable. Successful implementation of both algorithms along with inferences and comments on the model performances.**

Let’s first tune the 4 initial models with **Gridsearch_CV**-

1. **Logistic Regression with tuning-**

Tuned Hyper parameters: {'C': 0.615848211066026, 'penalty': 'l1', 'solver': 'saga'}

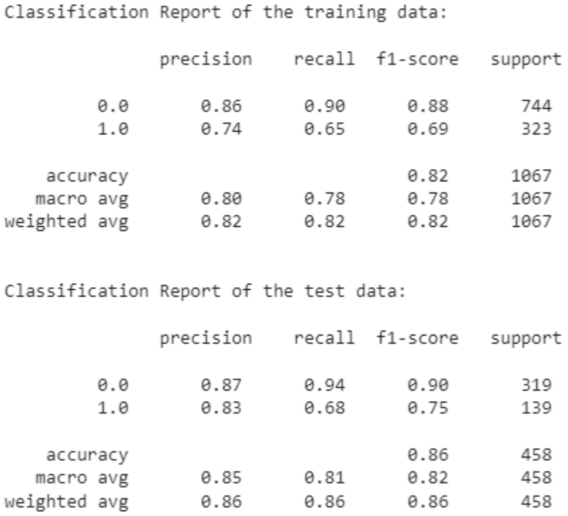


```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.85      0.91      0.88       744
         1.0       0.75      0.63      0.68       323

    accuracy                           0.82      1067
   macro avg       0.80      0.77      0.78      1067
weighted avg       0.82      0.82      0.82      1067


Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.87      0.94      0.90       319
         1.0       0.83      0.67      0.74       139

    accuracy                           0.86       458
   macro avg       0.85      0.80      0.82       458
weighted avg       0.86      0.86      0.85       458
```

For logistic regression we still get the same accuracies as we were getting before tuning the model.

2. **LDA with tuning -**

Tuned Hyper parameters: {'solver': 'svd'}



```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.86      0.90      0.88       744
         1.0       0.74      0.65      0.69       323

    accuracy                           0.82      1067
   macro avg       0.80      0.78      0.78      1067
weighted avg       0.82      0.82      0.82      1067


Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.87      0.94      0.90       319
         1.0       0.83      0.68      0.75       139

    accuracy                           0.86       458
   macro avg       0.85      0.81      0.82       458
weighted avg       0.86      0.86      0.86       458
```

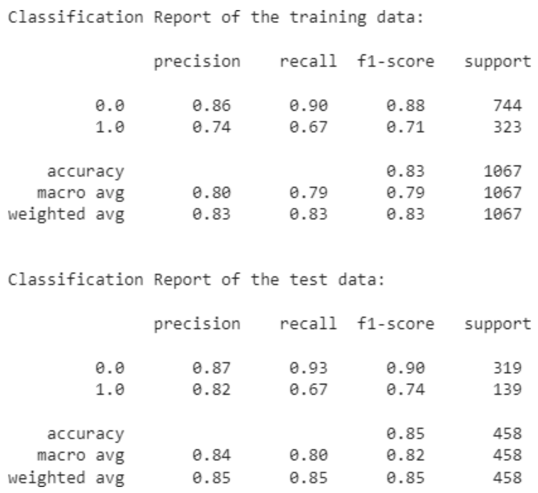
For LDA we still get the same accuracies as we were getting before tuning the model.

### 3. KNN with tuning -

Tuned Hyper parameters: {'n_neighbors': 16, 'weights': 'uniform'}

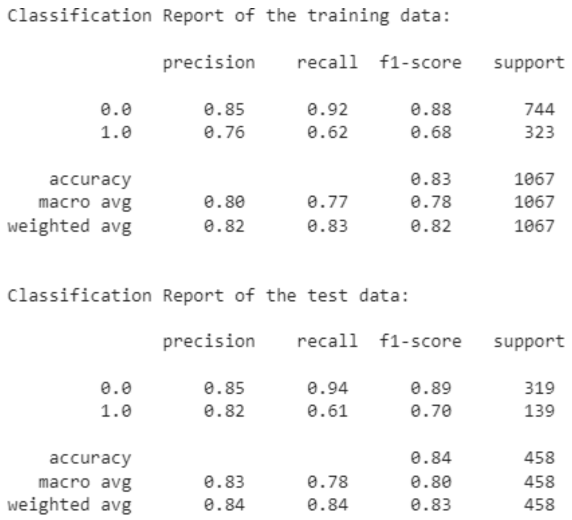
We get a slight improvement in accuracies with these hyper-parameters-
Previously we were getting

KNN- previous (K=9)



```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.86      0.90      0.88       744
         1.0       0.74      0.67      0.71       323

    accuracy                           0.83      1067
   macro avg       0.80      0.79      0.79      1067
weighted avg       0.83      0.83      0.83      1067

Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.87      0.93      0.90       319
         1.0       0.82      0.67      0.74       139

    accuracy                           0.85       458
   macro avg       0.84      0.80      0.82       458
weighted avg       0.85      0.85      0.85       458
```

KNN With Gridsearch_CV(K=16)



```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.85      0.92      0.88       744
         1.0       0.76      0.62      0.68       323

    accuracy                           0.83      1067
   macro avg       0.80      0.77      0.78      1067
weighted avg       0.82      0.83      0.82      1067

Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.85      0.94      0.89       319
         1.0       0.82      0.61      0.70       139

    accuracy                           0.84       458
   macro avg       0.83      0.78      0.80       458
weighted avg       0.84      0.84      0.83       458
```

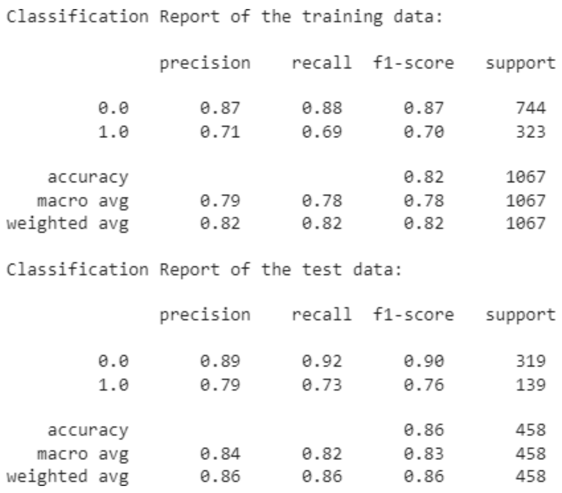
With K-16 our test accuracy is now closer to train accuracy.

