Business Report

Machine Learning

Ayush Sharma

Table of Contents

G. 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.	16

A. **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.**

Ans: Following inferences can be made about the dataset:

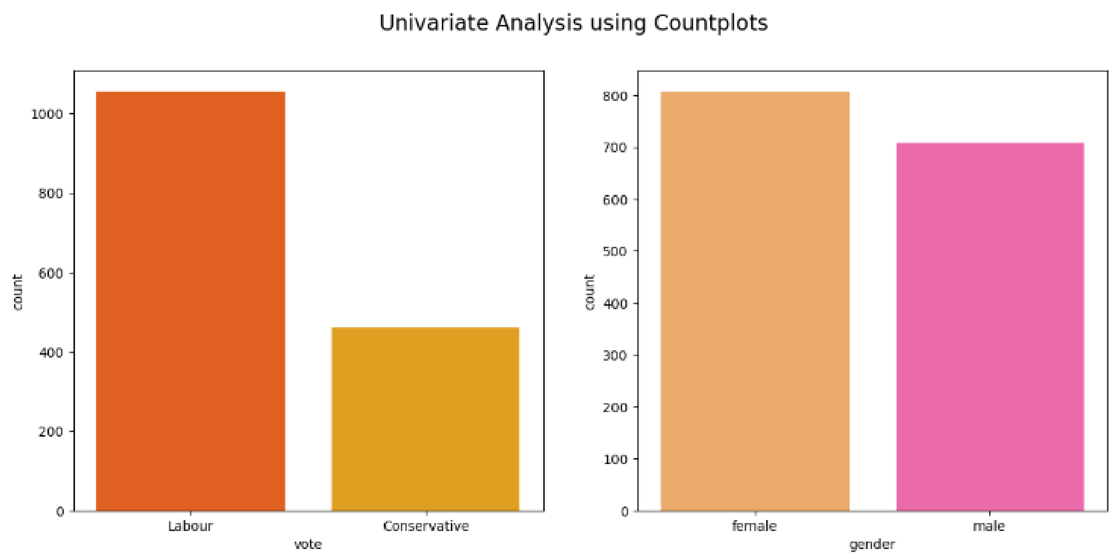
- The data consists of 1525 rows and 10 columns
- The data consists of 2 categorical and 8 continuous variables
- The data consists of 2 object type and 8 integer type columns
- The 1st column represents the "S. No" hence it has been dropped
- No null values exist in the dataset
- 8 duplicate values were found and were dropped from the dataset
- Except the age column, all the other numeric columns have certain fixed levels therefore they can also be treated as categorical variables
- It can also be observed that the target/dependent variable "vote" seems to be unevenly distributed as the number of votes for the "Conservative" level are approximately half when compared to its counterpart "Labour" level
- This is an indication of an unbalanced dependent variable which means that over-sampling techniques such as SMOTE can be used when building the ML models to compensate for this balance

```
Categorical levels:

vote:
Labour          1057
Conservative     460
Name: vote, dtype: int64

gender:
female    808
male      709
Name: gender, dtype: int64

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
```

