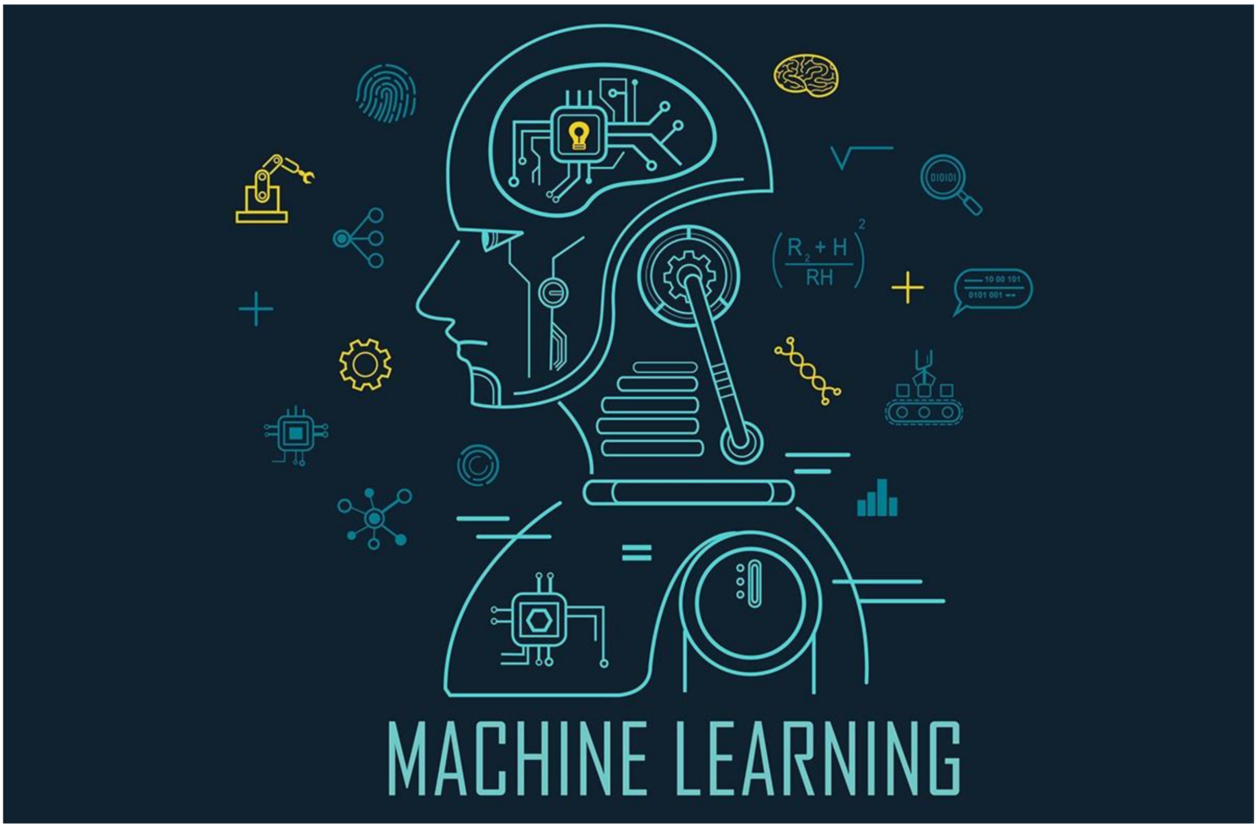
Business Report

MACHINE LEARNING



Prepared By: ARUNKUMAR S

Date: 07.08.2022

Batch Name: PGPDSBA Online Jan_E 2022

TABLE CONTENTS

Machine Learning project report

Problem 1:

Problem Statement:

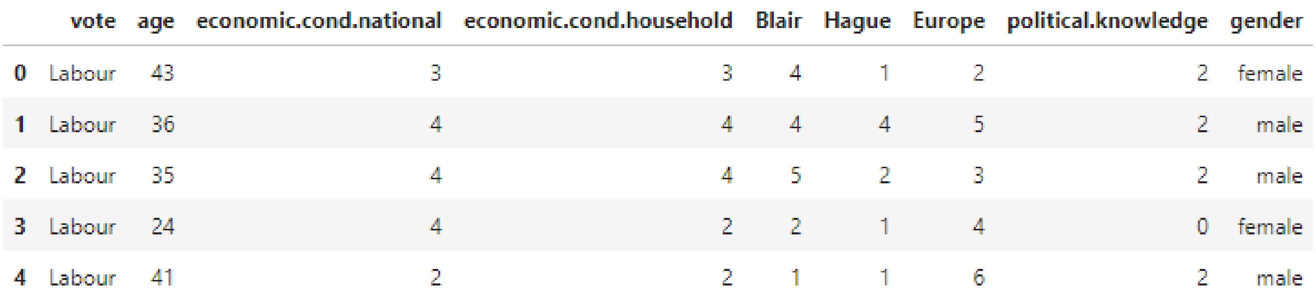
You are hired by one of the leading news channels CNBE who wants to analyze recent elections. This survey was conducted on 1525 voters with 9 variables. You have to build a model, to predict which party a voter will vote for on the basis of the given information, to create an exit poll that will help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

Domain:

Election_Data

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

The head of given data set "Election_Data.xlsx”



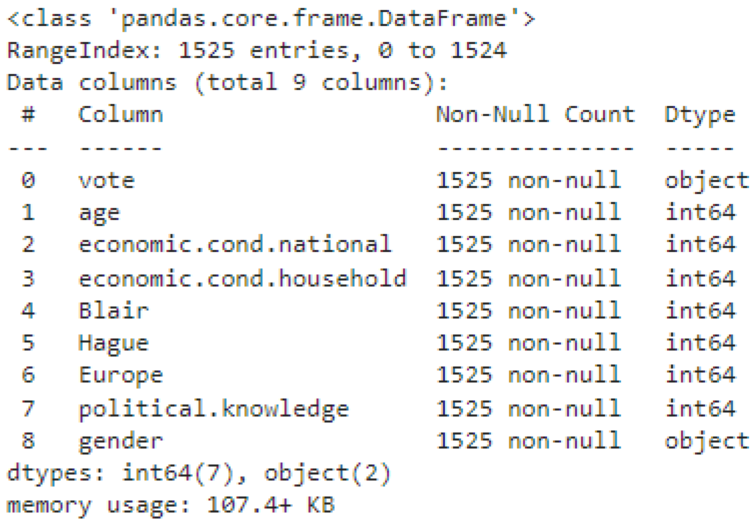
	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	Labour	43	3	3	4	1	2	2	female
1	Labour	36	4	4	4	4	5	2	male
2	Labour	35	4	4	5	2	3	2	male
3	Labour	24	4	2	2	1	4	0	female
4	Labour	41	2	2	1	1	6	2	male

Figure no: 1 – Head of given ‘Election_Data’ data set

Shape of the dataset:

Rows – 1525
Column - 9

Information about the dataset:

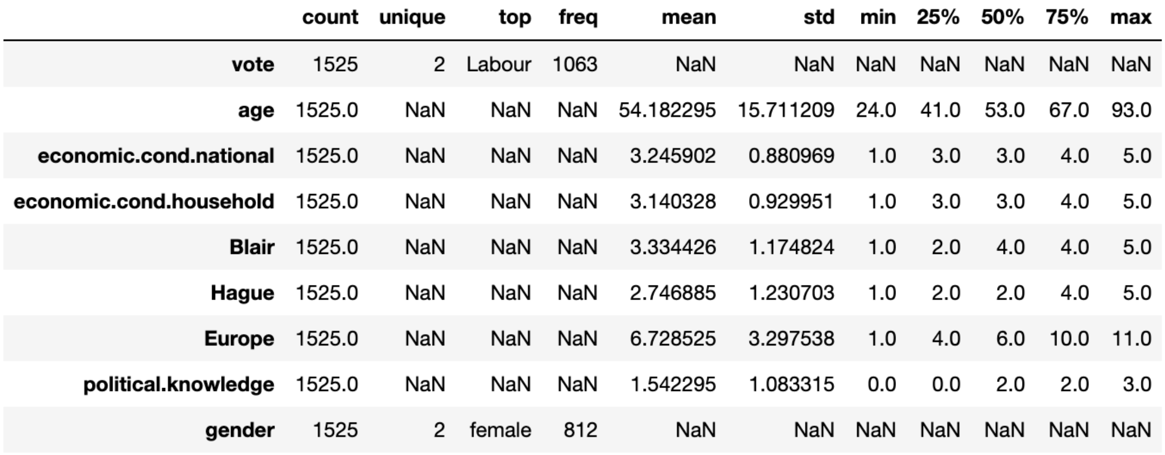


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   vote                     1525 non-null   object
 1   age                      1525 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
```

Figure no: 2 – Info of ‘Election_Data’ data set

Descriptive Statistics for the dataset

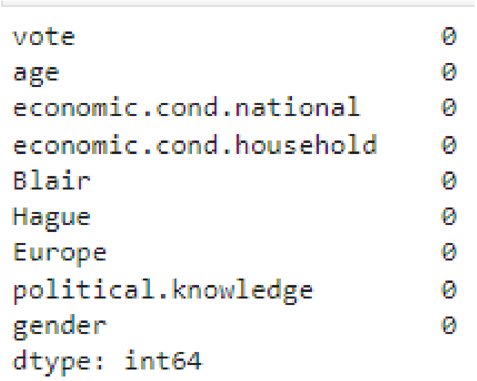
Description of the dataset:



	count	unique	top	freq	mean	std	min	25%	50%	75%	max
vote	1525	2	Labour	1063	NaN	NaN	NaN	NaN	NaN	NaN	NaN
age	1525.0	NaN	NaN	NaN	54.182295	15.711209	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	NaN	NaN	NaN	3.245902	0.880969	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	NaN	NaN	NaN	3.140328	0.929951	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	NaN	NaN	NaN	3.334426	1.174824	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	NaN	NaN	NaN	2.746885	1.230703	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	NaN	NaN	NaN	6.728525	3.297538	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	NaN	NaN	NaN	1.542295	1.083315	0.0	0.0	2.0	2.0	3.0
gender	1525	2	female	812	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure no: 3 – Description of 'Election_Data' data set

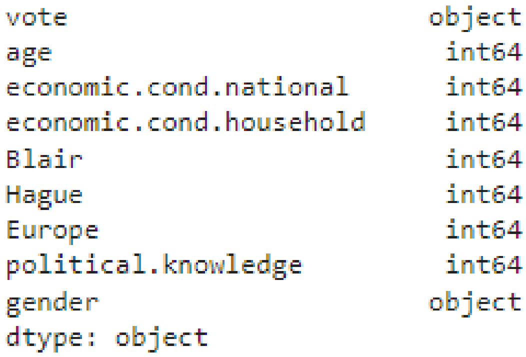
Null value presence:



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

Figure no: 4 – Result of null value checking

Data types of given dataset values:



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

Figure no: 5 – Data types of 'Election_Data' data set values

Duplicate value presence:

Result:

```
Total no of duplicate values = 8
```

Inference from the Observation:

- The Election dataset have 1525 rows and 9 columns.
- The mean and median for the only integer column ‘age’ is almost same indicating the column is normally distributed.
- ‘vote’ have two unique values Labour and Conservative, which is also a dependent variable
- ‘Gender’ has two unique values male and female.
- The data doesn’t contains the null value
- The dataset has few duplicates and removing them is the best choice as duplicates does not add any value
- All the variables except vote and gender are int64 datatypes. But when looking at the values in the dataset for the other variables, they all look like categorical columns except age

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

Univariate Analysis:

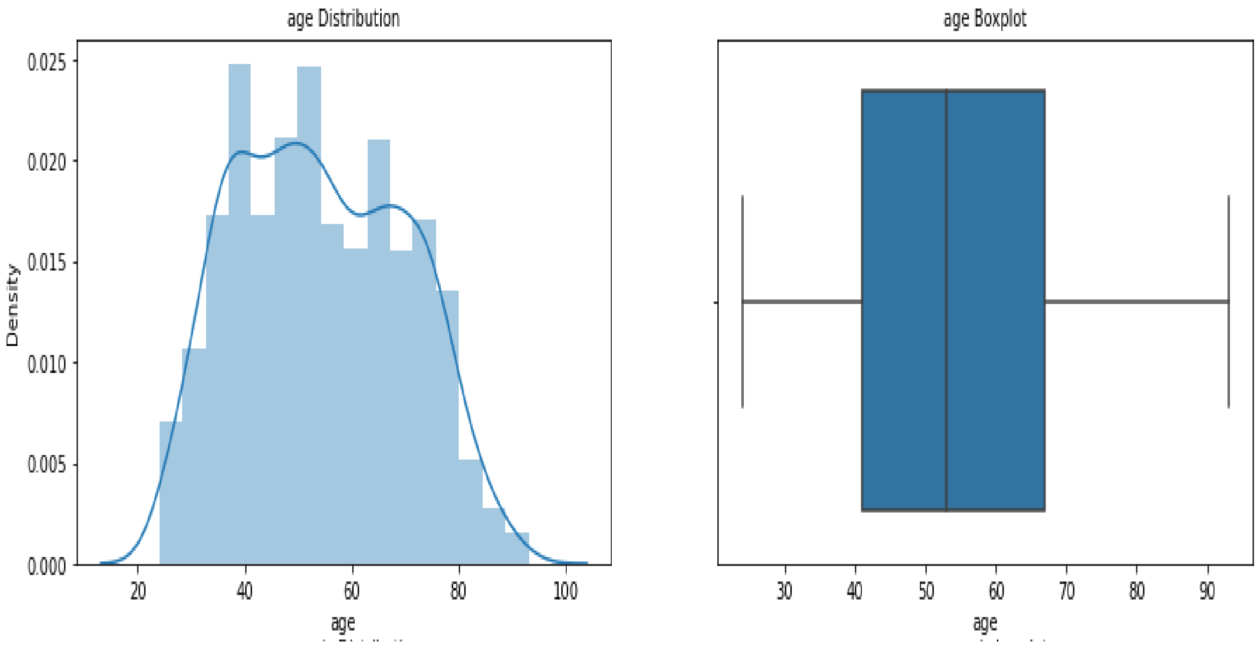
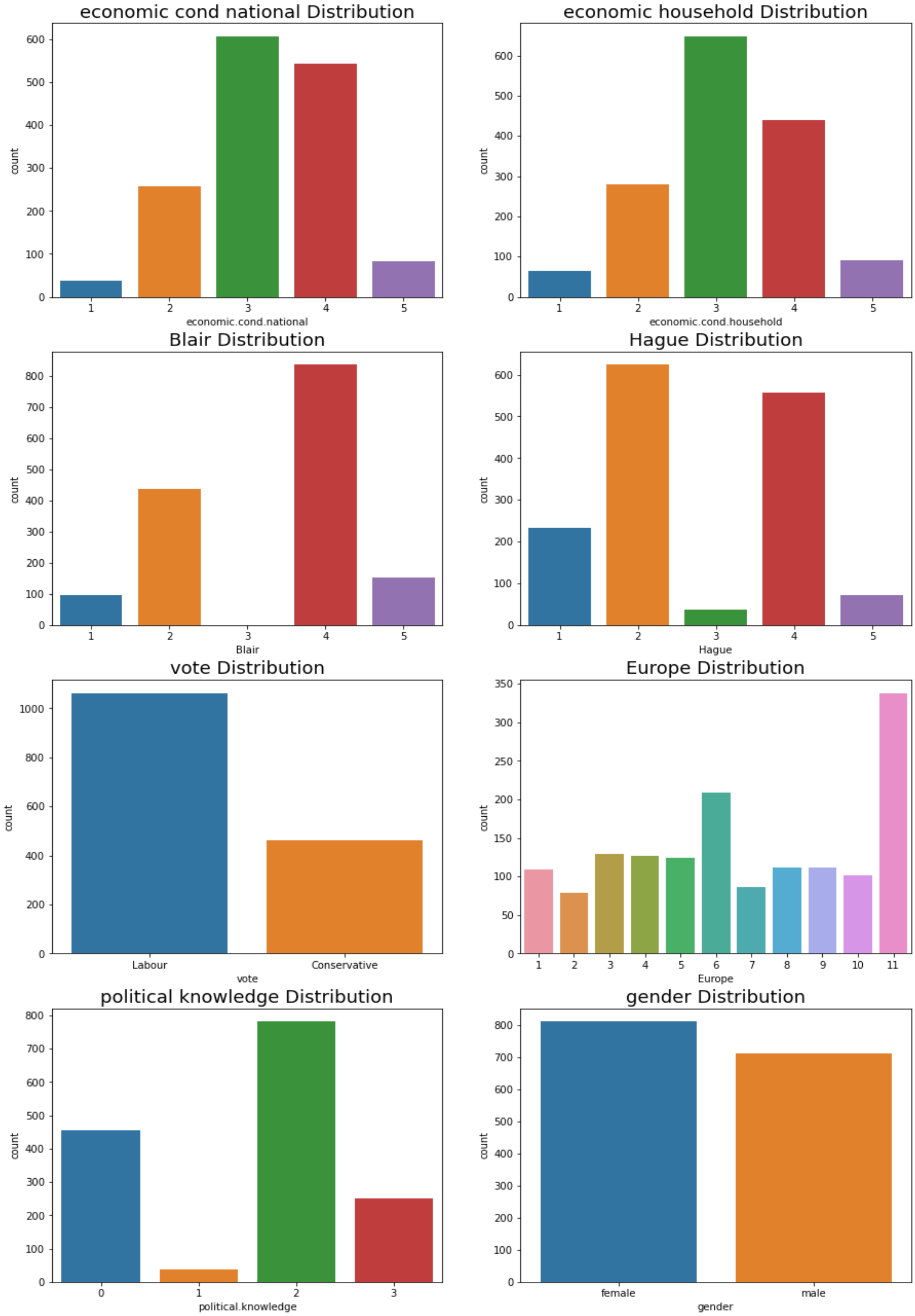


Figure no: 6 – Distribution of variable ‘age’ and it’s outliers checking

- Converting the necessary variables to object as it is meant to be. Because these variables have values that are numeric but are a categorical column.

- ‘age’ is the only integer variable and it is not having outliers. Also, the dist. plot shows that the variable is normally distributed.

Frequency distribution of the categorical variables:



economic cond national Distribution

economic household Distribution

Blair Distribution

Hague Distribution

vote Distribution

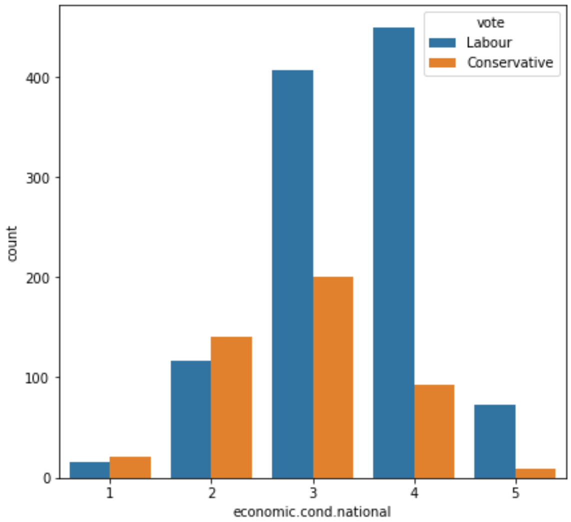
Europe Distribution

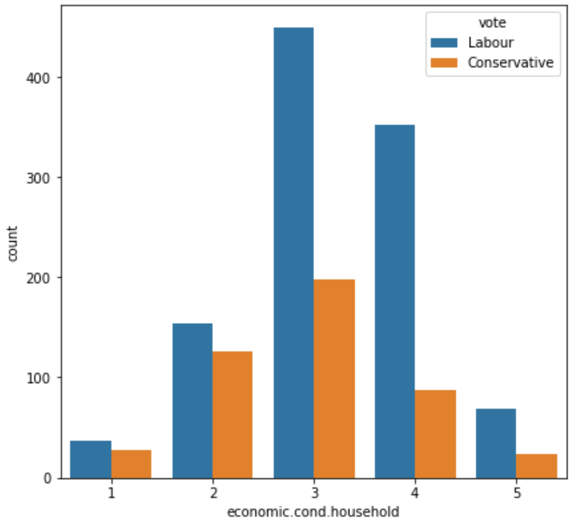
political knowledge Distribution

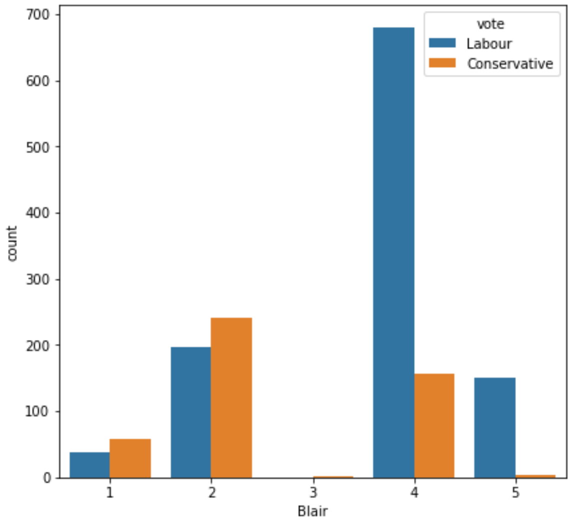
gender Distribution

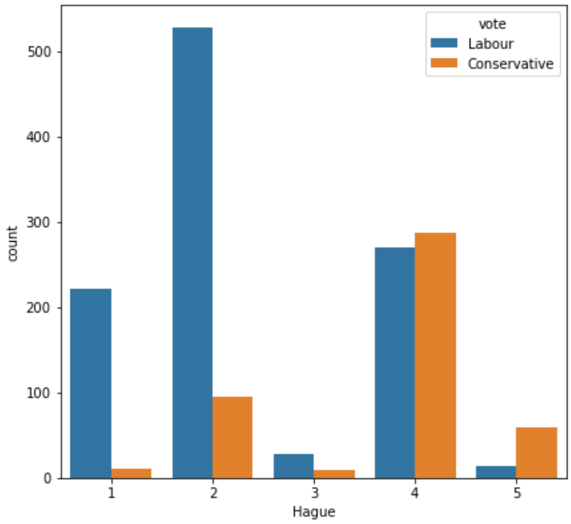