Train accuracy = 83%, Test accuracy= 84%.

It is a bit less under-fit model than the previous one where K was equal to 9.

### 4. Naïve Bayes with tuning -

There are as such hyper-parameters for naïve Bayes that can be tuned so its accuracy is as it is.



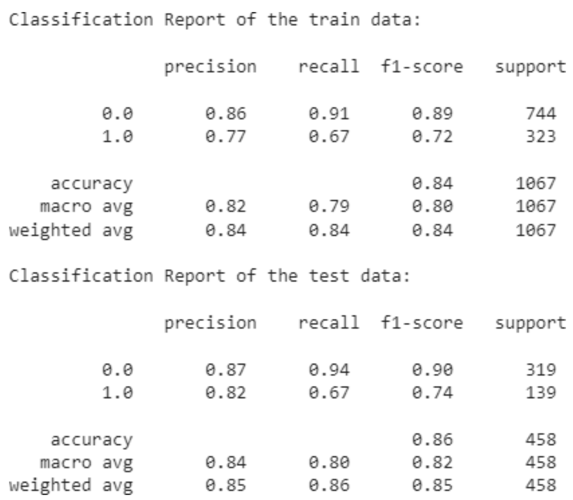
```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.87      0.88      0.87       744
         1.0       0.71      0.69      0.70       323

    accuracy                           0.82      1067
   macro avg       0.79      0.78      0.78      1067
weighted avg       0.82      0.82      0.82      1067

Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.89      0.92      0.90       319
         1.0       0.79      0.73      0.76       139

    accuracy                           0.86       458
   macro avg       0.84      0.82      0.83       458
weighted avg       0.86      0.86      0.86       458
```

For all these 4 models the feature importance remains the same as it was discussed above also when we didn’t tuned them.

5. **Ada Boosting**-

   Let’s try how Ada boost will work with this dataset-

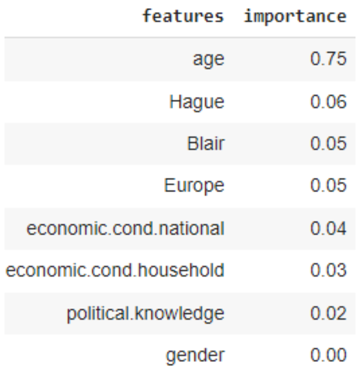
   With Ada boosting using n-estimators as 100 ,we get-



```
Classification Report of the train data:

              precision    recall  f1-score   support

         0.0       0.86      0.91      0.89       744
         1.0       0.77      0.67      0.72       323

    accuracy                           0.84      1067
   macro avg       0.82      0.79      0.80      1067
weighted avg       0.84      0.84      0.84      1067

Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.87      0.94      0.90       319
         1.0       0.82      0.67      0.74       139

    accuracy                           0.86       458
   macro avg       0.84      0.80      0.82       458
weighted avg       0.85      0.86      0.85       458
```

For Ada boosting too we get train and test accuracies as 84% and 86% respectively. Again this model is under-fit and we have to try other models too.

Feature importance that we get from this model is –

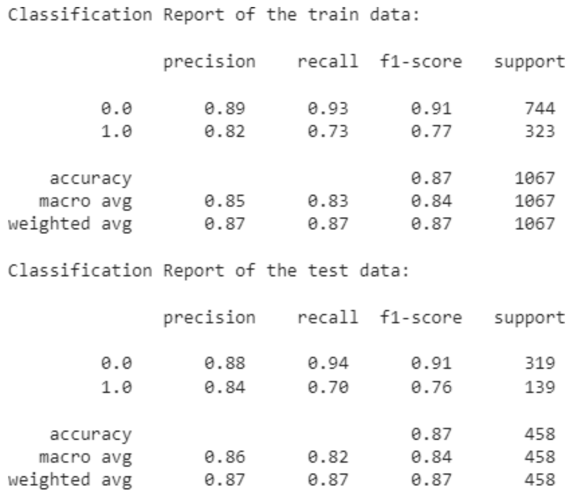


| features | importance |
|---:|---:|
| age | 0.75 |
| Hague | 0.06 |
| Blair | 0.05 |
| Europe | 0.05 |
| economic.cond.national | 0.04 |
| economic.cond.household | 0.03 |
| political.knowledge | 0.02 |
| gender | 0.00 |

Age is the biggest feature that is being used by Ada Boost.

### 6. **Gradient Boosting**-

With gradient boosting we are getting-

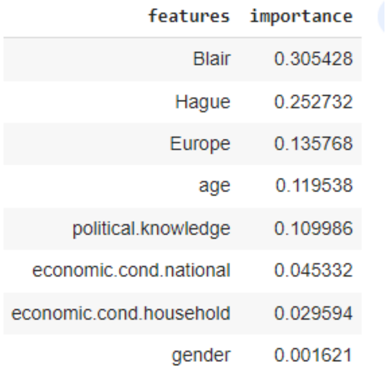


```
Classification Report of the train data:

              precision    recall  f1-score   support

         0.0       0.89      0.93      0.91       744
         1.0       0.82      0.73      0.77       323

    accuracy                           0.87      1067
   macro avg       0.85      0.83      0.84      1067
weighted avg       0.87      0.87      0.87      1067

Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.88      0.94      0.91       319
         1.0       0.84      0.70      0.76       139

    accuracy                           0.87       458
   macro avg       0.86      0.82      0.84       458
weighted avg       0.87      0.87      0.87       458
```

With gradient boosting we are getting the highest training and testing accuracy out of all tried models, we are getting accuracy of 87% for both training and testing.