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

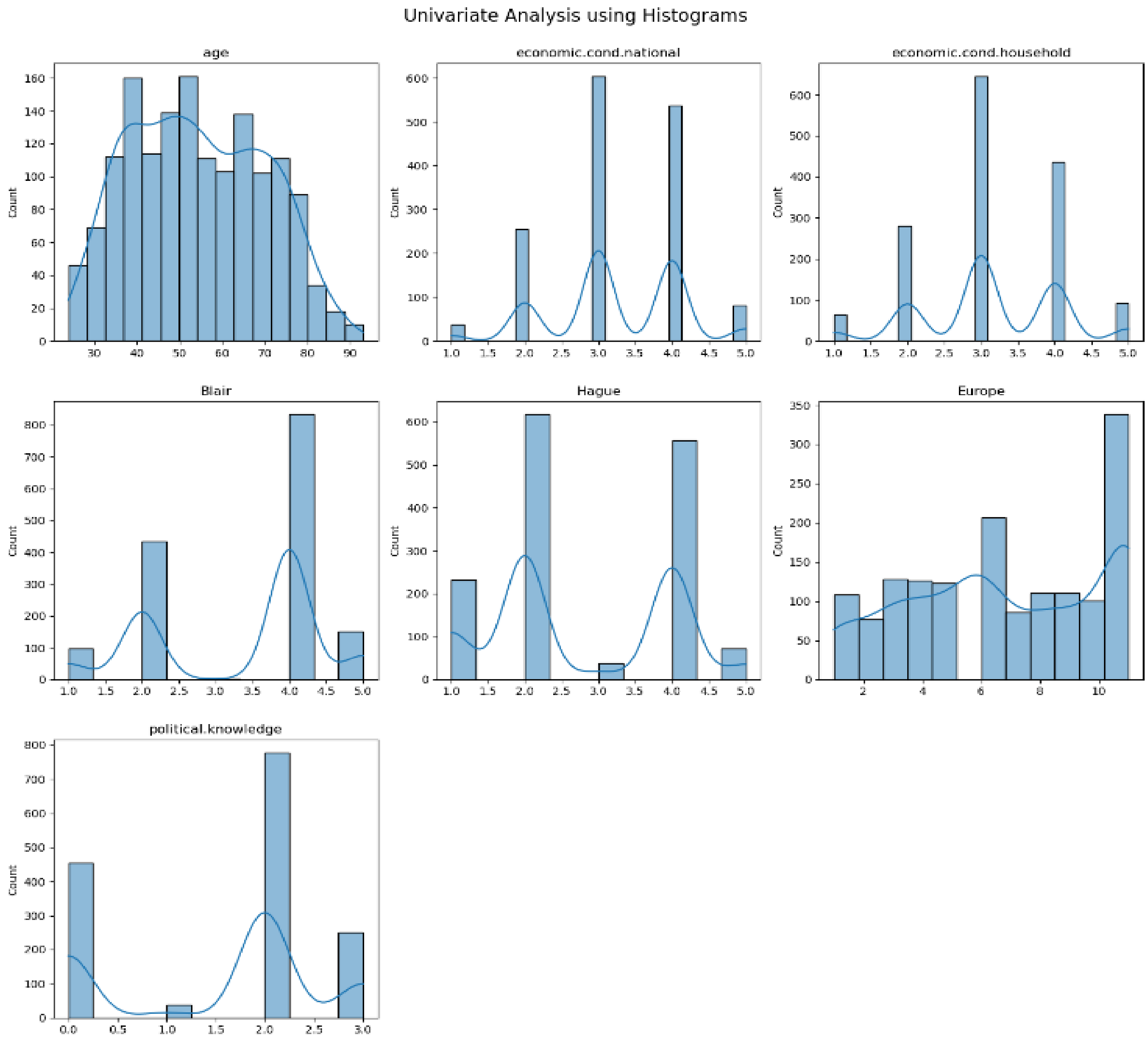
B. Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also check for outliers (4 pts). Interpret the inferences for each (3 pts) Distribution plots(histogram) or similar plots for the continuous columns. Box plots. Appropriate plots for categorical variables. Inferences on each plot. Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.

Ans: **Univariate Analysis**



Inferences for the Categorical variables:

- Vote variable
  - As discussed previously, it can be clearly observed that the number of votes for the Labour party is approximately double than that of the Conservative party
  - The number of votes for the Labour part is about 1060 and that of the Conservative party is 450.
- Gender variable
  - The male to female ratio is approximately equal with about 800 females and 700 males