Figure no: 7 – Frequency distribution of the categorical variables

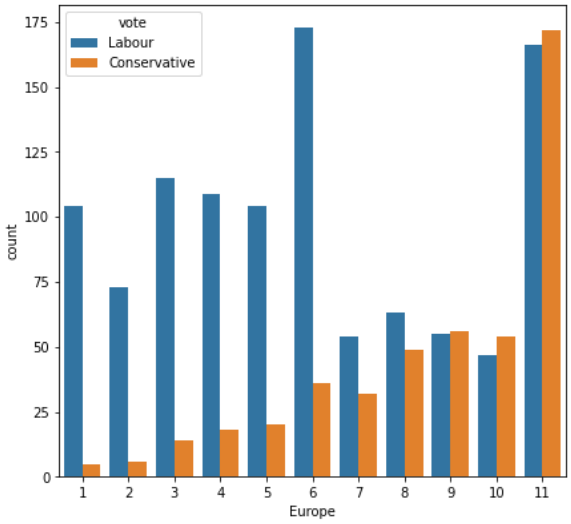
Bivariate Analysis

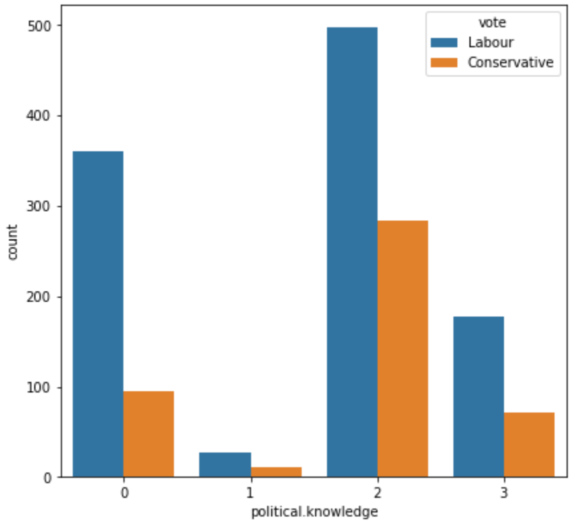












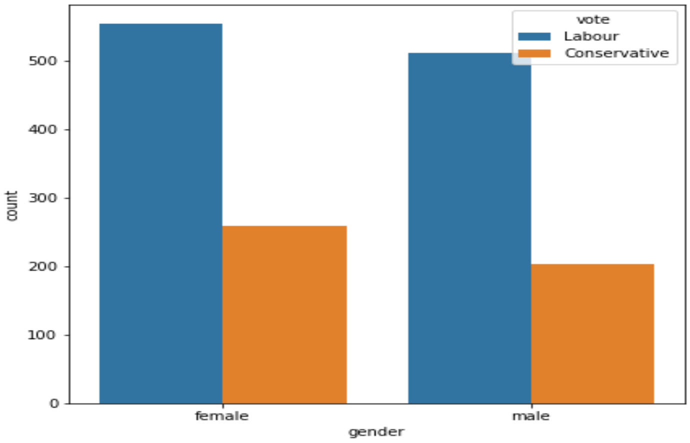


Figure no: 8 – Bivariate Analysis

- Labour gets the highest voting from both female and male voters.
- Almost in all the categories Labour is getting the maximum votes
- Conservative gets a little bit high votes from Europe ‘11’.

Pair Plot:

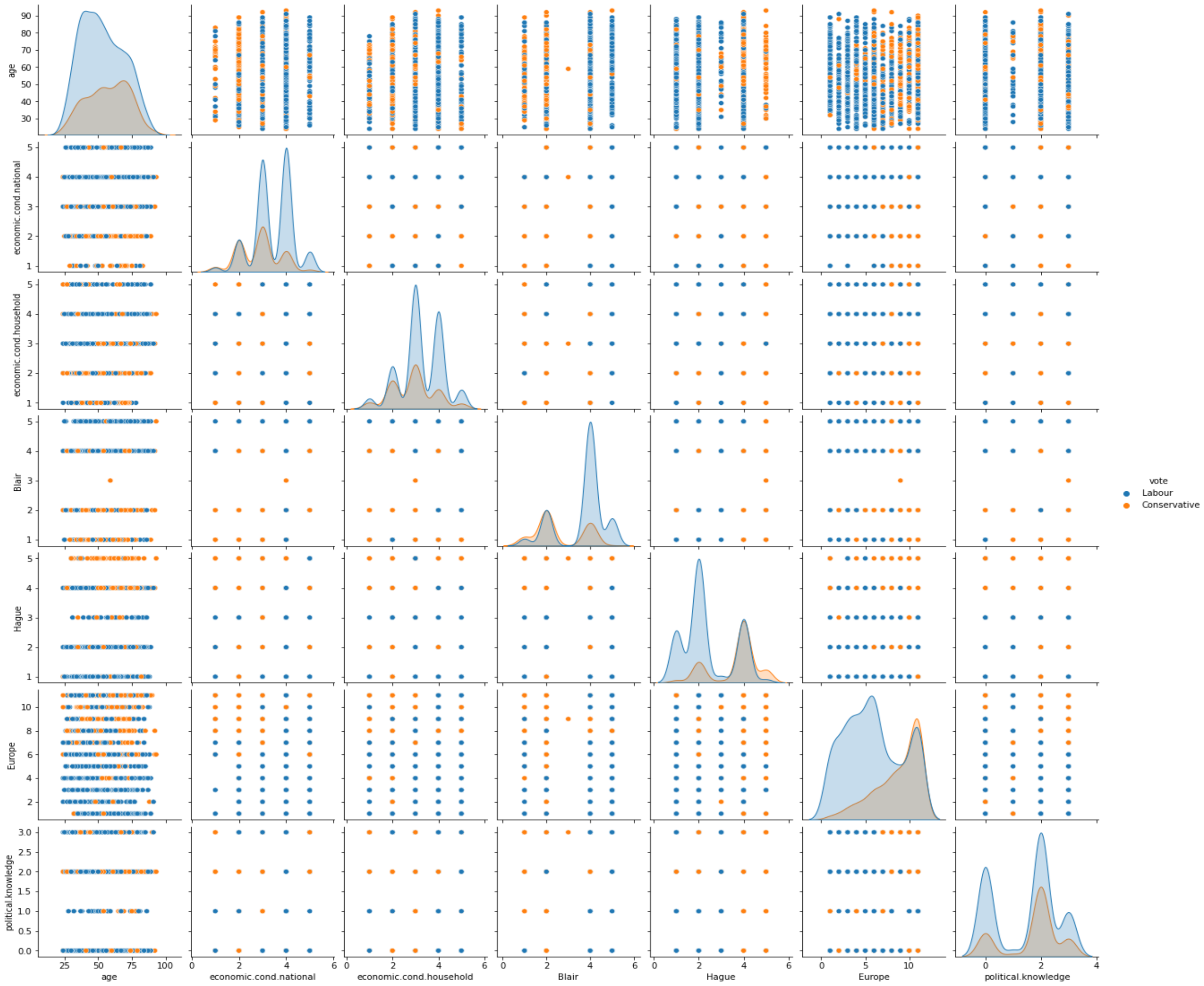


Figure no: 9– Pair plot

Heat Map:

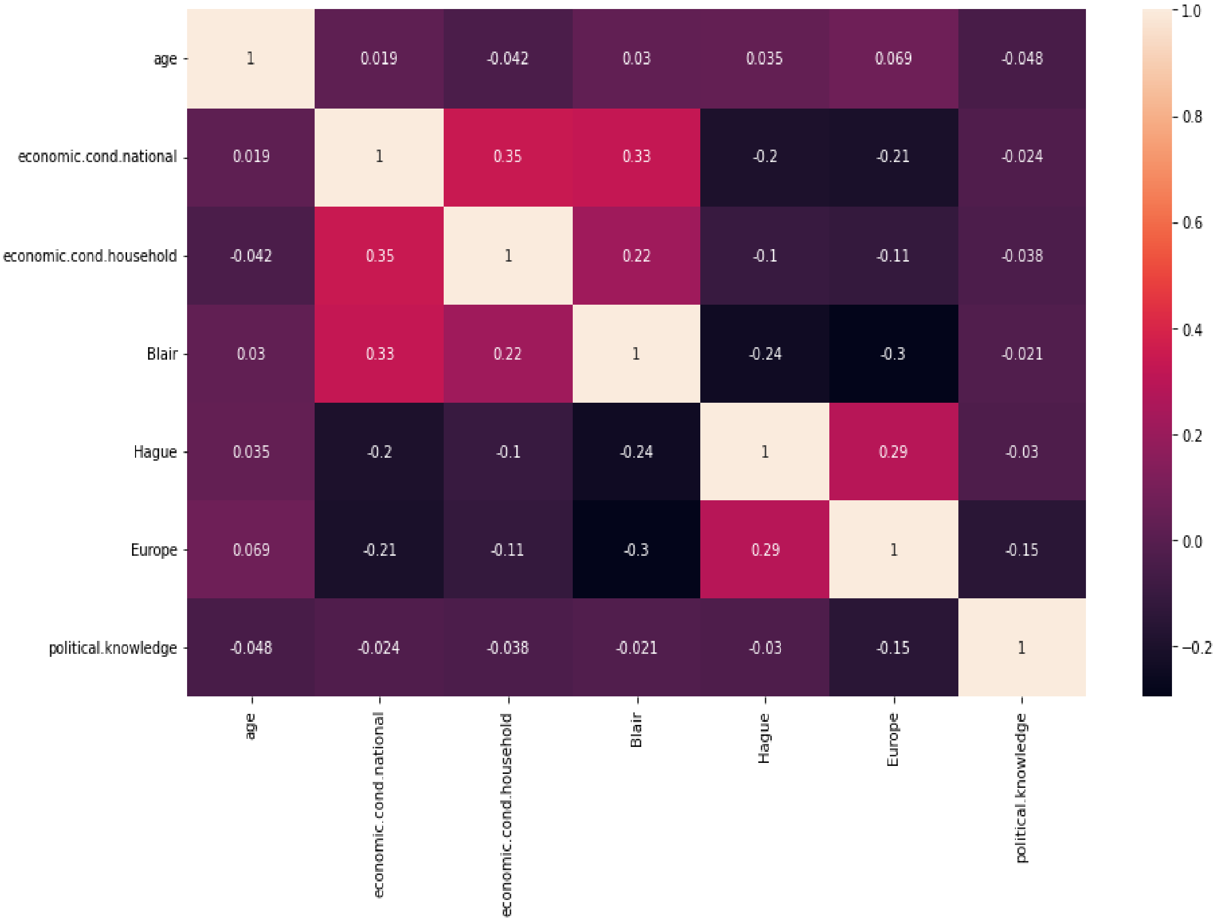


Figure no: 10– Heat map

- There is no correlation between the variables.

Data Preparation:

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).

- Encoding the dataset the variables ‘vote’ and ‘gender’ have string values. Converting them into numeric values for modelling,
- Splitting the data into train and test

Scaling:

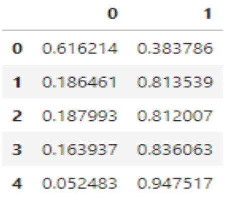
- We are not going to scale the data for Logistic regression, LDA and Naive Baye’s models as it is not necessary.
- But in case of KNN it is necessary to scale the data, as it a distance-based algorithm (typically based on Euclidean distance). Scaling the data gives similar weightage to all the variables

Modelling:

1.4 Apply Logistic Regression and LDA (linear discriminant analysis)

Logistic Regression:

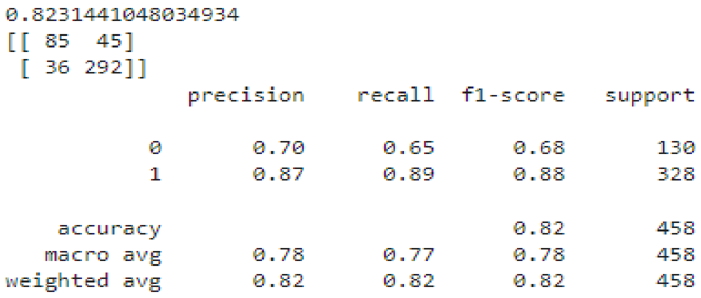
- Applying Logistic Regression and fitting the training data
- Predicting train and test,



	0	1
0	0.616214	0.383786
1	0.186461	0.813539
2	0.187993	0.812007
3	0.163937	0.836063
4	0.052483	0.947517

Figure no: 11– Predicting train and test,

Accuracy report for Logistic Regression:



```
0.8231441048034934
[[ 85  45]
 [ 36 292]]
              precision    recall  f1-score   support

           0       0.70      0.65      0.68       130
           1       0.87      0.89      0.88       328

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

Figure no: 12– Accuracy report for Logistic Regression

- The model is not overfitting or underfitting. Training and testing results shows that the model is excellent with good precision and recall values.

AUC ROC curve for Logistic Regression Test and Train:

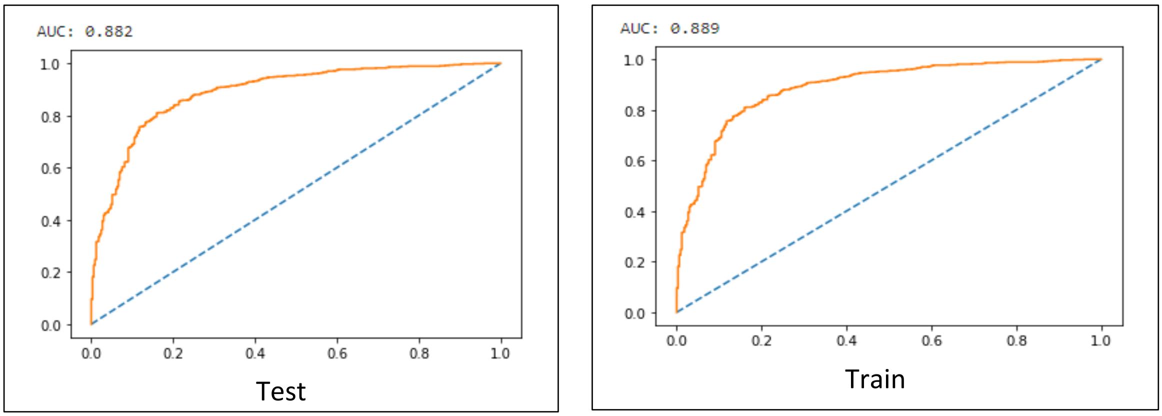
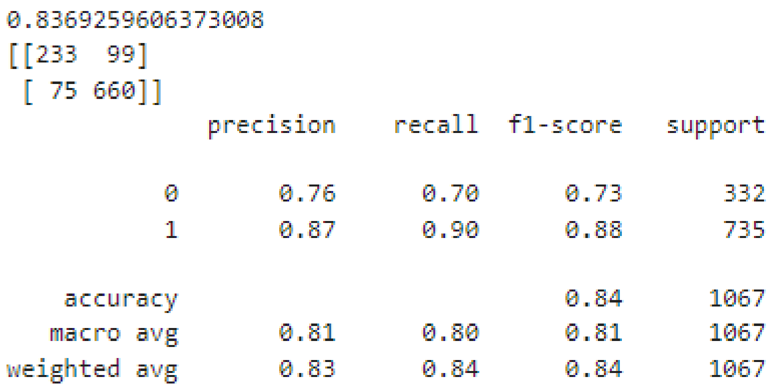


Figure no: 13– AUC ROC curve for Logistic Regression Test and Train

LDA (linear discriminant analysis):

- Applying LDA and fitting the training data
- Predicting train and test

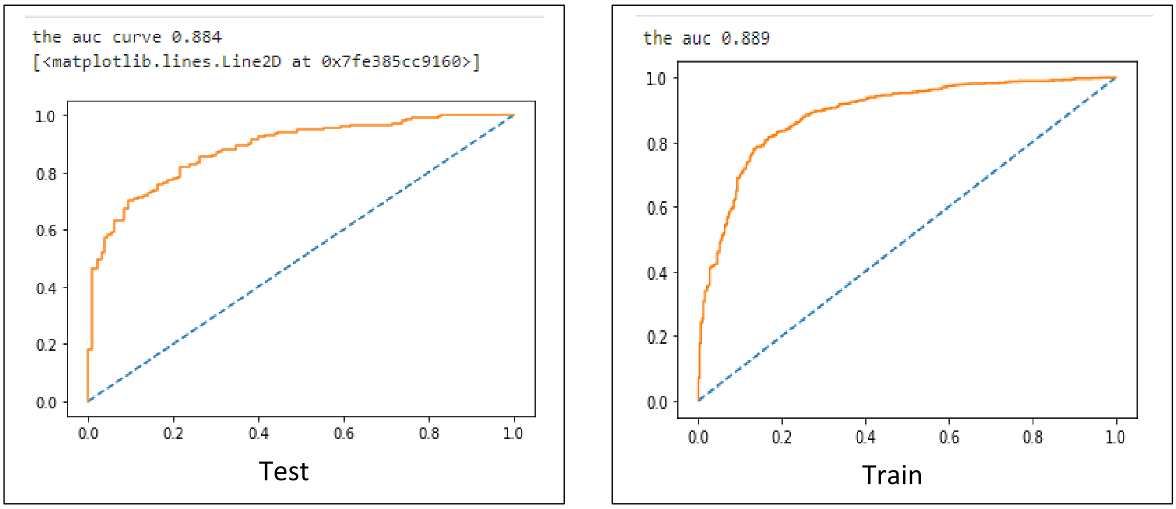
Accuracy report for linear discriminant analysis:



```
0.8369259606373008
[[233  99]
 [ 75 660]]
              precision    recall  f1-score   support

           0       0.76      0.70      0.73       332
           1       0.87      0.90      0.88       735

    accuracy                           0.84      1067
   macro avg       0.81      0.80      0.81      1067
weighted avg       0.83      0.84      0.84      1067
```

Figure no: 14– Accuracy report for linear discriminant analysis

AUC ROC curve for LDA Test and Train:



```
the auc curve 0.884
[<matplotlib.lines.Line2D at 0x7fe385cc9160>]
```

```
the auc 0.889
```

Figure no: 15– AUC ROC curve for LDA Test and Train

Inference from the LDA (linear discriminant analysis):

- Training and testing results shows that the model is excellent with good precision and recall values.