There is no over or under-fitting in this model.

Let's check its feature importance-



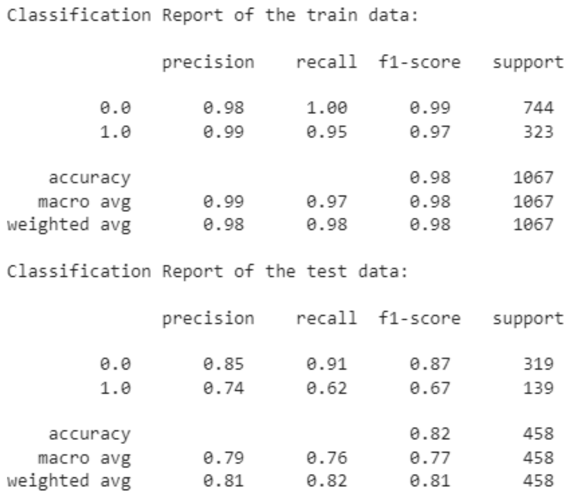
| features | importance |
|---|---|
| Blair | 0.305428 |
| Hague | 0.252732 |
| Europe | 0.135768 |
| age | 0.119538 |
| political.knowledge | 0.109986 |
| economic.cond.national | 0.045332 |
| economic.cond.household | 0.029594 |
| gender | 0.001621 |

Like other models Blair, Hague, Europe and age are the top 4 features that helped gradient boosting in achieving this accuracy.

7. **Bagging-**

With bagging we are getting-

We yet got a highly over-fit model



```
Classification Report of the train data:

              precision    recall  f1-score   support

         0.0       0.98      1.00      0.99       744
         1.0       0.99      0.95      0.97       323

    accuracy                           0.98      1067
   macro avg       0.99      0.97      0.98      1067
weighted avg       0.98      0.98      0.98      1067

Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.85      0.91      0.87       319
         1.0       0.74      0.62      0.67       139

    accuracy                           0.82       458
   macro avg       0.79      0.76      0.77       458
weighted avg       0.81      0.82      0.81       458
```

Training and testing accuracies are – 98% and 82% only.

So we are not going to use this model.

**1.7-Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model, classification report (4 pts) Final Model - Compare and comment on all models on the basis of the performance metrics in a structured tabular manner. Describe on which model is best/optimized, After comparison which model suits the best for the problem in hand on the basis of different measures. Comment on the final model.(3 pts)**

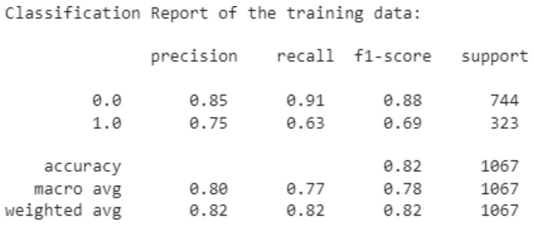
As the performance of Logistic regression, LDA, Naïve Bayes, KNN are giving same results even after tuning the parameters so we will just make use of basic model for checking accuracies, confusion matric , ROC curve and ROC_AUC score.

After that we will plot things for ADA boosting, Gradient boosting and Bagging.

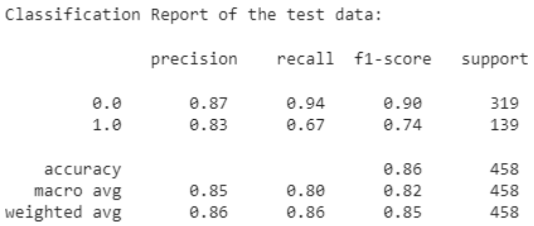
From these 7 combinations we will try to figure out the best one –

Logistic regression and LDA -

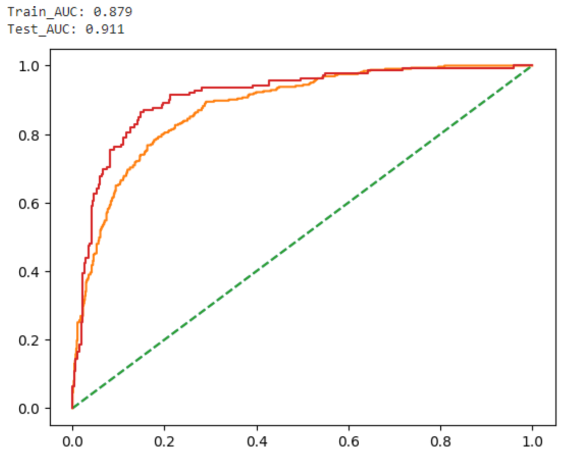
### Logistic regression-



```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.85      0.91      0.88       744
         1.0       0.75      0.63      0.69       323

    accuracy                           0.82      1067
   macro avg       0.80      0.77      0.78      1067
weighted avg       0.82      0.82      0.82      1067
```

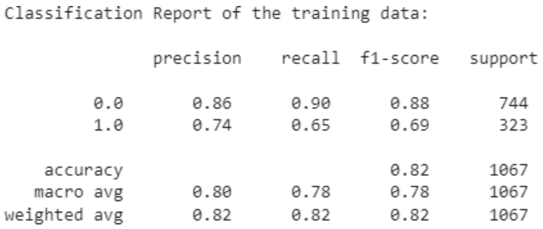


```
Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.87      0.94      0.90       319
         1.0       0.83      0.67      0.74       139

    accuracy                           0.86       458
   macro avg       0.85      0.80      0.82       458
weighted avg       0.86      0.86      0.85       458
```

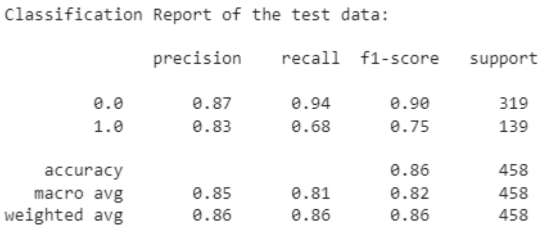


```
Train_AUC: 0.879
Test_AUC: 0.911
```