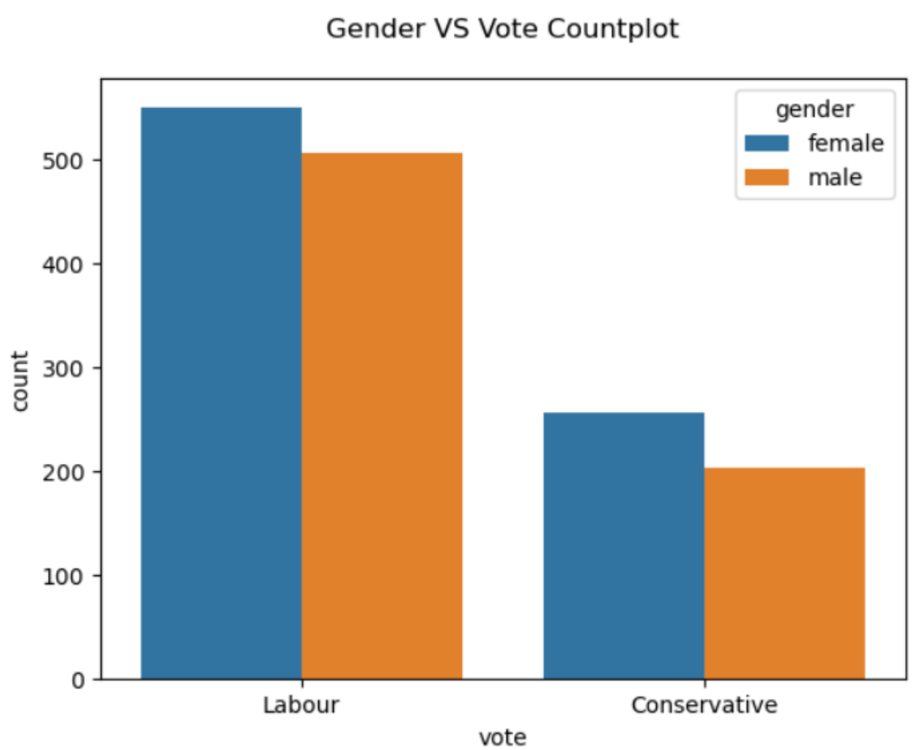


Inferences for the Continuous variables:

- Age variable
    - The age of the people ranges from 25-90 years
    - A majority of the people pertain to the age group of 35-55 years
    - The number of voters pertaining to the age group of 60-80 follows after this
- Economic Condition National and Household
    - A similar trend can be observed in both these variables with the categories 3 and 4 accounting for the highest numbers of people
    - The categories 1 and 5 constitute for the lowest numbers of people
- Blair
    - Approximately 800 people have given a rating of 4 to Blair; the leader of the Labour party followed by 2 (400 people) and 5 (200 people) respectively
- Hague

