MACHINE LEARNING

Project

Aisha Khan

Table of Contents

Table of Figures

List of Tables

Problem 1:

1. You are hired by one of the leading news channels CNBE who wants to analyze recent elections. This survey was conducted on 1525 voters with 9 variables. You have to build a model, to predict which party a voter will vote for on the basis of the given information, to create an exit poll that will help in predicting overall win and seats covered by a particular party.

Data set: Election Data

Data Dictionary:

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

Data Ingestion:

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

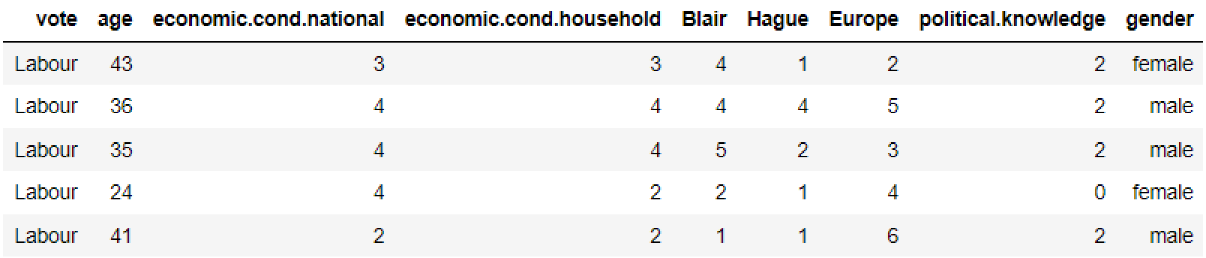
- The required packages were loaded.
- The data is loaded.
- After loading the dataset it is observed that one column in the dataset is unnamed and has no significance in model building, hence we drop the column before proceeding in the EDA phase.
- The Dataset has 1525 rows and 9 features.
- The data type of the variables are as follows:

Data Type	Count of Columns
int64	7
object	2
Grand Total	**9**

Table 2. Count of Each Data type

➢ Data Exploration was performed using the following functions:

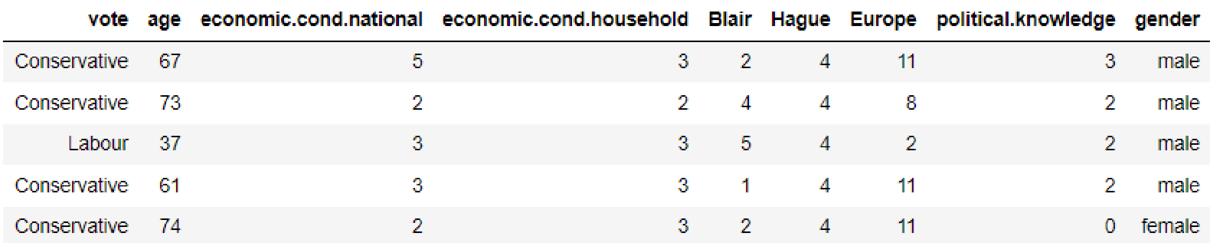
- Head



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

Table 3. First 5 rows of data

- Tail

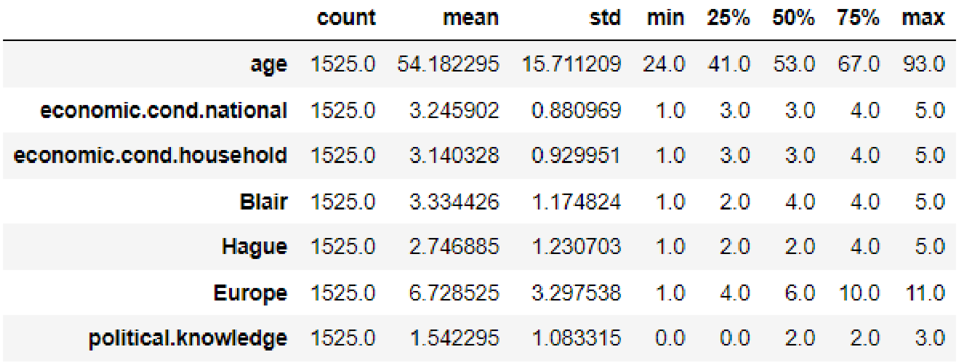


vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
Conservative	67	5	3	2	4	11	3	male
Conservative	73	2	2	4	4	8	2	male
Labour	37	3	3	5	4	2	2	male
Conservative	61	3	3	1	4	11	2	male
Conservative	74	2	3	2	4	11	0	female

Table 4. Last 5 Rows of Data

- Shape
  The dataset has 1525 rows and 9 variables (After removing the unnamed variable).