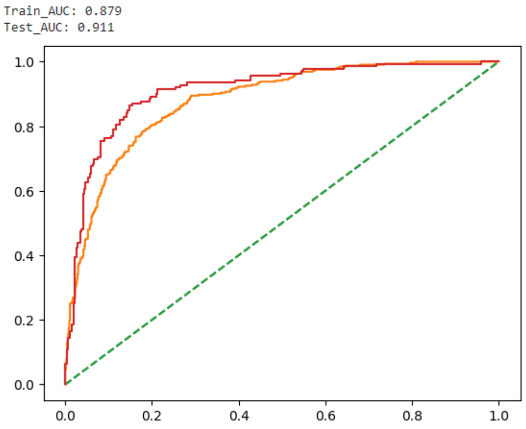
### LDA



```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.86      0.90      0.88       744
         1.0       0.74      0.65      0.69       323

    accuracy                           0.82      1067
   macro avg       0.80      0.78      0.78      1067
weighted avg       0.82      0.82      0.82      1067
```



```
Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.87      0.94      0.90       319
         1.0       0.83      0.68      0.75       139

    accuracy                           0.86       458
   macro avg       0.85      0.81      0.82       458
weighted avg       0.86      0.86      0.86       458
```



```
Train_AUC: 0.879
Test_AUC: 0.911
```

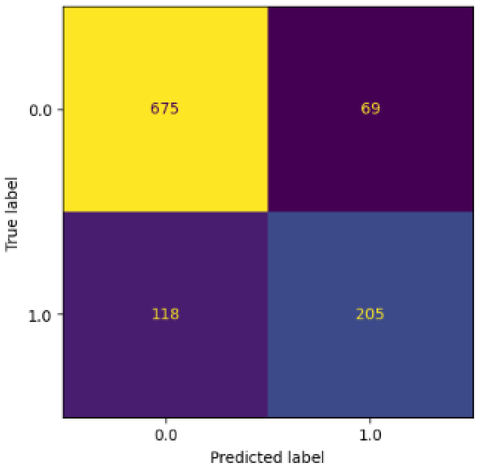
| **Model** | **Train accuracy** | **Test accuracy** |
|---|---|---|
| **Logistic** | 82 | 86 |
| **LDA** | 82 | 86 |

It could clearly be seen that both training and testing accuracy for both the models is 82% and 86% respectively.
This model is under-fit; it is fully capturing the patterns available in training data.

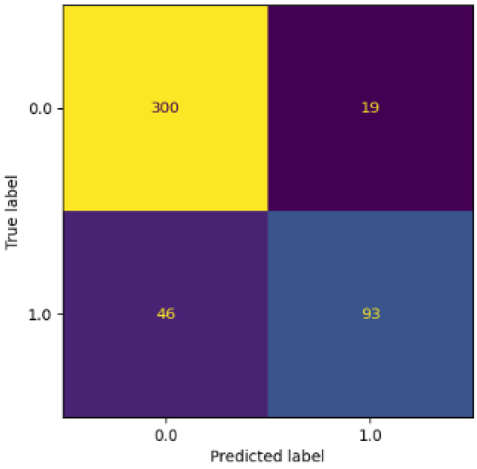
AUC score for both models is also same, and curves are identical too.

Let’s check there confusion matrix-

Logistic regression

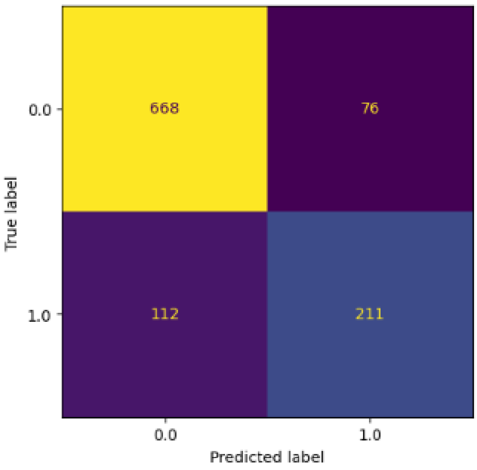


| True label \ Predicted label | 0.0 | 1.0 |
|---|---|---|
| 0.0 | 675 | 69 |
| 1.0 | 118 | 205 |

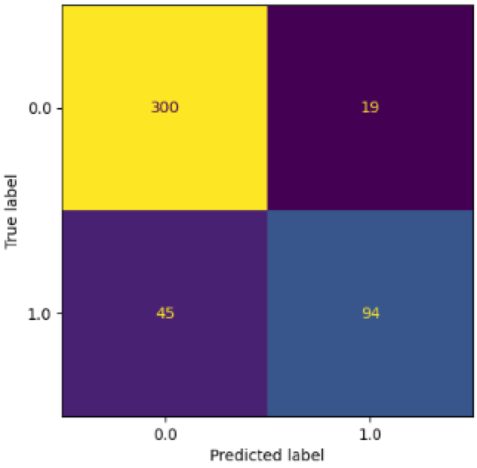


| True label \ Predicted label | 0.0 | 1.0 |
|---|---|---|
| 0.0 | 300 | 19 |
| 1.0 | 46 | 93 |

LDA



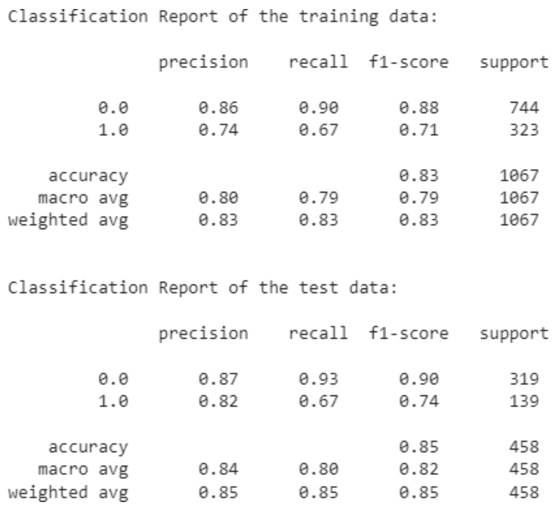
| True label \ Predicted label | 0.0 | 1.0 |
|---|---|---|
| 0.0 | 668 | 76 |
| 1.0 | 112 | 211 |



| True label \ Predicted label | 0.0 | 1.0 |
|---|---|---|
| 0.0 | 300 | 19 |
| 1.0 | 45 | 94 |