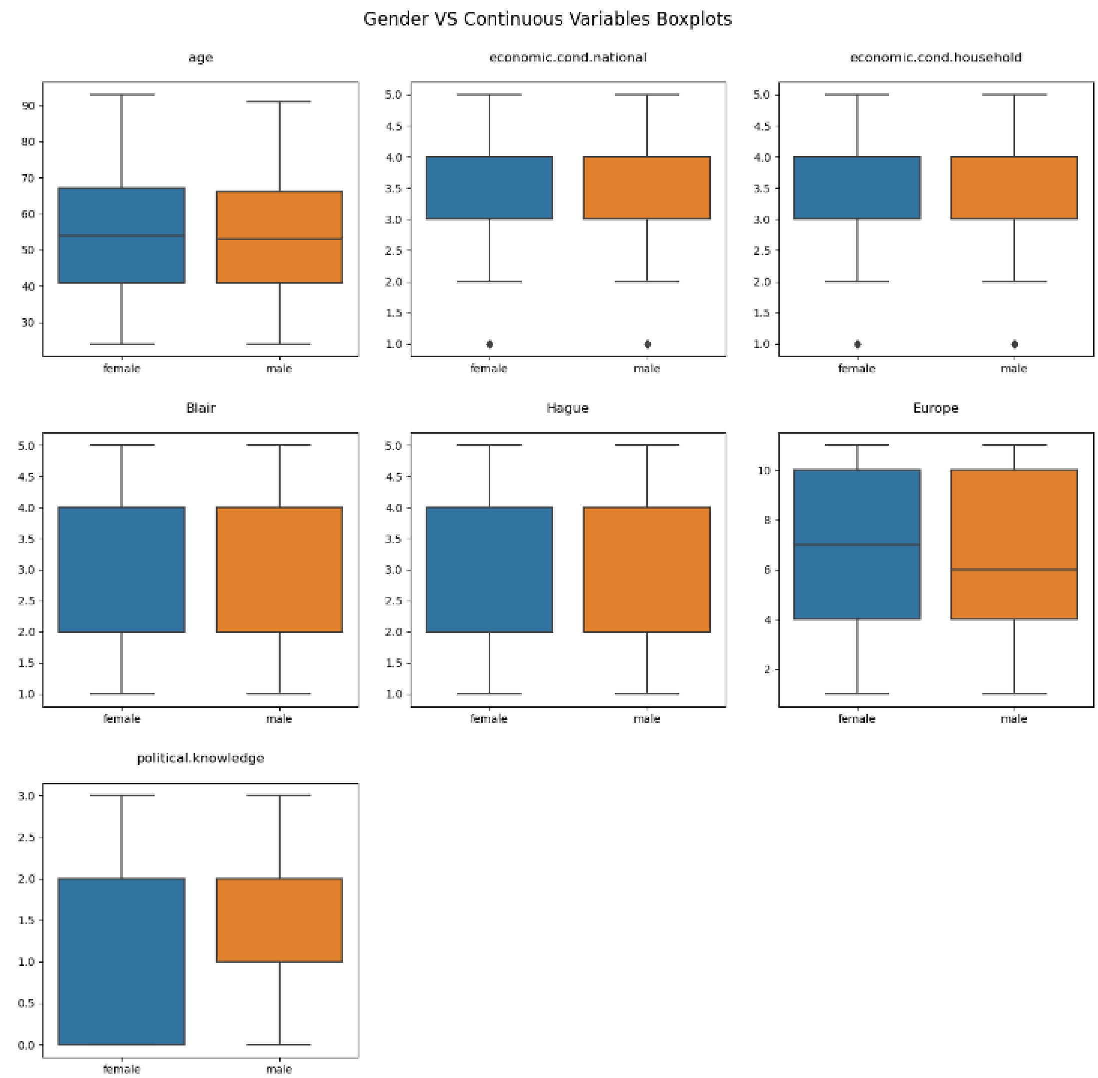
    - Approximately 600 people have given a rating of 2 to Hague; the leader of the Conservative party followed by 4 (550 people) and 1 (200 people) respectively
- Europe
    - Approximately 350 people have given a rating of 11 indicating their sentiment to be “Eurosceptic”.
    - This is followed by approximately 200 people who have given a rating of 6 indicating a neutral attitude towards the European integration

Bivariate Analysis



Inferences for the Categorical/Categorical variables:

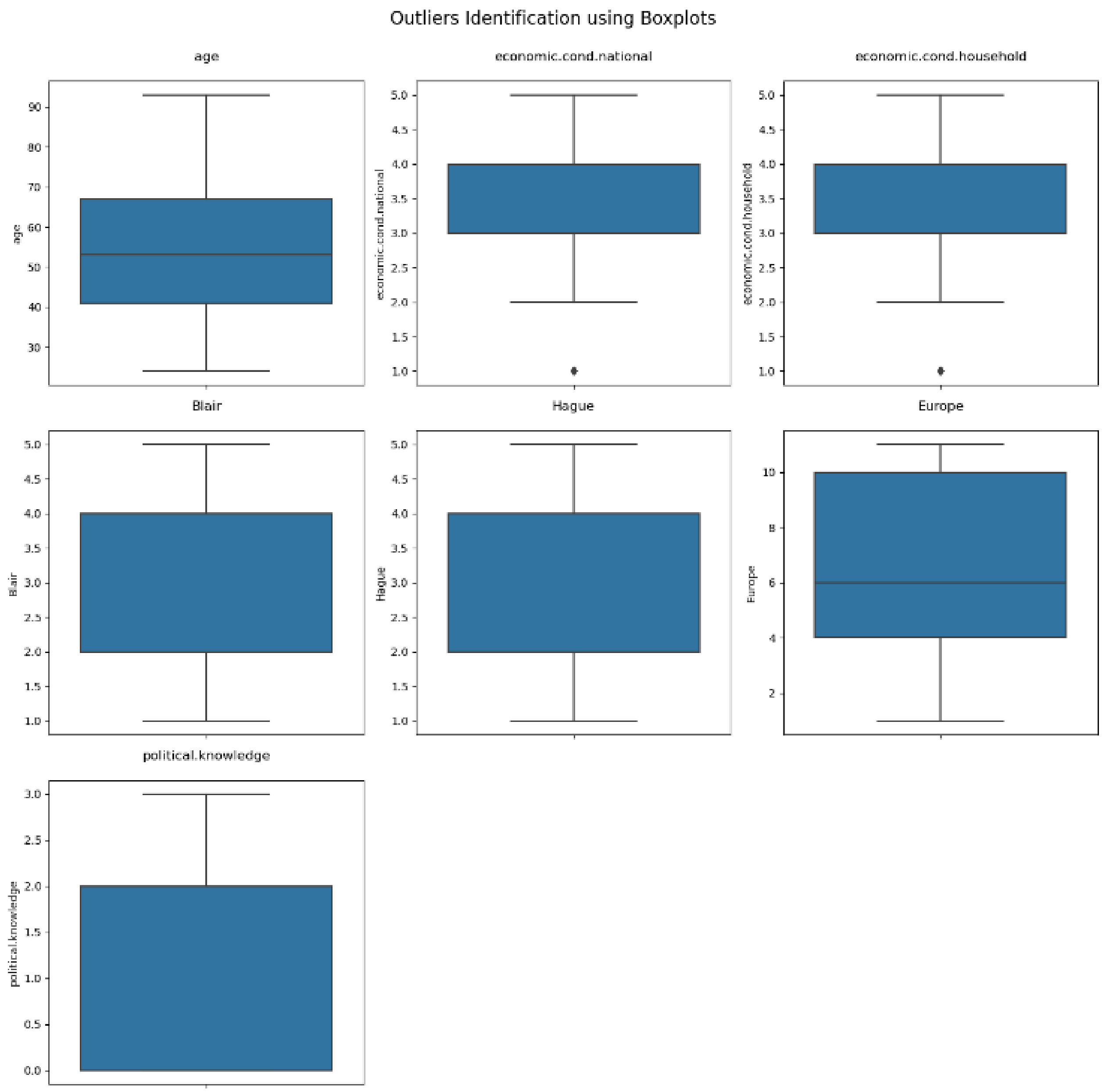
- The number of female voters is higher in both the voting categories with approximately 550 and 250 female votes belonging to the Labour and Conservative parties respectively
- Approximately 500 and 200 male votes belong to the Labour and Conservative parties respectively



Inferences for the Categoric/Numeric variables:

- Gender Categoric variable
  - The majority of the male and female voters pertain to the age group of 50-75 years

Outliers Treatment



- It can be seen that there aren't many outliers in our dataset hence outlier treatment is not required for this dataset

C. 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).

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. (pd.categorical().codes(), pd.get_dummies(drop_first=True)) Data split, ratio defined for the split, train-test split should be discussed.

Ans: Scaling is not required here as except the age variable; all the other variables have similar values. Scaling is required as certain models rely on distance calculation as their algorithms such as Logistic Regression and KNN. However, in our dataset, all the columns already follow certain fixed values hence scaling would be redundant.

Dataset after Encoding

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male	vote_Labour
0	43	3	3	4	1	2	2	0	1
1	36	4	4	4	4	5	2	1	1
2	35	4	4	5	2	3	2	1	1
3	24	4	2	2	1	4	0	0	1
4	41	2	2	1	1	6	2	1	1

```
Training data shape:
(1061, 8)       (1061,)

Testing data shape:
(456, 8)        (456,)
```

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

Ans:

```
Model Summary for Logistic Regression Model:

Train Data:

Model Score: 0.841
Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.67      0.72       319
           1       0.87      0.91      0.89       742

    accuracy                           0.84      1061
   macro avg       0.82      0.79      0.80      1061
weighted avg       0.84      0.84      0.84      1061


Test Data:

Model Score: 0.818
Classification Report
              precision    recall  f1-score   support

           0       0.75      0.61      0.67       141
           1       0.84      0.91      0.87       315

    accuracy                           0.82       456
   macro avg       0.80      0.76      0.77       456
weighted avg       0.81      0.82      0.81       456
```

```
Model Summary for LDA Model:

Train Data:

Model Score: 0.839
Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.69      0.72       319
           1       0.87      0.90      0.89       742

    accuracy                           0.84      1061
   macro avg       0.81      0.80      0.80      1061
weighted avg       0.84      0.84      0.84      1061


Test Data:

Model Score: 0.825
Classification Report
              precision    recall  f1-score   support

           0       0.75      0.65      0.69       141
           1       0.85      0.90      0.88       315

    accuracy                           0.82       456
   macro avg       0.80      0.78      0.79       456
weighted avg       0.82      0.82      0.82       456
```