- Summary



	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

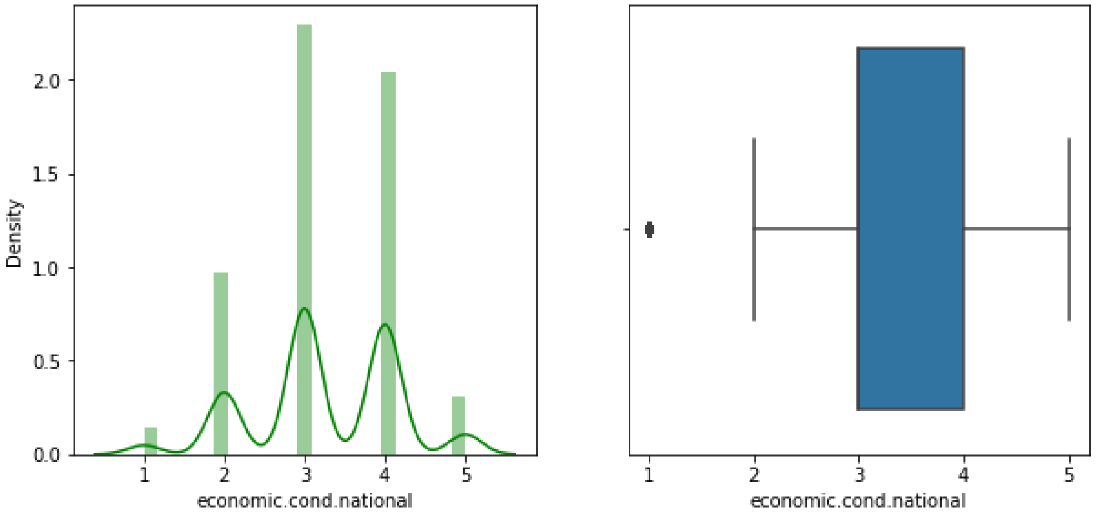
Table 5. Summary of the Data

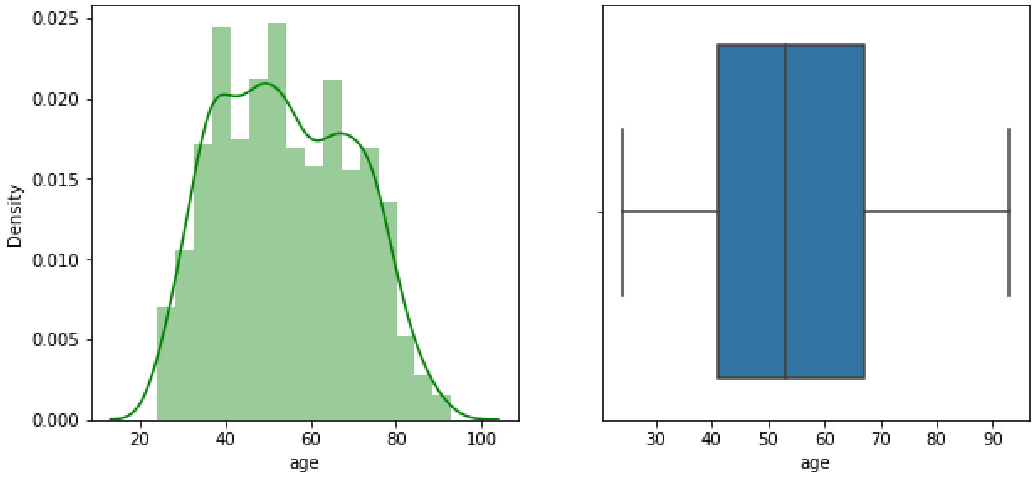
- Check Duplicates

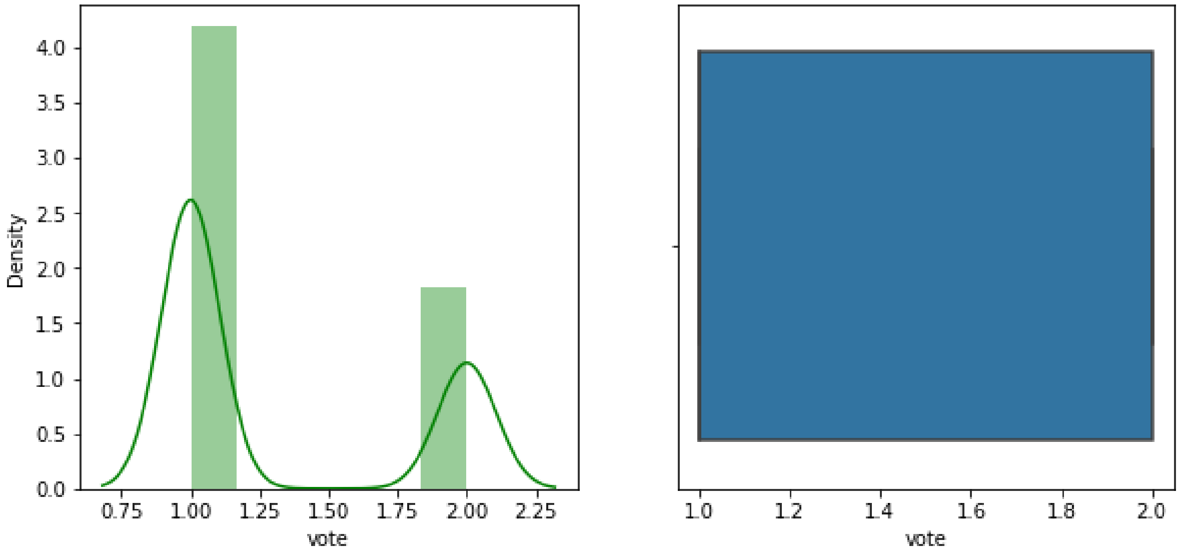
After checking duplicates, we found that there are 8 duplicates values which were removed from the dataset.

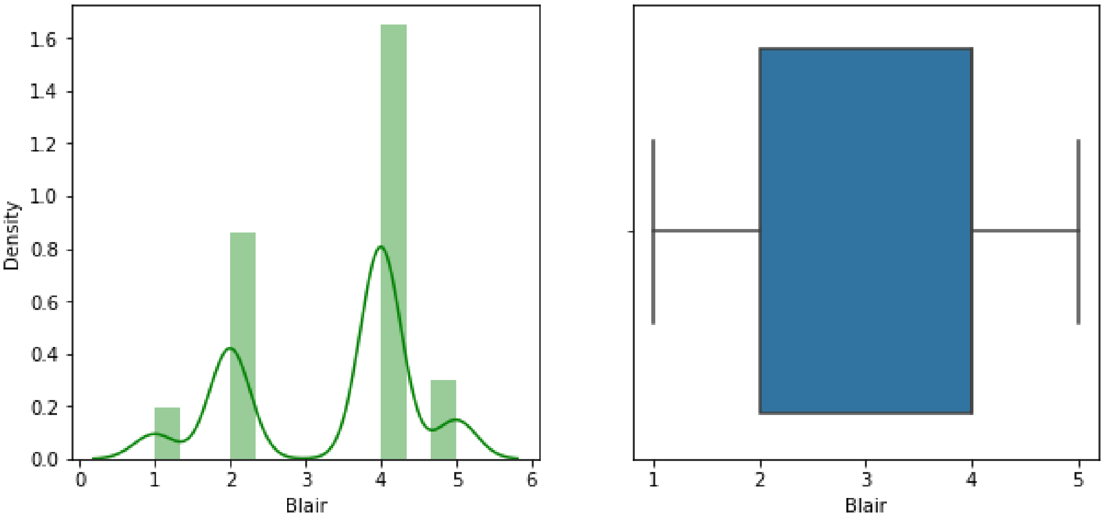
- Null Values
No null values were observed in the dataset.

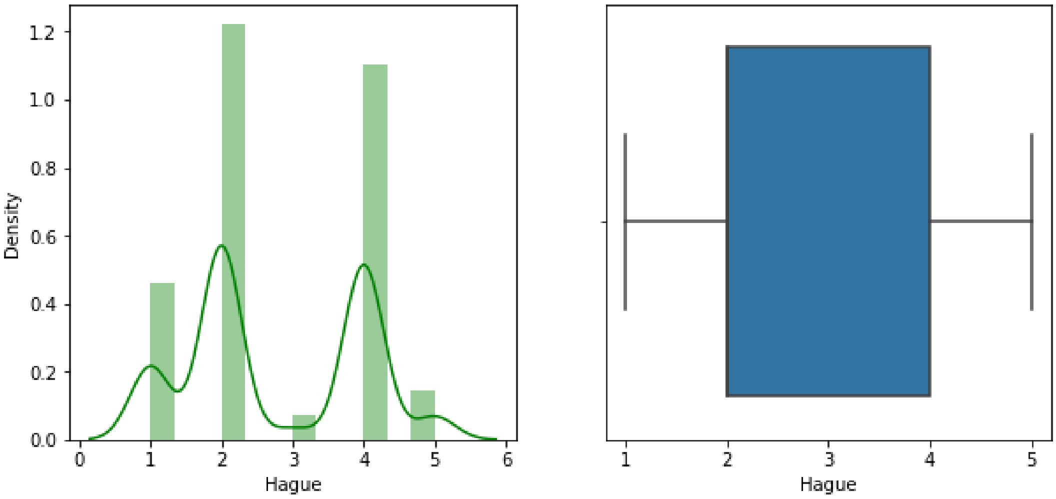
1.2. Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.

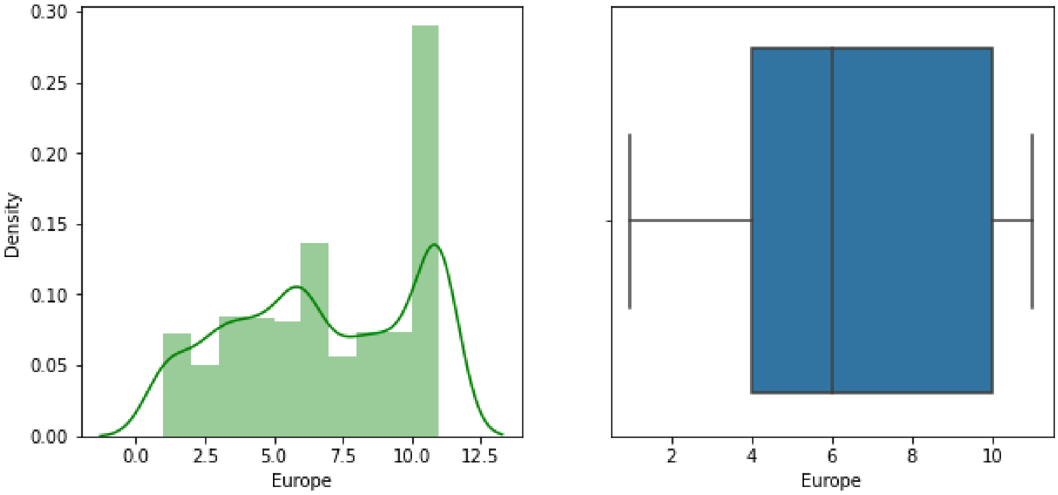


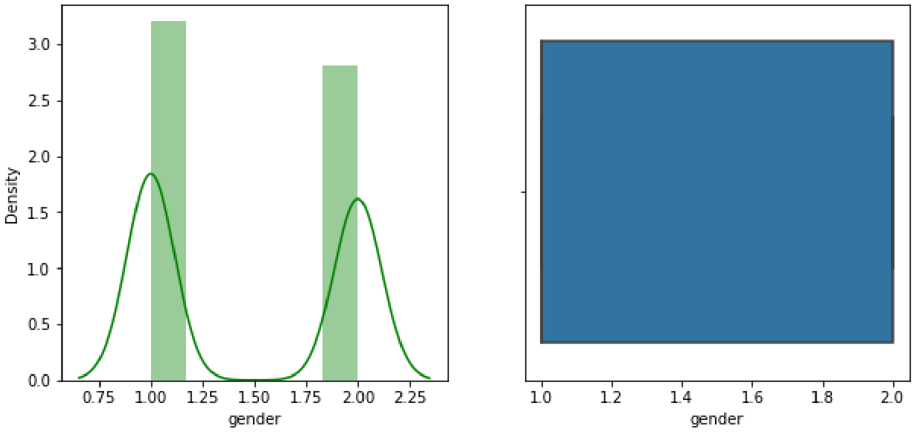





1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Density
0
1
2
3
4
5
6
Blair
1
2
3
4
5
Blair