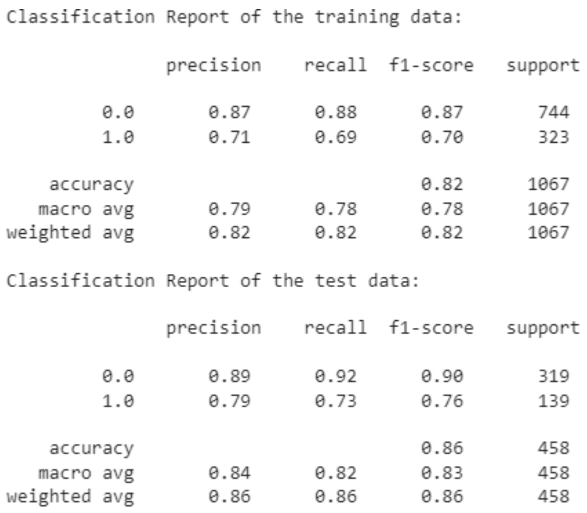
**KNN and Naïve Bayes-**

### KNN



```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.86      0.90      0.88       744
         1.0       0.74      0.67      0.71       323

    accuracy                           0.83      1067
   macro avg       0.80      0.79      0.79      1067
weighted avg       0.83      0.83      0.83      1067

Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.87      0.93      0.90       319
         1.0       0.82      0.67      0.74       139

    accuracy                           0.85       458
   macro avg       0.84      0.80      0.82       458
weighted avg       0.85      0.85      0.85       458
```

### Naïve Bayes



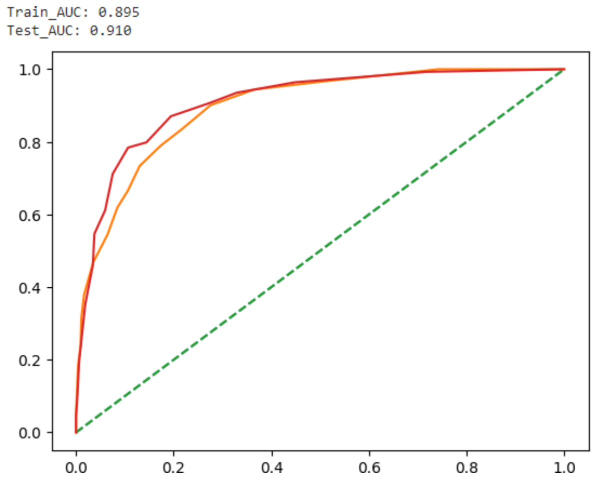
```
Classification Report of the training data:

              precision    recall  f1-score   support

         0.0       0.87      0.88      0.87       744
         1.0       0.71      0.69      0.70       323

    accuracy                           0.82      1067
   macro avg       0.79      0.78      0.78      1067
weighted avg       0.82      0.82      0.82      1067
Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.89      0.92      0.90       319
         1.0       0.79      0.73      0.76       139

    accuracy                           0.86       458
   macro avg       0.84      0.82      0.83       458
weighted avg       0.86      0.86      0.86       458
```

For both the models the train and test accuracies are-

| **Model** | **Train accuracy** | **Test accuracy** |
|---|---|---|
| **KNN** | 83 | 85 |
| **Naïve Bayes** | 82 | 86 |

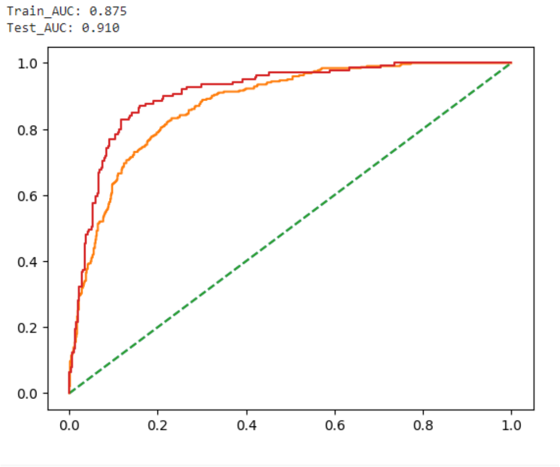
Both these models are under-fit, again the test accuracy is greater then train accuracy.

KNN



```
Train_AUC: 0.895
Test_AUC: 0.910
```

Naïve Bayes



```
Train_AUC: 0.875
Test_AUC: 0.910
```

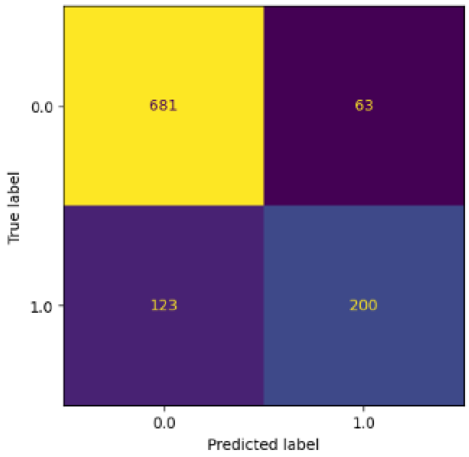
Area under the curve is better for KNN than Naïve Bayes and other 2 above algorithms too (Logistic regression and LDA)