- The Logistic Regression model has been built with the "liblinear" solver as it is appropriate for small datasets and is efficient when handling a one-versus-rest approach
- The model scores and classification report metrics for both the models are close to each other.
- Both the models have their accuracy at 84% for the train data and 82% for the test data respectively which seems to be decent enough.
- The fact that the test data also shows similar accuracy means that the models are not under or over fits.
- For the test data, the precision values of the Logistic Regression model are at 75% and 84% (0/1) and the recall values are at 61% and 91% (0/1) respectively.

- For the test data, the precision values of the LDA model are at 75% and 85% (0/1) and the recall values are at 65% and 90% (0/1) respectively.
- In comparison, the LDA model performs better in predicting the recall values for the test data
- These values are comparatively low which means there is still scope of building a better model by using model tuning techniques such as grid search, passing hyperparameters etc.

E. **Apply KNN Model and Naïve Bayes Model. Interpret the inferences of each model. 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)**

Ans:

```
Model Summary for KNN Model (weights=uniform):

Train Data:

Model Score: 0.866
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.73      0.77       319
           1       0.89      0.93      0.91       742

    accuracy                           0.87      1061
   macro avg       0.85      0.83      0.84      1061
weighted avg       0.86      0.87      0.86      1061


Test Data:

Model Score: 0.772
Classification Report
              precision    recall  f1-score   support

           0       0.66      0.55      0.60       141
           1       0.81      0.87      0.84       315

    accuracy                           0.77       456
   macro avg       0.73      0.71      0.72       456
weighted avg       0.76      0.77      0.77       456
```

```
Model Summary for KNN Model (weights=distance):

Train Data:

Model Score: 0.999
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       319
           1       1.00      1.00      1.00       742

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


Test Data:

Model Score: 0.768
Classification Report
              precision    recall  f1-score   support

           0       0.65      0.55      0.59       141
           1       0.81      0.87      0.84       315

    accuracy                           0.77       456
   macro avg       0.73      0.71      0.71       456
weighted avg       0.76      0.77      0.76       456
```

- Two variants of the KNN model have been built with “uniform” and “distance” as the hyperparameters of weights.
- The “distance” hyperparameter assigns greater influence of the neighbours closer to a point than the ones further away. The “uniform” hyperparameter on the other hand assigns all the neighbours a uniform weight.
- It is evident from the classification report that the KNN model with the “distance” hyperparameter is an overfit model as it performs exceptionally well on the train data while it fails to do the same on the test data
- The KNN model with the “uniform” hyperparameter also seems to be a weak model due to its low precision and recall values

- We’ve observed previously that our dependent variable (vote) is an unbalance in nature and since KNN model is a distance-based algorithm, we can proceed by employing SMOTE in order to improve our model’s efficiency

```
Model Summary for the Naive Bayes model:

Train Data:

Model Score: 0.841
Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.72      0.73       319
           1       0.88      0.89      0.89       742

    accuracy                           0.84      1061
   macro avg       0.81      0.81      0.81      1061
weighted avg       0.84      0.84      0.84      1061


Test Data:

Model Score: 0.816
Classification Report
              precision    recall  f1-score   support

           0       0.73      0.65      0.68       141
           1       0.85      0.89      0.87       315

    accuracy                           0.82       456
   macro avg       0.79      0.77      0.78       456
weighted avg       0.81      0.82      0.81       456
```

- The Naïve Bayes model performs comparatively well with an accuracy of 81% on the test dataset
- The recall and precision values for the test data for class 1 are pretty decent at 89% and 85% respectively.
- However, the recall and precision values for the test data for class 0 are at 73% and 65% respectively which are relatively low.