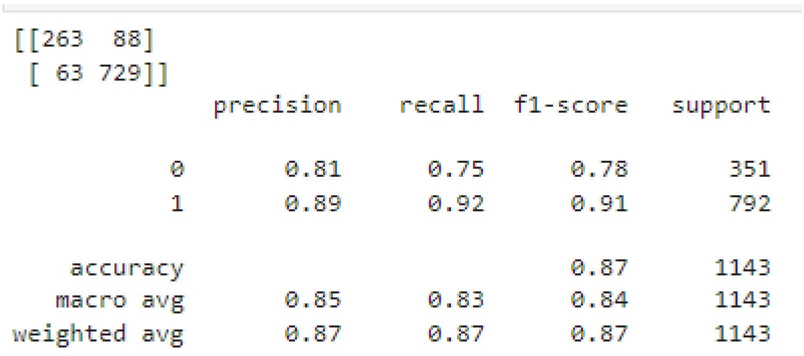
- The LDA model is better than Logistic regression with better Test accuracy and recall values

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

KNN Model:

- Scaling the dataset as it is required because KNN is a distance-based algorithm,
- Applying KNN and fitting the training data
- Predicting train and test,

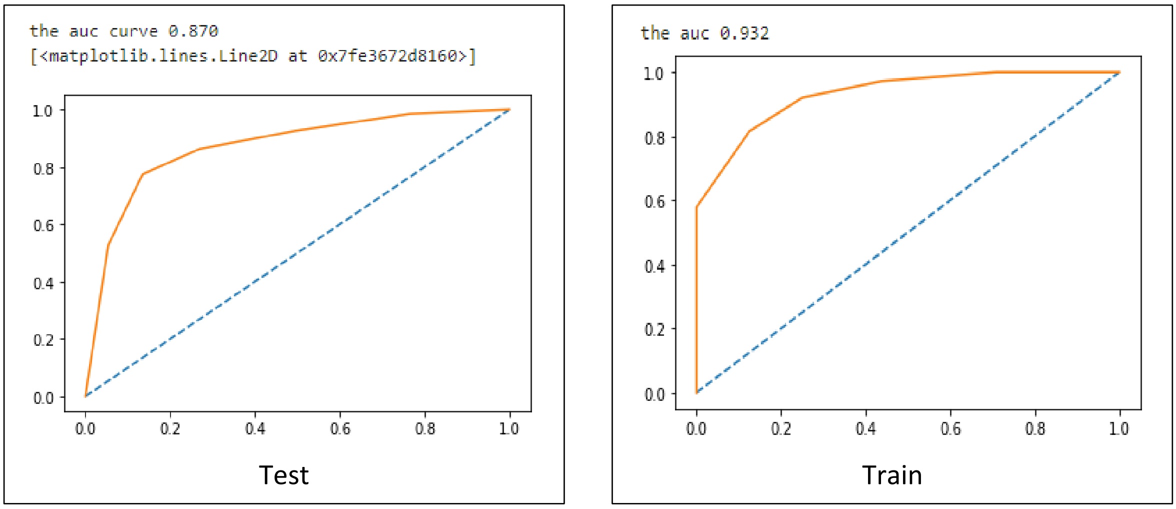
Accuracy report for KNN:



```
[[263  88]
 [ 63 729]]
              precision    recall  f1-score   support

           0       0.81      0.75      0.78       351
           1       0.89      0.92      0.91       792

    accuracy                           0.87      1143
   macro avg       0.85      0.83      0.84      1143
weighted avg       0.87      0.87      0.87      1143
```

Figure no: 16– Accuracy report for KNN

AUC ROC curve for KNN Test and Train:



```
the auc curve 0.870
[<matplotlib.lines.Line2D at 0x7fe3672d8160>]
```

```
the auc 0.932
```

Figure no: 17– AUC ROC curve for KNN Test and Train

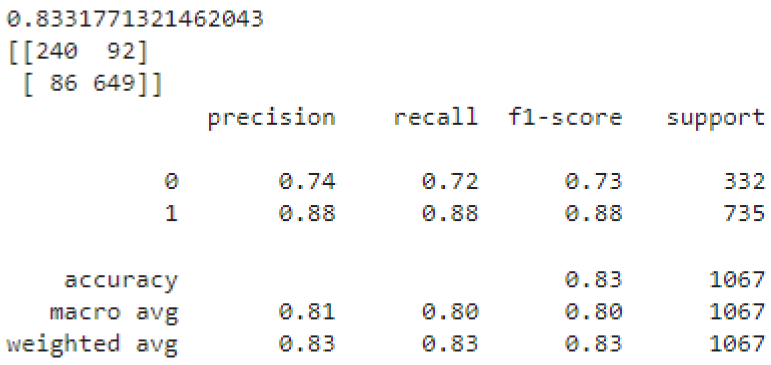
Inference from the KNN model:

- Training and testing results shows that the model is excellent with good precision and recall values.
-
- This KNN model have good accuracy and recall values.

Naive Bayes:

- Importing GaussianNB from sklearn and applying NB model
- Fitting the training data
- Predicting train and test,

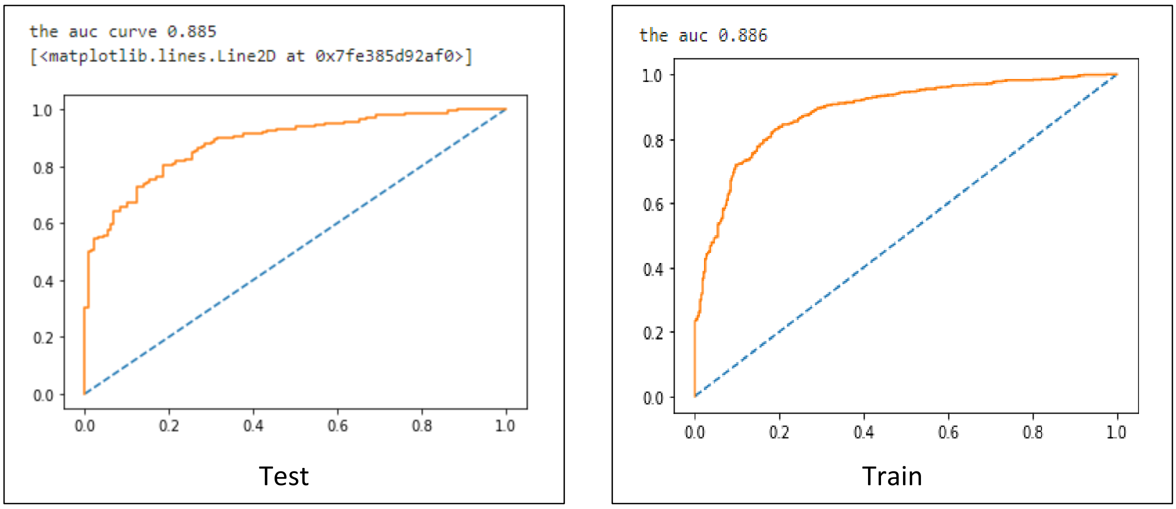
Train and Test accuracy:



```
0.8331771321462043
[[240  92]
 [ 86 649]]
              precision    recall  f1-score   support

           0       0.74      0.72      0.73       332
           1       0.88      0.88      0.88       735

    accuracy                           0.83      1067
   macro avg       0.81      0.80      0.80      1067
weighted avg       0.83      0.83      0.83      1067
```

Figure no: 18– Train and Test accuracy report for Naïve Bayes

AUC ROC curve for Naive Bayes Test and Train:



```
the auc curve 0.885
[<matplotlib.lines.Line2D at 0x7fe385d92af0>]
```

```
the auc 0.886
```

Figure no: 19– AUC ROC curve for Naive Bayes Test and Train

Inference from the Naive Bayes:

- Training and Testing results shows that the model neither overfitting nor underfitting.

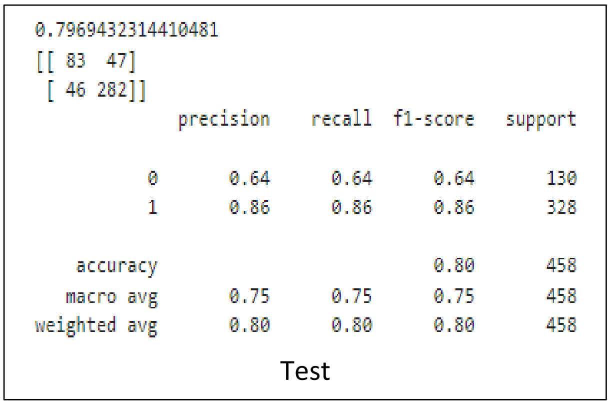
- The Naive Bayes model also performs well with better accuracy and recall values.