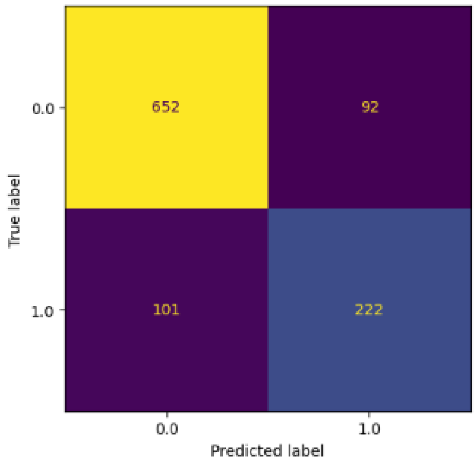
AUC is closer for both train and test set in KNN. It again show a slight under-fitting but it is a better model than the other 3 (Logistic, LDA and NB)

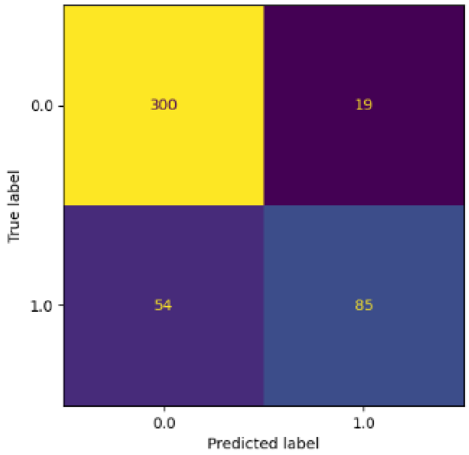
Let’s plot confusion matrix for both of them-

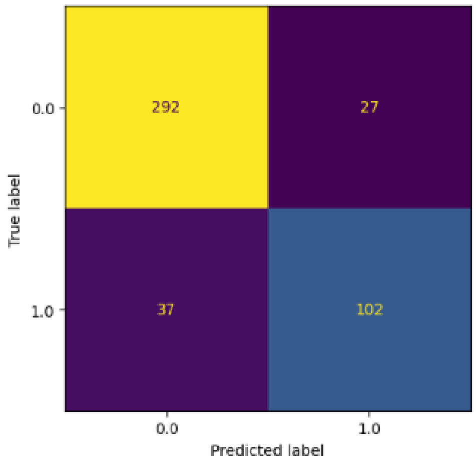
KNN

Naïve Bayes





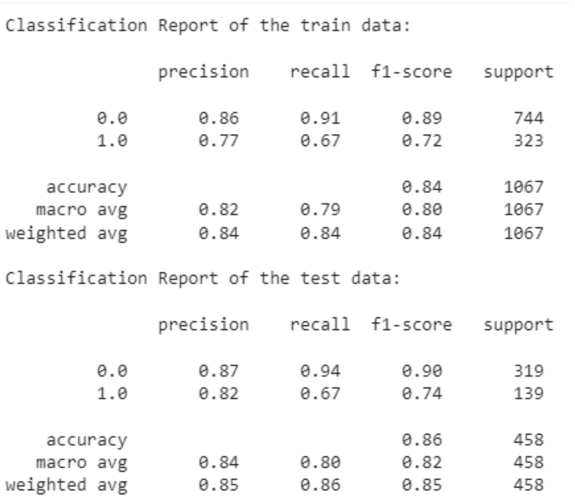




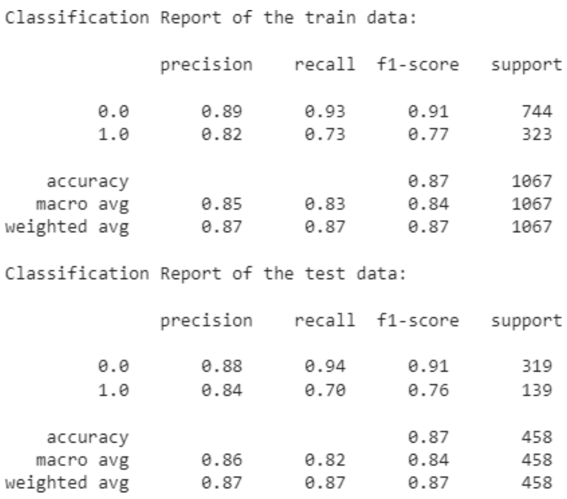
ADA boosting and Gradient Boosting-

ADA boosting

Gradient Boosting



```
Classification Report of the train data:

              precision    recall  f1-score   support

         0.0       0.86      0.91      0.89       744
         1.0       0.77      0.67      0.72       323

    accuracy                           0.84      1067
   macro avg       0.82      0.79      0.80      1067
weighted avg       0.84      0.84      0.84      1067

Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.87      0.94      0.90       319
         1.0       0.82      0.67      0.74       139

    accuracy                           0.86       458
   macro avg       0.84      0.80      0.82       458
weighted avg       0.85      0.86      0.85       458
```



```
Classification Report of the train data:

              precision    recall  f1-score   support

         0.0       0.89      0.93      0.91       744
         1.0       0.82      0.73      0.77       323

    accuracy                           0.87      1067
   macro avg       0.85      0.83      0.84      1067
weighted avg       0.87      0.87      0.87      1067

Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.88      0.94      0.91       319
         1.0       0.84      0.70      0.76       139

    accuracy                           0.87       458
   macro avg       0.86      0.82      0.84       458
weighted avg       0.87      0.87      0.87       458
```

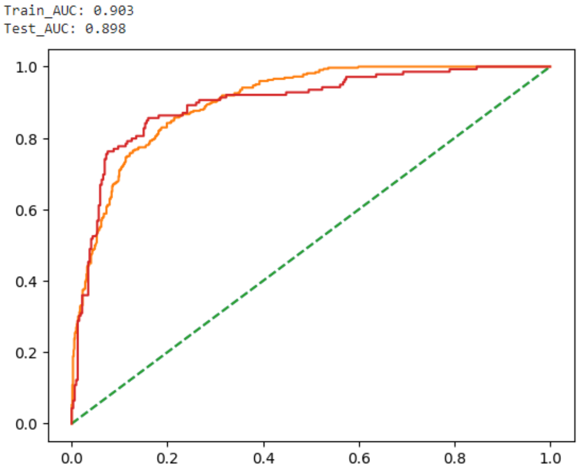
For both the models the train and test accuracies are-

| Model | Train accuracy | Test accuracy |
|---|---|---|
| **ADA Boosting** | 84 | 86 |
| **Gradient Boosting** | 87 | 87 |