1.2
1.0
0.8
0.6
0.4
0.2
0.0
Density
0
1
2
3
4
5
6
Hague
1
2
3
4
5
Hague


0.30
0.25
0.20
0.15
0.10
0.05
0.00
Density
0.0
2.5
5.0
7.5
10.0
12.5
Europe
2
4
6
8
10
Europe


3.0
2.5
2.0
1.5
1.0
0.5
0.0
Density
0.75
1.00
1.25
1.50
1.75
2.00
2.25
gender
1.0
1.2
1.4
1.6
1.8
2.0
gender

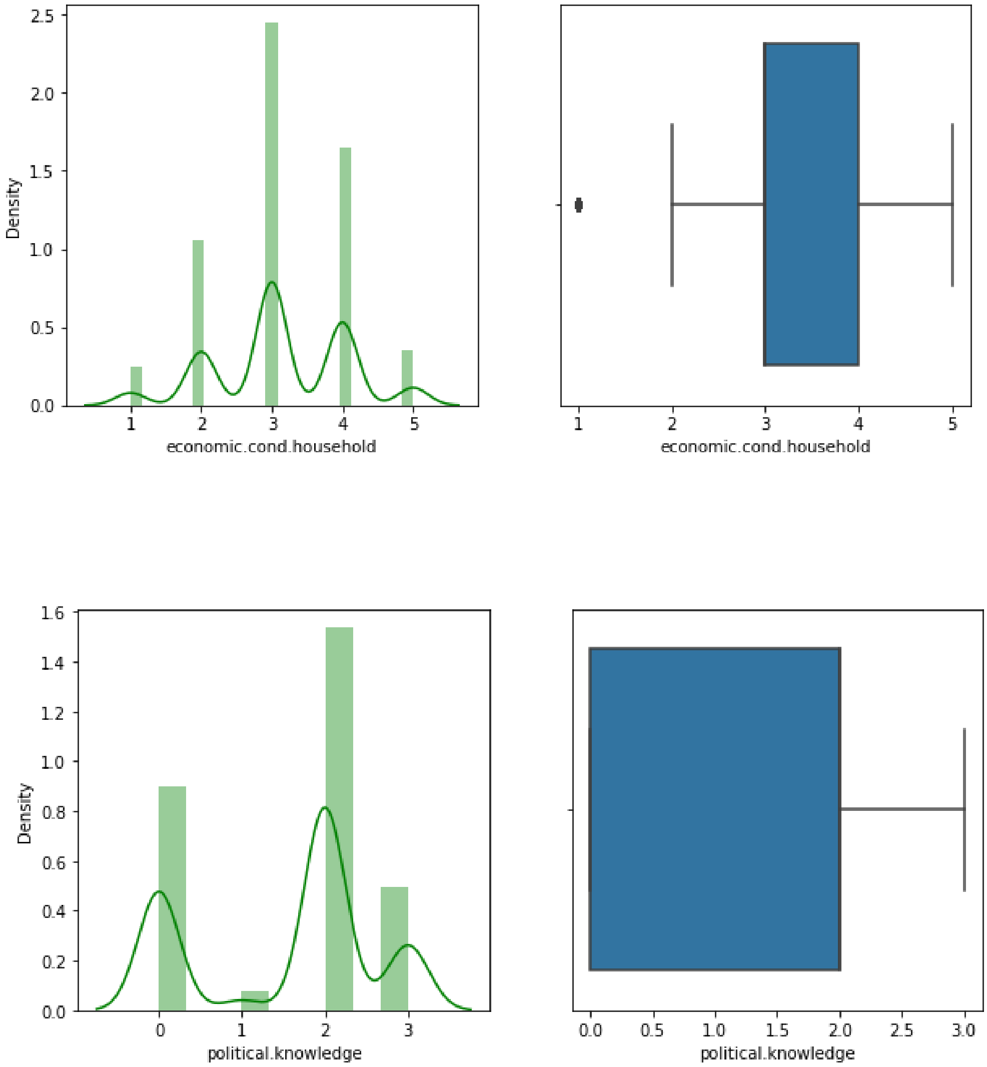


Fig. 1. Univariate Analysis

Inferences:

- Outlier is observed in ' economic.cond.national' and the survey states there is a majority of neutral response.
- The survey is taken of majorly people within the age group 40-75.
- 'Blair' & 'Hague' states that very less people have neutral response, 'Blair' has a majority of positive responses, however 'Hague' has a majority of negative responses.
- After conversion of the categoric data 'Vote' which had two levels- 'Conservative' and 'Labour', it is observed that 'Labour' group has higher count.
- Almost equal contribution is observed from the gender factor.

Bivariate Analysis:

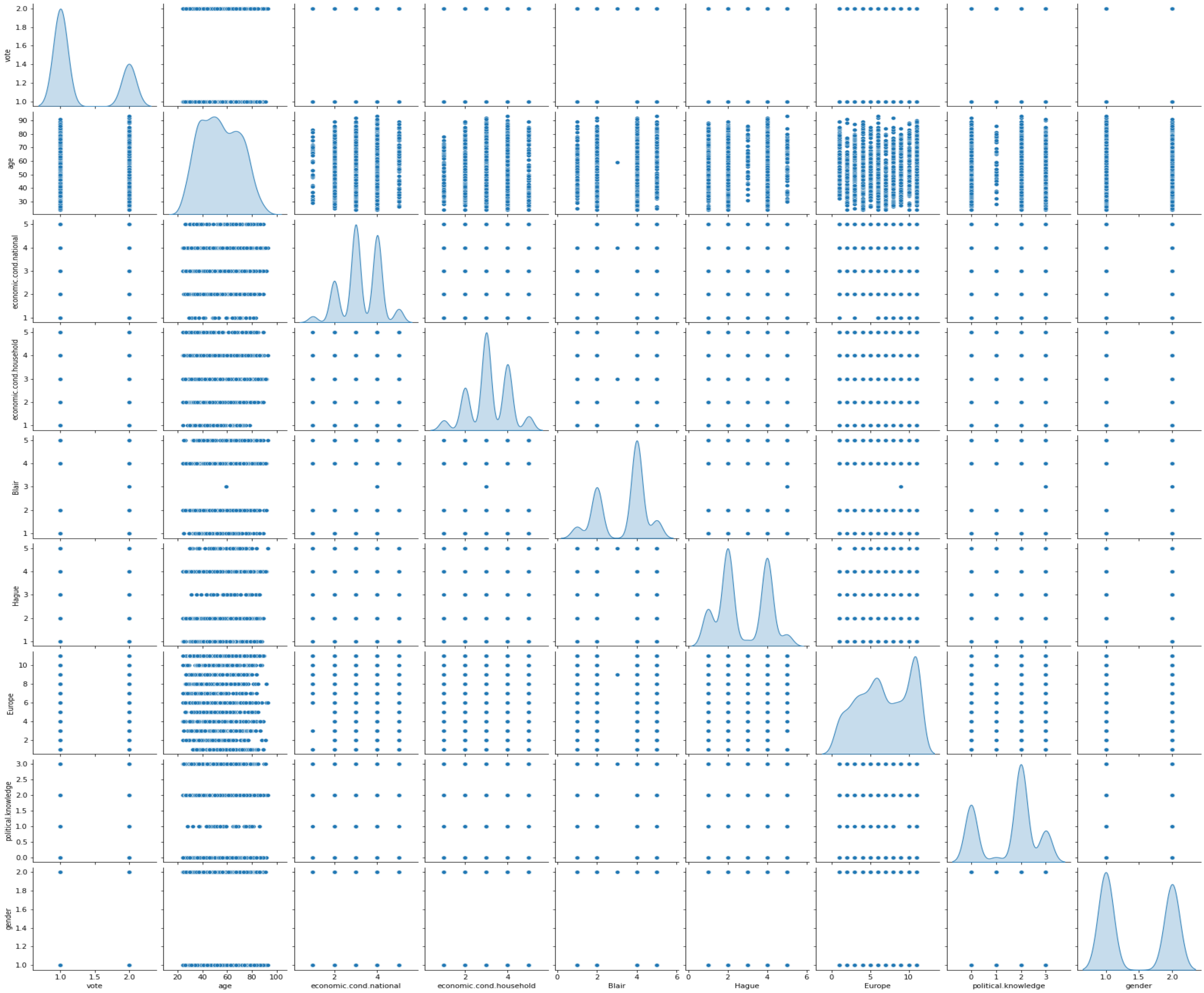


Fig. 2. Pair Plot for Bivariate Analysis

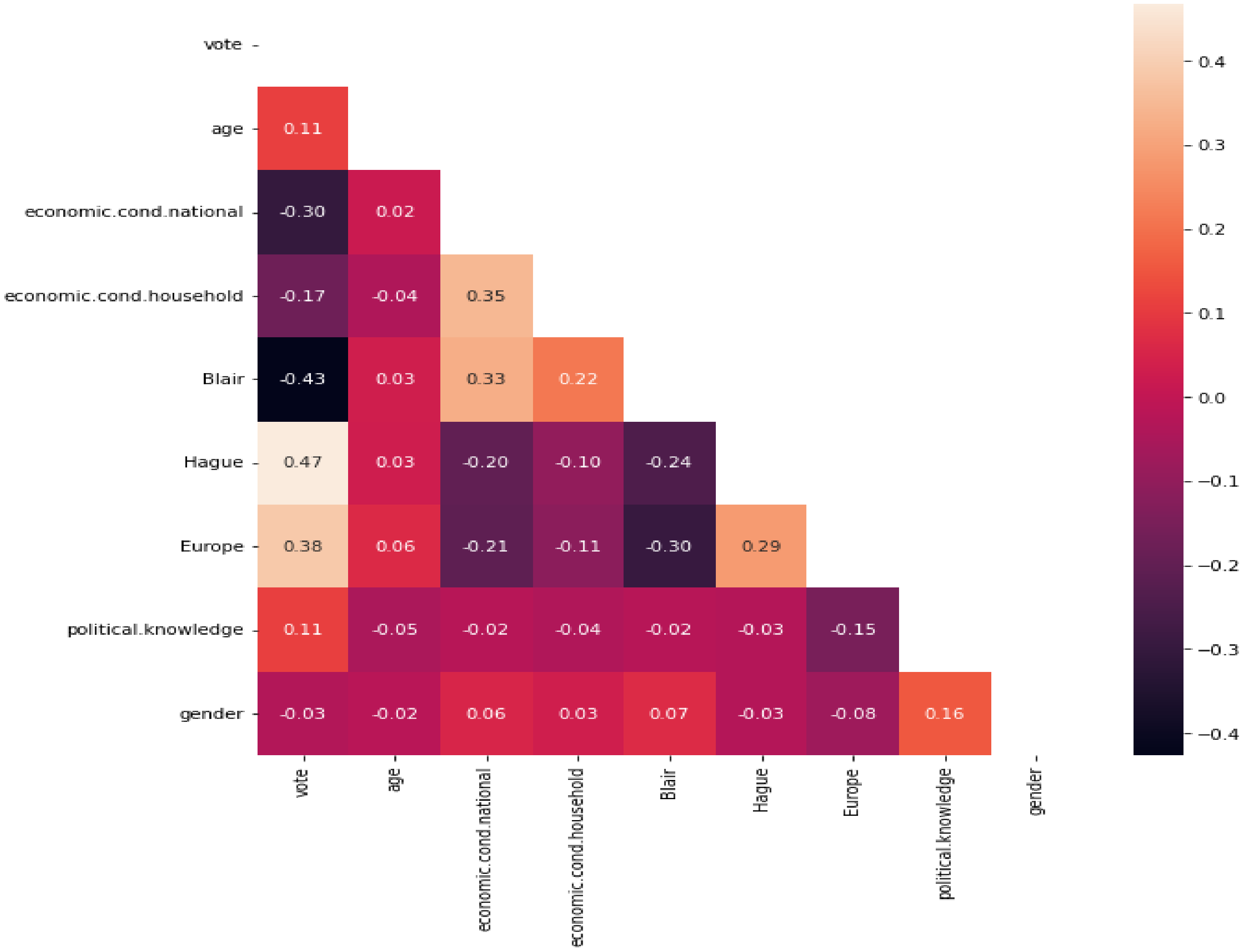


Fig. 3. Heat Map

- Outliers have been observed in 'economic.cond.national' &' economic.cond.household'.

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).

- Two variables were observed to be in a non-numeric data-type for which we manually encoded the variable and categorised them in 2 levels (1 & 2).
- Scaling is optional for models like Linear regression model, LDA &Logistic regression. However, for distance-based models like KNN scaling is required.

- Scaling is done only after the dataset is split into train & test.
- After scaling the data is first divided into two variables 'X' & 'y' which includes the independent and dependant variables respectively. The dependant variable is our target variable.
- After the identification of the target variable the dataset is divided into train-test split with 70:30 proportion.

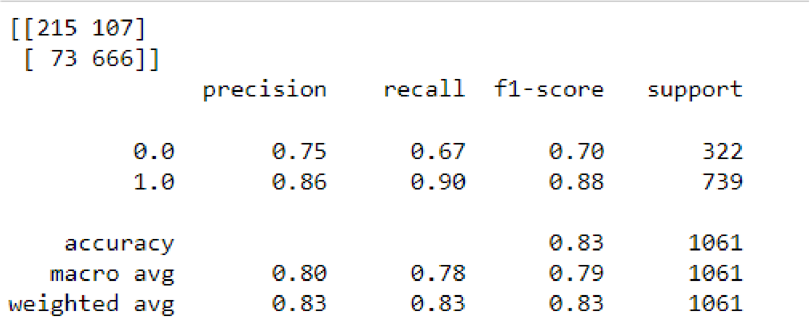
1.4 Apply Logistic Regression and LDA (linear discriminant analysis).

Logistic Regression:

For **Train Data:**

Model Score: 83.03%

Confusion Matrix & Metrics:

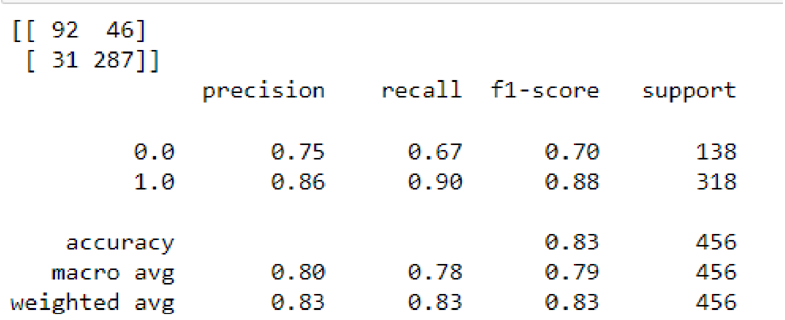


```
[[215 107]
 [ 73 666]]
              precision    recall  f1-score   support

         0.0       0.75      0.67      0.70       322
         1.0       0.86      0.90      0.88       739

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

For **Test Data**:

Model Score:83.11%

Confusion Matrix & Metrics:



```
[[ 92  46]
 [ 31 287]]
              precision    recall  f1-score   support

         0.0       0.75      0.67      0.70       138
         1.0       0.86      0.90      0.88       318

    accuracy                           0.83       456
   macro avg       0.80      0.78      0.79       456
weighted avg       0.83      0.83      0.83       456
```

LDA:

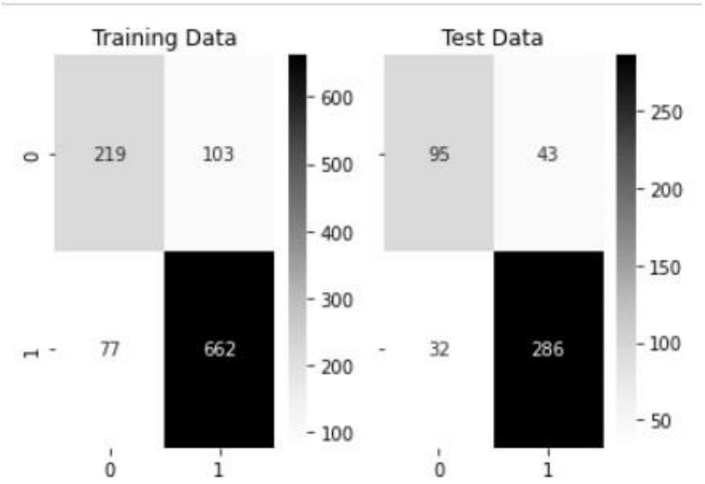
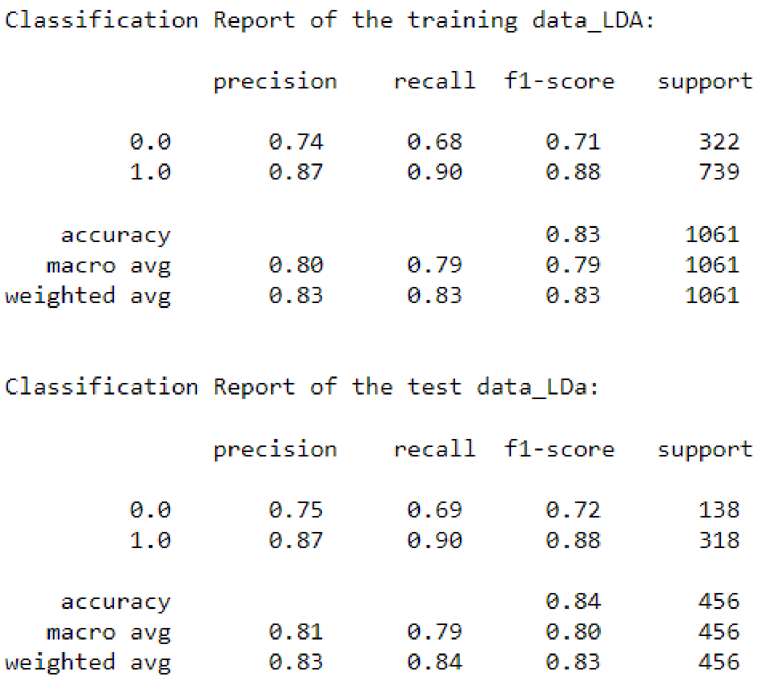


Fig. 4. LDA Confusion Matrix graph

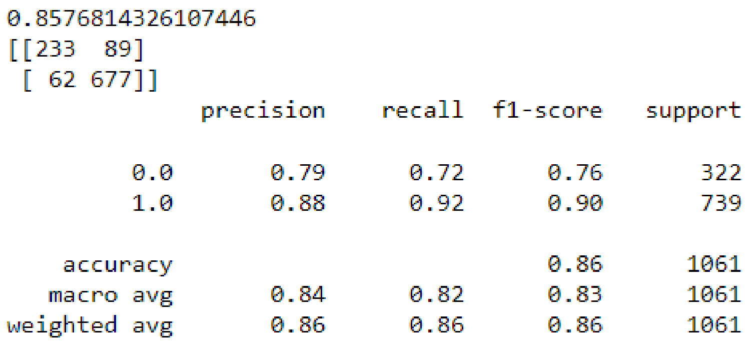


```
Classification Report of the training data_LDA:

              precision    recall  f1-score   support

         0.0       0.74      0.68      0.71       322
         1.0       0.87      0.90      0.88       739

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

Classification Report of the test data_LDa:

              precision    recall  f1-score   support

         0.0       0.75      0.69      0.72       138
         1.0       0.87      0.90      0.88       318

    accuracy                           0.84       456
   macro avg       0.81      0.79      0.80       456
weighted avg       0.83      0.84      0.83       456
```

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

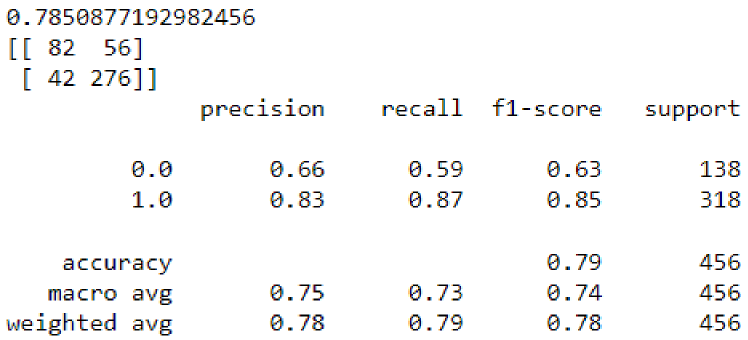
KNN:

For Train Data:



```
0.8576814326107446
[[233  89]
 [ 62 677]]
              precision    recall  f1-score   support

         0.0       0.79      0.72      0.76       322
         1.0       0.88      0.92      0.90       739

    accuracy                           0.86      1061
   macro avg       0.84      0.82      0.83      1061
weighted avg       0.86      0.86      0.86      1061
```

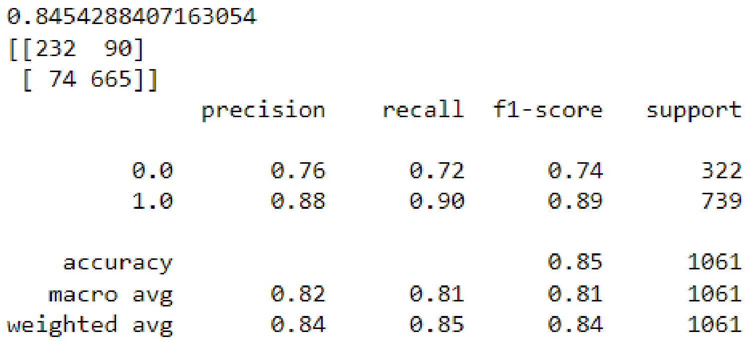
For Test Data:



```
0.7850877192982456
[[ 82  56]
 [ 42 276]]
              precision    recall  f1-score   support

         0.0       0.66      0.59      0.63       138
         1.0       0.83      0.87      0.85       318

    accuracy                           0.79       456
   macro avg       0.75      0.73      0.74       456
weighted avg       0.78      0.79      0.78       456
```

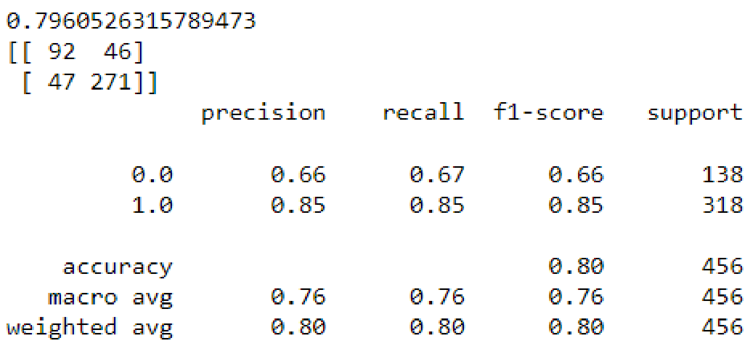
With K=10;

For Train Data:



```
0.8454288407163054
[[232  90]
 [ 74 665]]
              precision    recall  f1-score   support

         0.0       0.76      0.72      0.74       322
         1.0       0.88      0.90      0.89       739

    accuracy                           0.85      1061
   macro avg       0.82      0.81      0.81      1061
weighted avg       0.84      0.85      0.84      1061
```

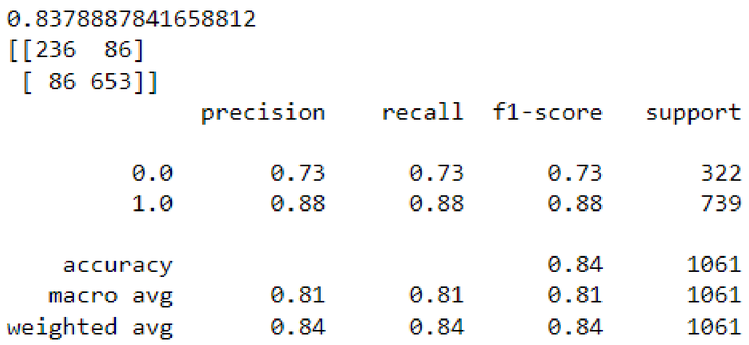
For Test Data:



```
0.7960526315789473
[[ 92  46]
 [ 47 271]]
              precision    recall  f1-score   support

         0.0       0.66      0.67      0.66       138
         1.0       0.85      0.85      0.85       318

    accuracy                           0.80       456
   macro avg       0.76      0.76      0.76       456
weighted avg       0.80      0.80      0.80       456
```

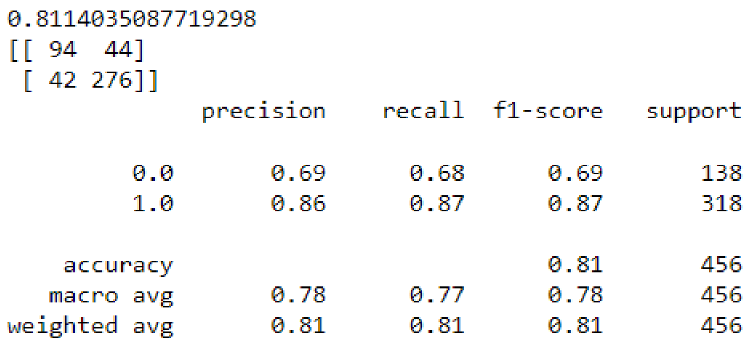
Naïve Baye's:

For Train Data:



```
0.8378887841658812
[[236  86]
 [ 86 653]]
              precision    recall  f1-score   support

         0.0       0.73      0.73      0.73       322
         1.0       0.88      0.88      0.88       739

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

For Test Data:



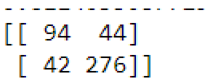
```
0.8114035087719298
[[ 94  44]
 [ 42 276]]
              precision    recall  f1-score   support

         0.0       0.69      0.68      0.69       138
         1.0       0.86      0.87      0.87       318

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

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

Using Random Forest:

Model Score:81.4%

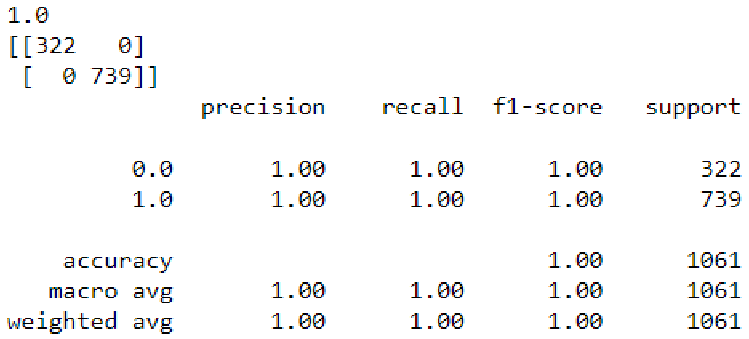
Confusion Matrix:



```
[[ 94  44]
 [ 42 276]]
```

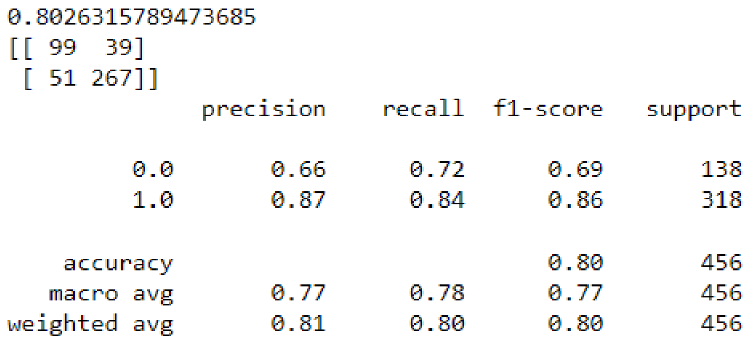
Bagging using Decision Tree:

For Train Data:



```
1.0
[[322   0]
 [  0 739]]
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00       322
         1.0       1.00      1.00      1.00       739

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

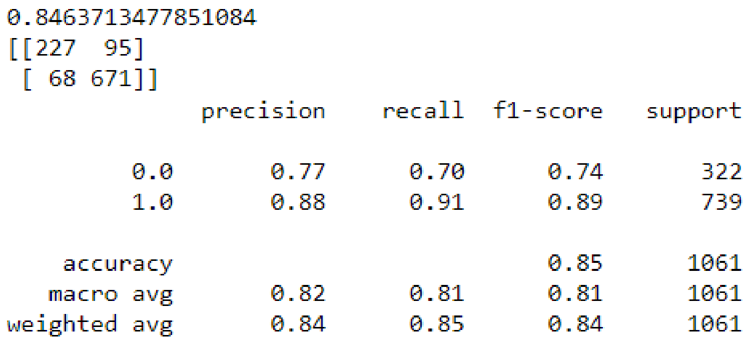
For Test Data:



```
0.8026315789473685
[[ 99  39]
 [ 51 267]]
              precision    recall  f1-score   support

         0.0       0.66      0.72      0.69       138
         1.0       0.87      0.84      0.86       318

    accuracy                           0.80       456
   macro avg       0.77      0.78      0.77       456
weighted avg       0.81      0.80      0.80       456
```

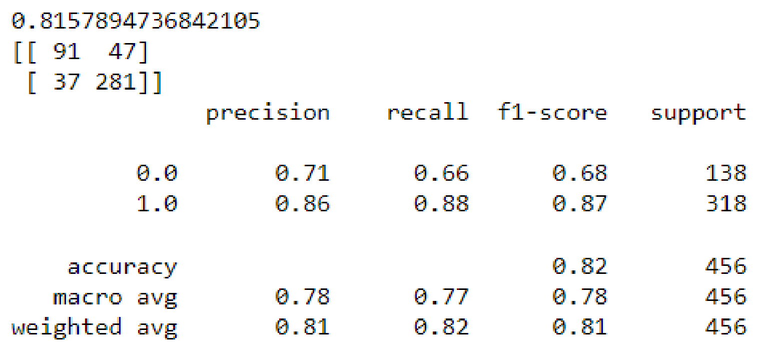
BOOSTING(AdaBoost):

For Train Data:



```
0.8463713477851084
[[227  95]
 [ 68 671]]
              precision    recall  f1-score   support

         0.0       0.77      0.70      0.74       322
         1.0       0.88      0.91      0.89       739

    accuracy                           0.85      1061
   macro avg       0.82      0.81      0.81      1061
weighted avg       0.84      0.85      0.84      1061
```

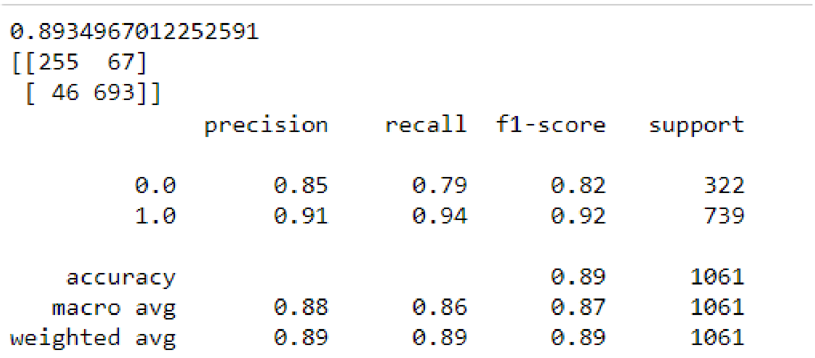
For Test Data:



```
0.8157894736842105
[[ 91  47]
 [ 37 281]]
              precision    recall  f1-score   support

         0.0       0.71      0.66      0.68       138
         1.0       0.86      0.88      0.87       318

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

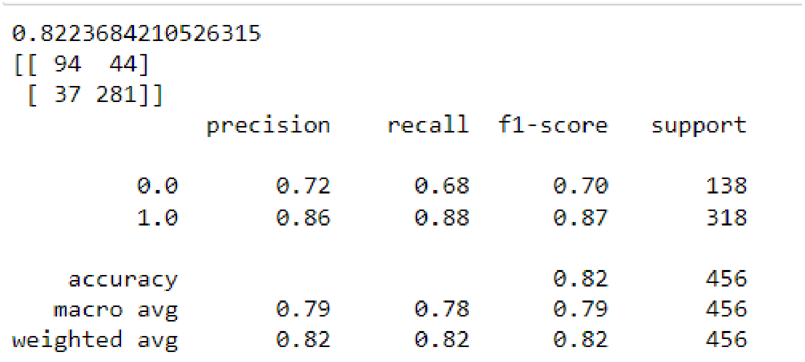
Boosting(Gradient Boosting):

For **Train** Data:



```
0.8934967012252591
[[255  67]
 [ 46 693]]
              precision    recall  f1-score   support

         0.0       0.85      0.79      0.82       322
         1.0       0.91      0.94      0.92       739

    accuracy                           0.89      1061
   macro avg       0.88      0.86      0.87      1061
weighted avg       0.89      0.89      0.89      1061
```

For **Test** Data:

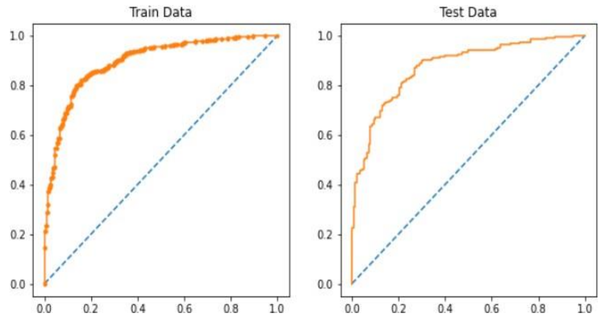
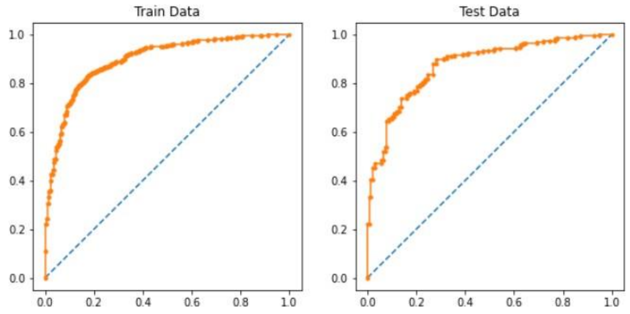
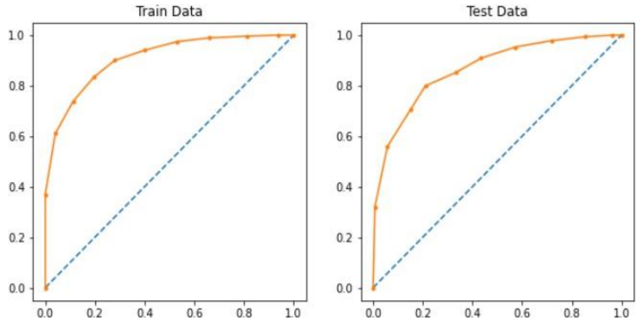


```
0.8223684210526315
[[ 94  44]
 [ 37 281]]
              precision    recall  f1-score   support

         0.0       0.72      0.68      0.70       138
         1.0       0.86      0.88      0.87       318

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

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.

ROC curves

Model	**Roc Curve(Train & Test)**

Logistic Regression	
LDA	
KNN	

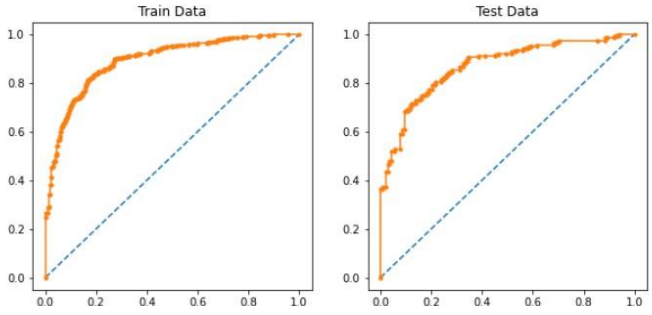
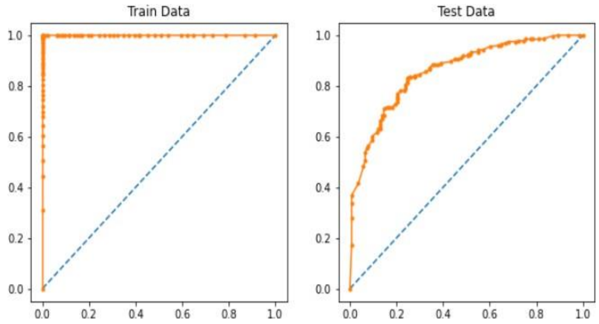
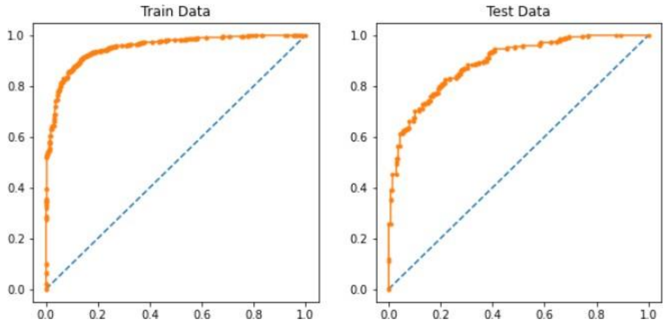
Naïve Bayes	
Bagging	
Boosting (Gradient Boosting)	

Fig. 5. ROC Curves of the Models

AUC Scores:

Model	**AUC Scores**

Logistic Regression	AUC CART TRAIN DATA: 0.895 AUC CART TEST DATA: 0.874
LDA	AUC CART TRAIN DATA: 0.895 AUC CART TEST DATA: 0.874 In [153]:
KNN	AUC CART TRAIN DATA: 0.908 AUC CART TEST DATA: 0.866
Naïve Bayes	AUC CART TRAIN DATA: 0.892 AUC CART TEST DATA: 0.869
Bagging	AUC CART TRAIN DATA: 1.000 AUC CART TEST DATA: 0.859
Gradient Boosting	AUC CART TRAIN DATA: 0.951 AUC CART TEST DATA: 0.891

Table 6. Area Under Curve Values

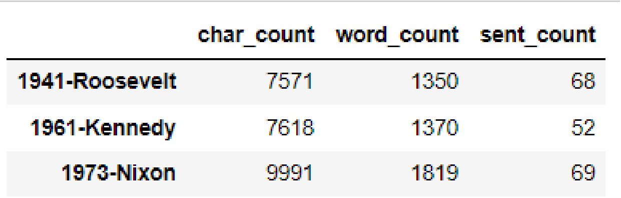
Problem 2:

In this project, we are going to work on the inaugural corpora from the nltk in Python. We will be looking at the following speeches of the Presidents of the United States of America:

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

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

- To find the characters we have used the raw() function.
- To find the words in the documents we have used the words() function.
- To find the sentences in the documents we have used the sents() function.
- The output of the same is as follows:



	char_count	word_count	sent_count
1941-Roosevelt	7571	1350	68
1961-Kennedy	7618	1370	52
1973-Nixon	9991	1819	69

Table 7. Count of Character, words, and Sentences

2.2 Remove all the stop words from all three speeches.

- Stop words are the words which occur most frequently and have no significance in the result. Hence, we have extracted the stop words from all the three texts.