- Even though NB and KNN have same Train and Test accuracy. Based on their recall value in test dataset it is evident that KNN performs better than Naive Bayes.

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

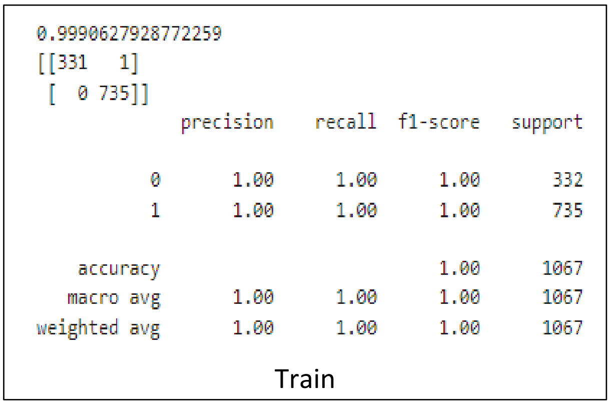
- Using GridSearchCV and tuning the model which helps us in finding the best parameters for the model]
- Predicting the Train and test,
- Basic Decision Tree classifier with gini index and random state of 1
- Using Bagging to improve the performance of the model.
- Applying the model and predicting the train and test data,

Bagging Test and Train accuracy report:



```
0.7969432314410481
[[ 83  47]
 [ 46 282]]
              precision    recall  f1-score   support

           0       0.64      0.64      0.64       130
           1       0.86      0.86      0.86       328

    accuracy                           0.80       458
   macro avg       0.75      0.75      0.75       458
weighted avg       0.80      0.80      0.80       458
```

Test

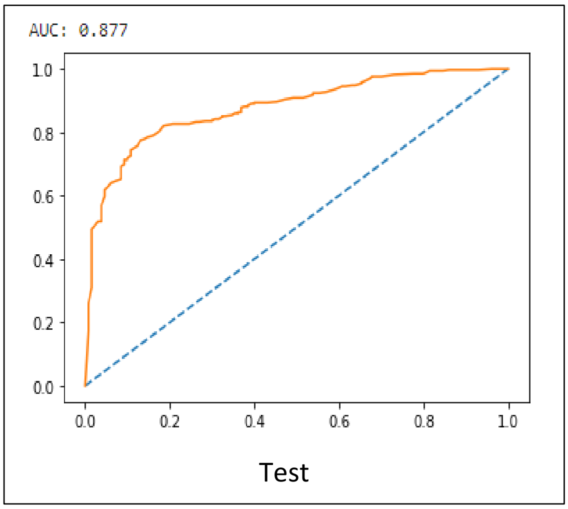


```
0.9990627928772259
[[331   1]
 [  0 735]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       332
           1       1.00      1.00      1.00       735

    accuracy                           1.00      1067
   macro avg       1.00      1.00      1.00      1067
weighted avg       1.00      1.00      1.00      1067
```

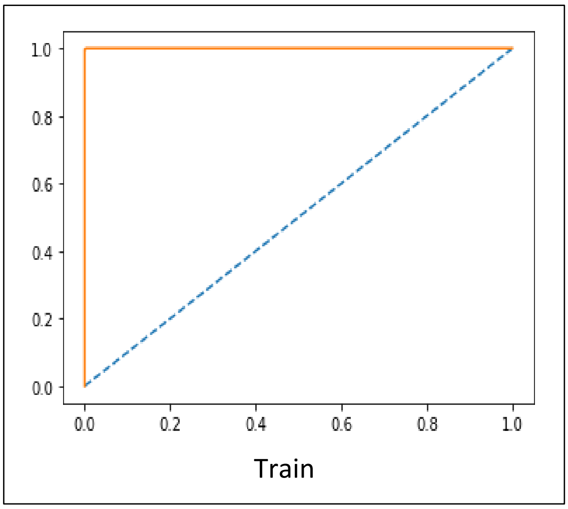
Train

Figure no: 20– Test and train accuracy report for Bagging

AUC _ROC Curve Bagging Test and train:



Test



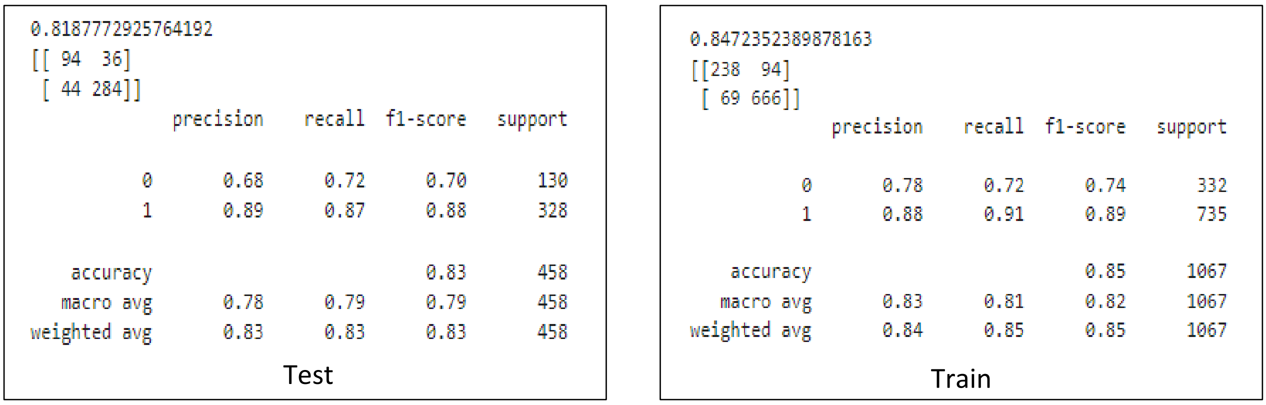
Train

Figure no: 21– AUC _ROC Curve Bagging Test and train

Boosting Test and train - Ada Boost:

- Applying Ada Boosting model and predicting the train and test,

Boosting Test and Train accuracy report (Ada Boost)



```
0.8187772925764192
[[ 94  36]
 [ 44 284]]
              precision    recall  f1-score   support

           0       0.68      0.72      0.70       130
           1       0.89      0.87      0.88       328

    accuracy                           0.83       458
   macro avg       0.78      0.79      0.79       458
weighted avg       0.83      0.83      0.83       458
```

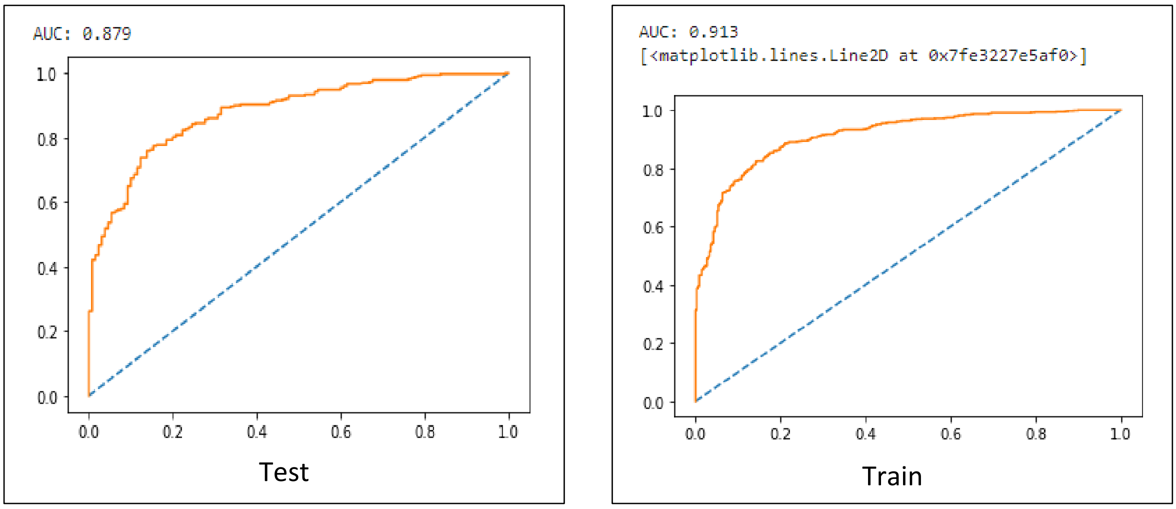
Test

```
0.8472352389878163
[[238  94]
 [ 69 666]]
              precision    recall  f1-score   support

           0       0.78      0.72      0.74       332
           1       0.88      0.91      0.89       735

    accuracy                           0.85      1067
   macro avg       0.83      0.81      0.82      1067
weighted avg       0.84      0.85      0.85      1067
```

Train

Figure no: 22– Test and train accuracy report for Boosting – Ada Boost

AUC _ROC Curve Boosting Test and train (Ada Boost):



```
AUC: 0.879
```

Test

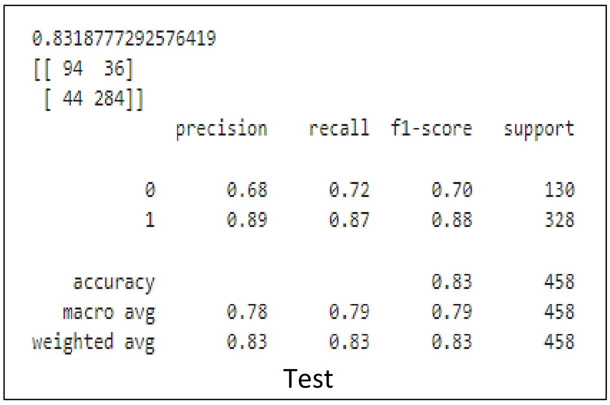
```
AUC: 0.913
[<matplotlib.lines.Line2D at 0x7fe3227e5af0>]
```