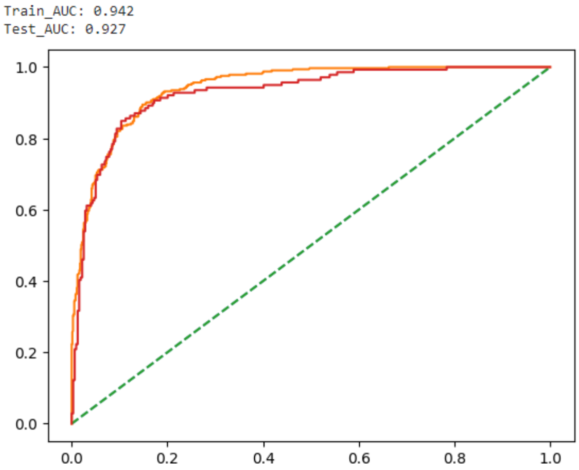
Ada Boosting is under-fit but there is still increase in overall accuracy from previous models. Gradient boosting is giving us the best accuracy out of all the models, and there is no over or under-fitting at all.

Let’s take a look at AUC score-

ADA boosting



```
Train_AUC: 0.903
Test_AUC: 0.898
```

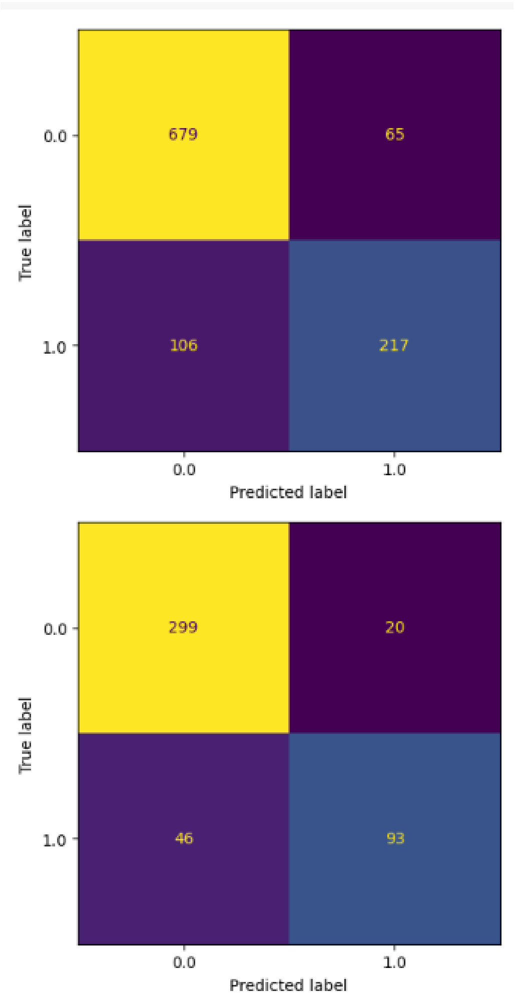


```
Train_AUC: 0.942
Test_AUC: 0.927
```

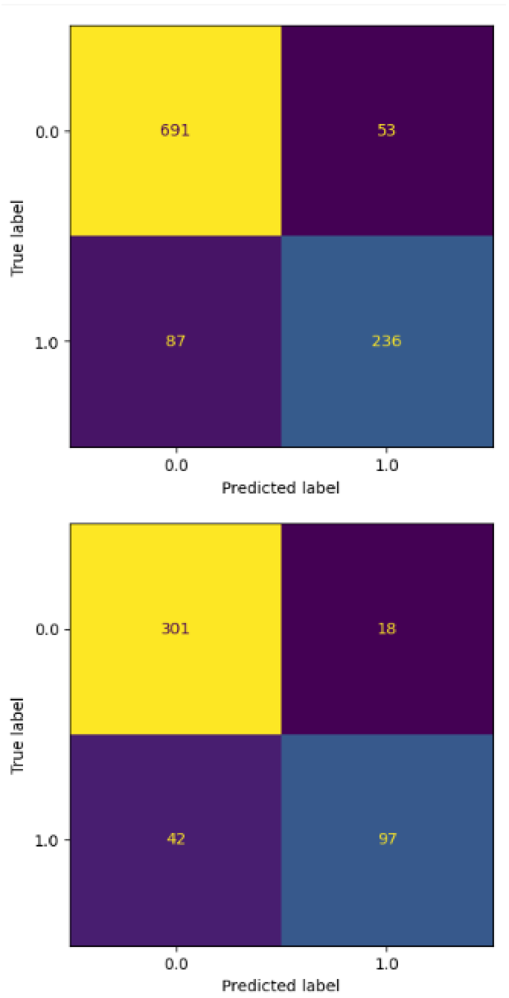
Ada boosting ROC score is higher than previous models but it’s below what’s being covered by gradient boosting.