F. **Model Tuning, Bagging and Boosting. 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.**

Ans:

```
Model Summary for Logistic Regression Model after Grid Search:

Train Data:

Model Score: 0.82
Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.68      0.72       319
           1       0.87      0.92      0.89       742

    accuracy                           0.84      1061
   macro avg       0.82      0.80      0.81      1061
weighted avg       0.84      0.84      0.84      1061


Test Data:

Model Score: 0.818
Classification Report
              precision    recall  f1-score   support

           0       0.76      0.62      0.68       141
           1       0.84      0.91      0.88       315

    accuracy                           0.82       456
   macro avg       0.80      0.76      0.78       456
weighted avg       0.82      0.82      0.81       456
```

```
Model Summary for LDA Model after Grid Search:

Train Data:

Model Score: 0.839
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.69      0.72       319
           1       0.87      0.90      0.89       742

    accuracy                           0.84      1061
   macro avg       0.81      0.80      0.80      1061
weighted avg       0.84      0.84      0.84      1061


Test Data:

Model Score: 0.825
Classification Report
              precision    recall  f1-score   support

           0       0.75      0.65      0.69       141
           1       0.85      0.90      0.88       315

    accuracy                           0.82       456
   macro avg       0.80      0.78      0.79       456
weighted avg       0.82      0.82      0.82       456
```

- The model scores and classification report metrics for both the models doesn’t seem to have improved after using grid search.
- Both the models have their accuracy at 82% for the test data
- The precision and recall values have not increased for either of the models
- This indicates that perhaps the Logistic Regression and LDA models are not appropriate for this dataset

```
Model Summary for KNN Model after Grid Search:

Train Data:

Model Score: 0.866
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       319
           1       1.00      1.00      1.00       742

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


Test Data:

Model Score: 0.772
Classification Report
              precision    recall  f1-score   support

           0       0.57      0.70      0.63       141
           1       0.85      0.76      0.80       315

    accuracy                           0.74       456
   macro avg       0.71      0.73      0.72       456
weighted avg       0.76      0.74      0.75       456
```

```
Fitting 5 folds for each of 320 candidates, totalling 1600 fits
```

```
{'algorithm': 'auto',
 'leaf_size': 20,
 'metric': 'cosine',
 'n_neighbors': 7,
 'weights': 'uniform'}
```

- Despite using grid search and using the best parameters, it seems that the KNN model is still under-performing as the precision and recall values are very low
- Seems like KNN is also not an appropriate model for our dataset

G. **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.**

Ans:

Classification Reports for the Test Data for various models:

Naive Bayes:

	precision	recall	f1-score	support
0	0.67	0.70	0.69	141
1	0.86	0.84	0.85	315
accuracy			0.80	456
macro avg	0.77	0.77	0.77	456
weighted avg	0.80	0.80	0.80	456

Random Forest:

	precision	recall	f1-score	support
0	0.76	0.55	0.64	141
1	0.82	0.92	0.87	315
accuracy			0.81	456
macro avg	0.79	0.74	0.76	456
weighted avg	0.80	0.81	0.80	456

Bagging:

	precision	recall	f1-score	support
0	0.75	0.55	0.64	141
1	0.82	0.92	0.87	315
accuracy			0.80	456
macro avg	0.79	0.74	0.75	456
weighted avg	0.80	0.80	0.80	456

AdaBoosting:

	precision	recall	f1-score	support
0	0.73	0.64	0.68	141
1	0.85	0.89	0.87	315
accuracy			0.81	456
macro avg	0.79	0.77	0.77	456
weighted avg	0.81	0.81	0.81	456

Gradient Boosting:

	precision	recall	f1-score	support
0	0.75	0.67	0.70	141
1	0.86	0.90	0.88	315
accuracy			0.83	456
macro avg	0.80	0.78	0.79	456
weighted avg	0.82	0.83	0.82	456

- It can be noted that the models Naïve Bayes and AdaBoosting relatively perform well in terms of the precision and recall values of both the classes
- The other models tend to have lower values of precision and recall