Train

Figure no: 23– AUC _ROC Curve Boosting – Ada Boost Test and train

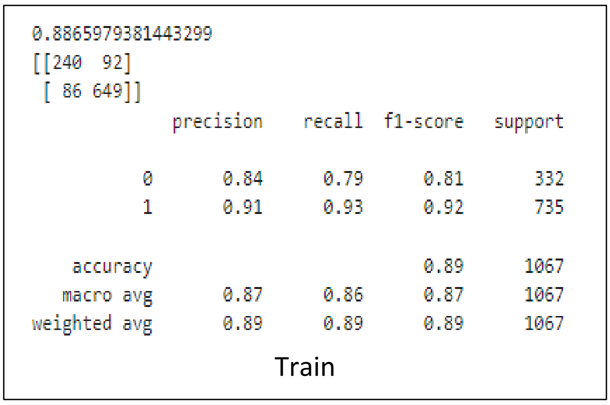
Gradient Boosting:

Boosting Test and Train accuracy report (Gradient Boost)



```
0.8318777292576419
[[ 94  36]
 [ 44 284]]
              precision    recall  f1-score   support

           0       0.68      0.72      0.70       130
           1       0.89      0.87      0.88       328

    accuracy                           0.83       458
   macro avg       0.78      0.79      0.79       458
weighted avg       0.83      0.83      0.83       458
```

Test

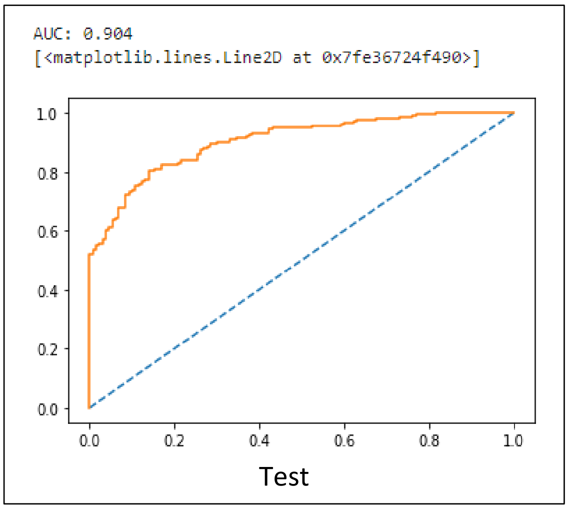


```
0.8865979381443299
[[240  92]
 [ 86 649]]
              precision    recall  f1-score   support

           0       0.84      0.79      0.81       332
           1       0.91      0.93      0.92       735

    accuracy                           0.89      1067
   macro avg       0.87      0.86      0.87      1067
weighted avg       0.89      0.89      0.89      1067
```

Train

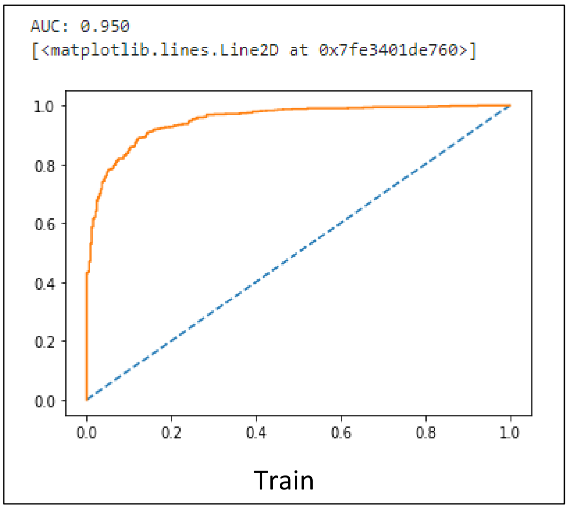
Figure no: 24– Test and train accuracy report for Boosting – Gradient Boost

AUC _ROC Curve Boosting Test and train (Gradient Boost):



```
AUC: 0.904
[<matplotlib.lines.Line2D at 0x7fe36724f490>]
```

Test



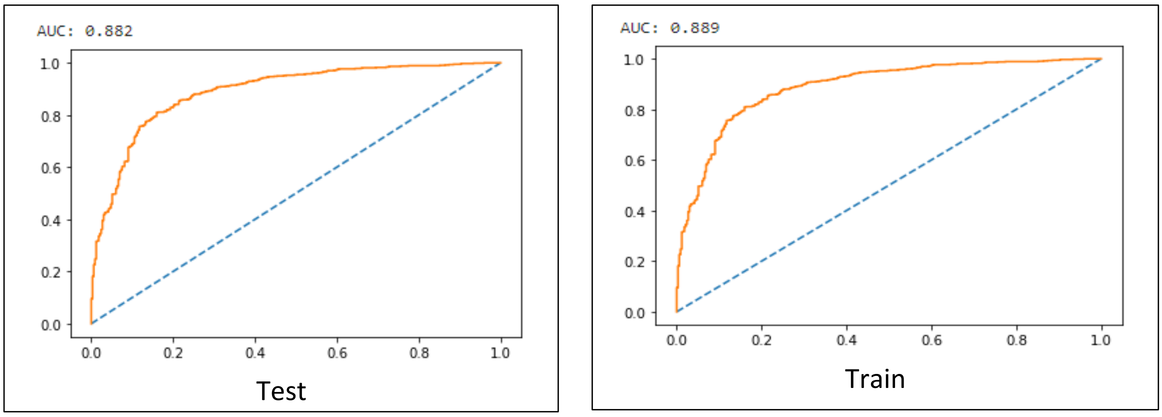
```
AUC: 0.950
[<matplotlib.lines.Line2D at 0x7fe3401de760>]
```

Train

Figure no: 25– AUC _ROC Curve Boosting – Gradient Boost Test and train

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.

AUC ROC curve for Logistic Regression Test and Train:

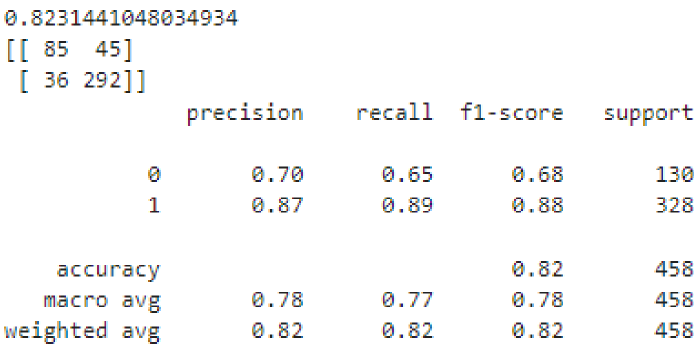


Test

Train

Figure no: 26– AUC ROC curve for Logistic Regression Test and Train

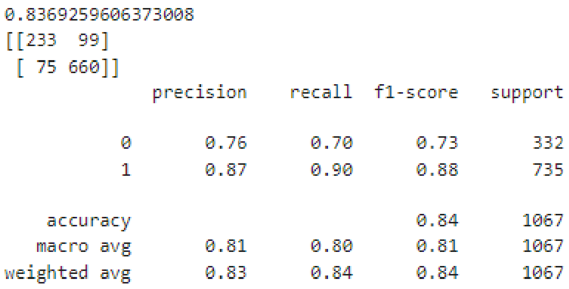
Logistic Regression confusion metrix and accuracy report:



```
0.8231441048034934
[[ 85  45]
 [ 36 292]]
              precision    recall  f1-score   support

           0       0.70      0.65      0.68       130
           1       0.87      0.89      0.88       328

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

Figure no: 27– Logistic Regression confusion metrix and accuracy report

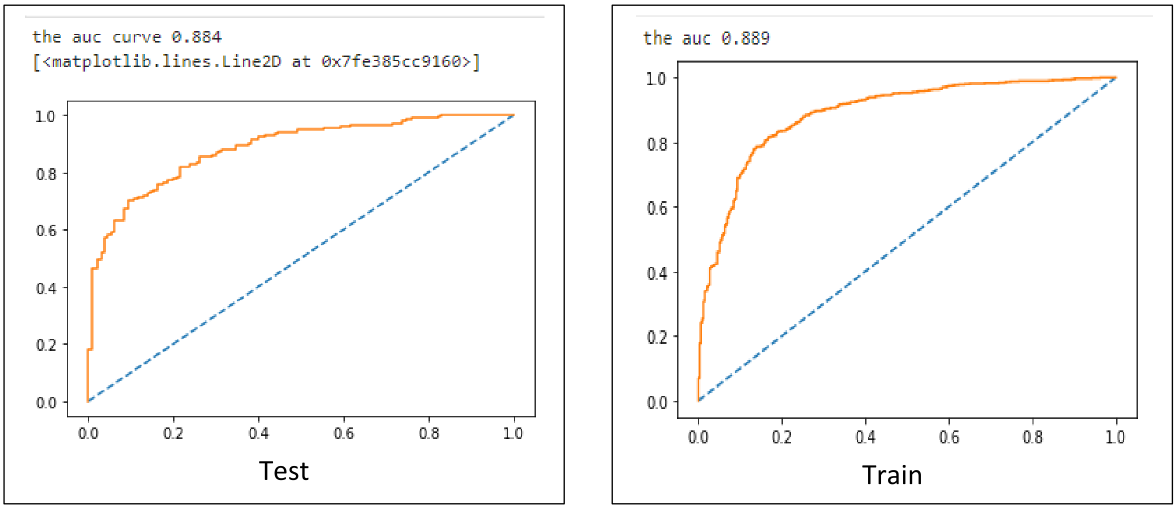
Accuracy report and confusion metrix for linear discriminant analysis:



```
0.8369259606373008
[[233  99]
 [ 75 660]]
              precision    recall  f1-score   support

           0       0.76      0.70      0.73       332
           1       0.87      0.90      0.88       735

    accuracy                           0.84      1067
   macro avg       0.81      0.80      0.81      1067
weighted avg       0.83      0.84      0.84      1067
```

Figure no: 28– Accuracy report for linear discriminant analysis

AUC ROC curve for LDA Test and Train:

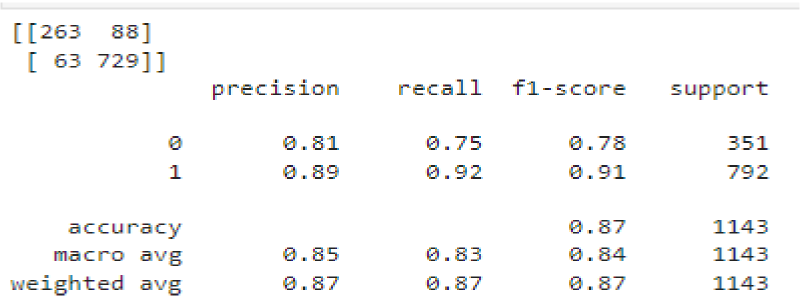