Let’s check the confusion matrix for both of them-

ADA boosting

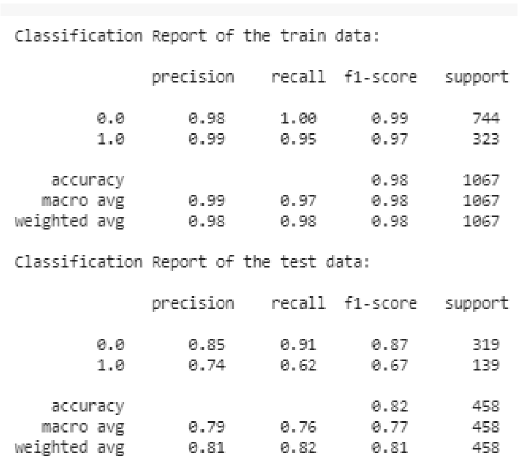


Gradient Boosting



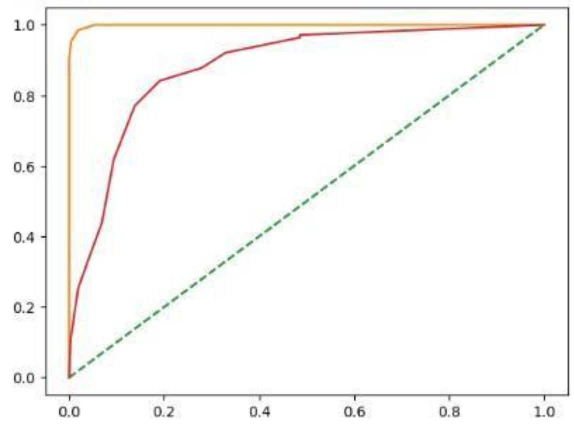
**Bagging-**

**Let’s check with bagging-**



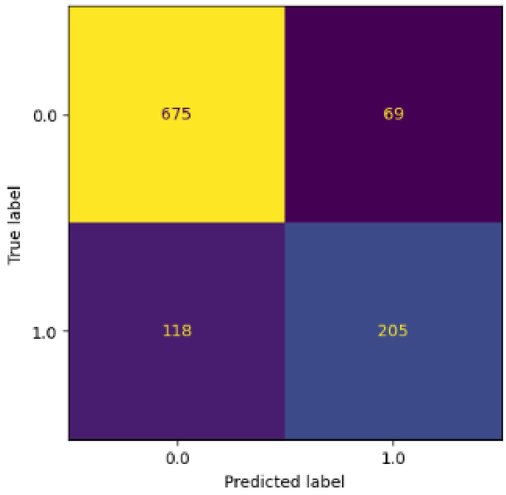
```
Classification Report of the train data:

              precision    recall  f1-score   support

         0.0       0.98      1.00      0.99       744
         1.0       0.99      0.95      0.97       323

    accuracy                           0.98      1067
   macro avg       0.99      0.97      0.98      1067
weighted avg       0.98      0.98      0.98      1067

Classification Report of the test data:

              precision    recall  f1-score   support

         0.0       0.85      0.91      0.87       319
         1.0       0.74      0.62      0.67       139

    accuracy                           0.82       458
   macro avg       0.79      0.76      0.77       458
weighted avg       0.81      0.82      0.81       458
```

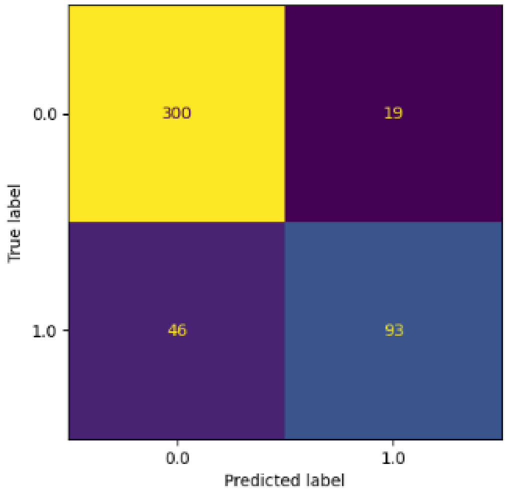
```
Train_AUC: 0.999
Test_AUC: 0.882
```



For bagging, model is totally over fitted, it is quite visible from AUC score too.

Let’s check the matrix-





Let’s make a table to figure out the best Model-

| Model | Train accuracy | Test accuracy | Comment |
|---|---|---|---|
| **Logistic** | 82 | 86 | Under-fit |
| **LDA** | 82 | 86 | Under-fit |
| **KNN** | 83 | 85 | Under-fit |
| **Naïve Bayes** | 82 | 86 | Under-fit |
| **ADA Boosting** | 84 | 86 | Under-fit |
| **Gradient Boosting** | **87** | **87** | **Best** |
| **Bagging** | 98 | 82 | Over-fit |

Gradient boosting is the best model as accuracy is highest for test set as well as training set and no over or under fitting is these which is quite evidently seen in other models.

### 1.8-Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective. There should be at least 3-4 Recommendations and insights in total. Recommendations should be easily understandable and business specific, students should not give any technical suggestions.

To get the actionable insights let's check out the most important features from all the algorithm's

From all the mentioned feature importance from different models we figured out that-

Blair, Hague and Europe - These are the most important features-

Let's pull some actionable insights from these features-

**Recommendations-**

**Blair-**

1. Respondents who rated Tony Blair positively (i.e. a rating of 4 or 5) were more likely to vote for the Labour party. Conversely, those who rated him negatively (i.e. a rating of 1 or 2) were more likely to vote for the Conservative party.

Labour party can work on increasing the popularity and image of Blair in a better way so as to further increase vote share.

**Hague-**

1. As the rating of the Hague increased, the percentage of votes going to the Labour party decreased, while the percentage of votes going to the Conservative party increased.
2. Furthermore, the Conservative party was more successful in retaining a higher vote share among voters who gave the highest rating of 5 to The Hague.

Conservative party can work on increasing the popularity and image of Blair in a better way, so that most of the people rate him highest so as to further increase vote share.

**Europe-**

1. At the lower end of the scale (scores 1-4), both parties receive a relatively low percentage of votes, with the Conservatives receiving less than 15% of the vote and Labour receiving around 85-90% of the vote. As Eurosceptic sentiment increases (scores 5-11), the Conservative vote share increases steadily, with the party receiving over 50% of the vote at the highest levels of Euroscepticism. In contrast, the Labour vote share decreases as Eurosceptic sentiment increases, with the party receiving less than 50% of the vote at the highest levels of Euroscepticism.

Overall, this suggests that Eurosceptic sentiment is associated with higher levels of support for the Conservative party and lower levels of support for the Labour party.