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

- After removing of the stop words the words which occur most no of times for each president is as follows:

President	Most Occurring Word
Roosevelt	`'nation': 12, 'know': 10, 'spirit' : 9`
Kennedy	`'let': 16, 'us': 12, 'world': 8`
Nixon	`'us': 26, 'let': 22, 'america': 21`

Table 8. Most Occurring Words

2.4 Plot the word cloud of each of the speeches of the variable. (after removing the stop words)

Word Cloud of the speeches of the variables after removing the stop words are as follows:

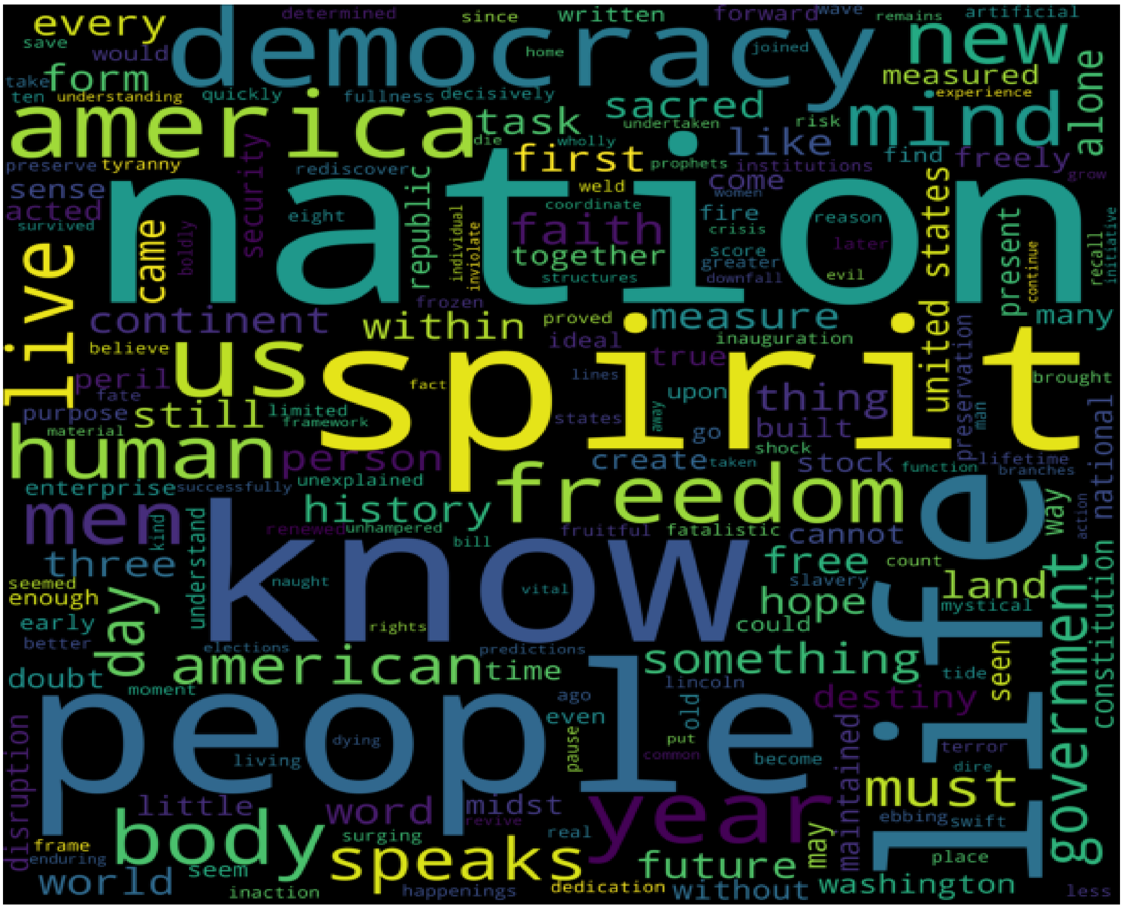


Fig. 6. Roosevelt Speech Words

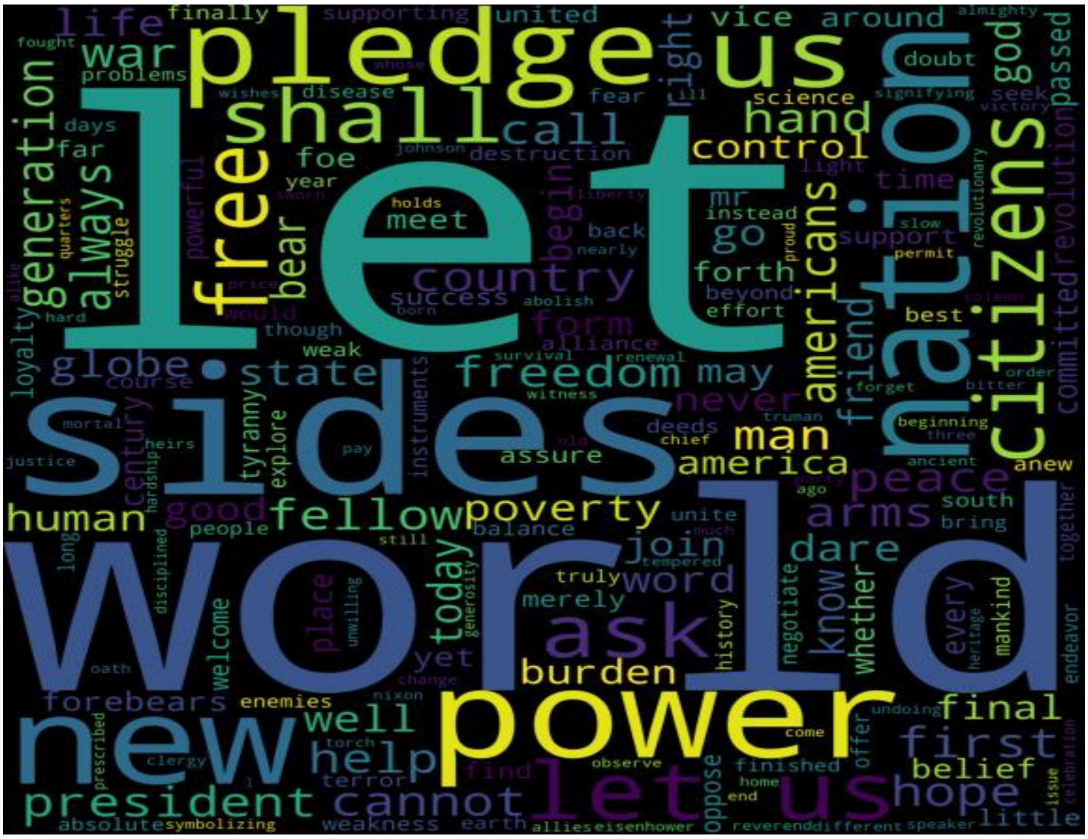


Fig. 7. Kennedy Speech Words

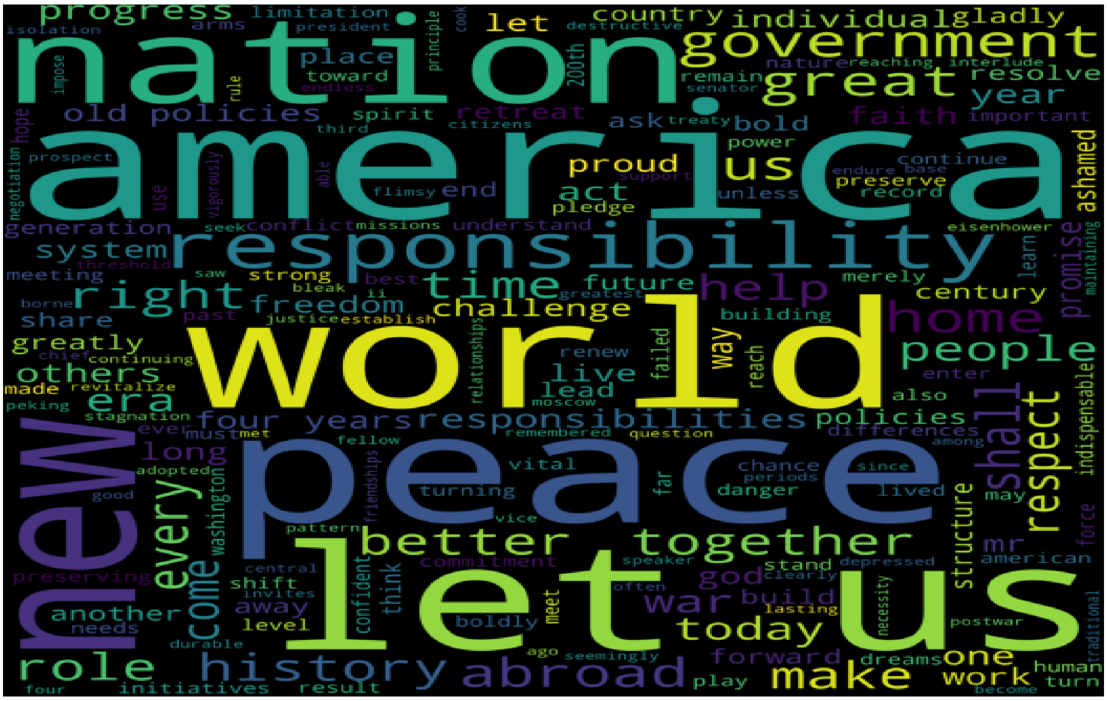


Fig. 8. Nixon Speech words