```
the auc curve 0.884
[<matplotlib.lines.Line2D at 0x7fe385cc9160>]
```

```
the auc 0.889
```

Figure no: 29– AUC ROC curve for LDA Test and Train

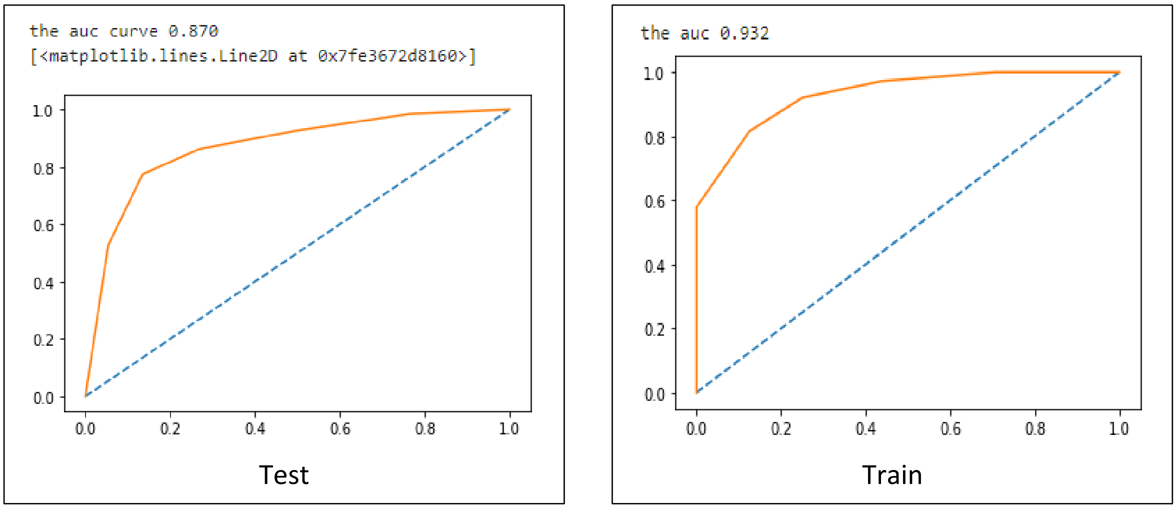
Accuracy report and confusion metrix for KNN:



```
[[263  88]
 [ 63 729]]
              precision    recall  f1-score   support

           0       0.81      0.75      0.78       351
           1       0.89      0.92      0.91       792

    accuracy                           0.87      1143
   macro avg       0.85      0.83      0.84      1143
weighted avg       0.87      0.87      0.87      1143
```

Figure no: 30– Accuracy and confusion metrix report for KNN

AUC ROC curve for KNN Test and Train:

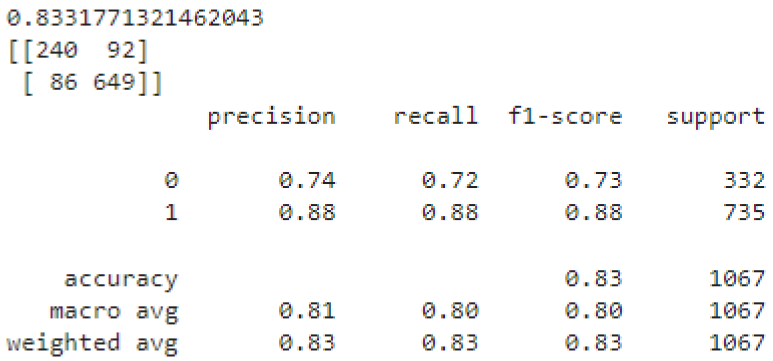


```
the auc curve 0.870
[<matplotlib.lines.Line2D at 0x7fe3672d8160>]
```

```
the auc 0.932
```

Figure no: 31– AUC ROC curve for KNN Test and Train

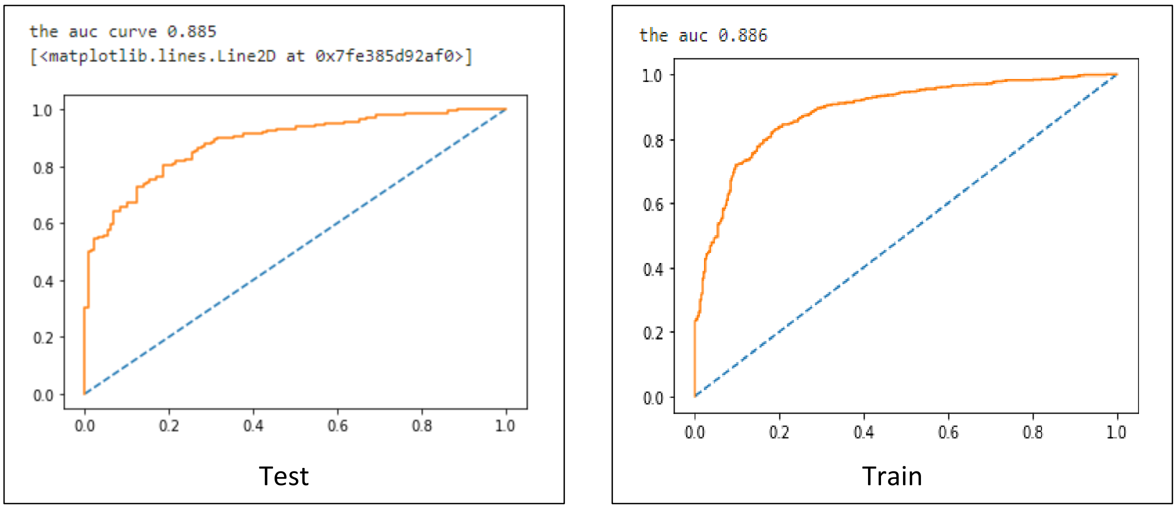
Train and Test accuracy and confusion metrix report:



```
0.8331771321462043
[[240  92]
 [ 86 649]]
              precision    recall  f1-score   support

           0       0.74      0.72      0.73       332
           1       0.88      0.88      0.88       735

    accuracy                           0.83      1067
   macro avg       0.81      0.80      0.80      1067
weighted avg       0.83      0.83      0.83      1067
```

Figure no: 32– Train and Test accuracy report and confusion metrix for Naïve Bayes

AUC ROC curve for Naive Bayes Test and Train:



```
the auc curve 0.885
[<matplotlib.lines.Line2D at 0x7fe385d92af0>]
```

Test

```
the auc 0.886
```

Train

Figure no: 33– AUC ROC curve for Naive Bayes Test and Train

Model Comparison and Best Model:

Gradient Boosting model performs the best with 89% train accuracy. And also have 91% precision and 93% recall which is better than any other models that we have performed in here with the Election dataset.

Rest all the models are more or less have same accuracy of 89%

1.8 Based on these predictions, what are the insights?

The important variable in predicting the dependent variables are

- **'Hague' and 'Blair'**

These are the ratings that the people gave to the Leaders of the 'Labour' and 'Conservative' party,

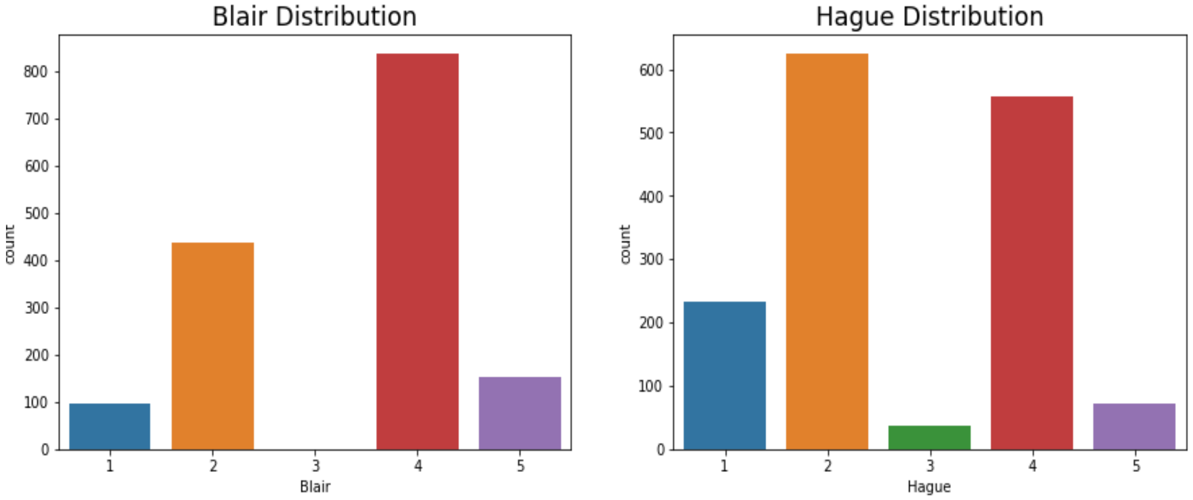


Figure no: 34– 'Hague' and 'Blair' count plot distribution

- As the frequency distribution suggests most of the people gave 4 stars to 'Blair' and there are larger number of people gave 2 stars to 'Hague' which made an impact in the dependent variable 'vote

End of Problem1

Problem 2:

Problem statement:

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

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

Domain:

Inaugural corpora

Presidents of the United States of America:

President Franklin D. Roosevelt in 1941

President John F. Kennedy in 1961

President Richard Nixon in 1973

Figure no: 35– Presidents of the United States of America

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

Number of Characters and words:

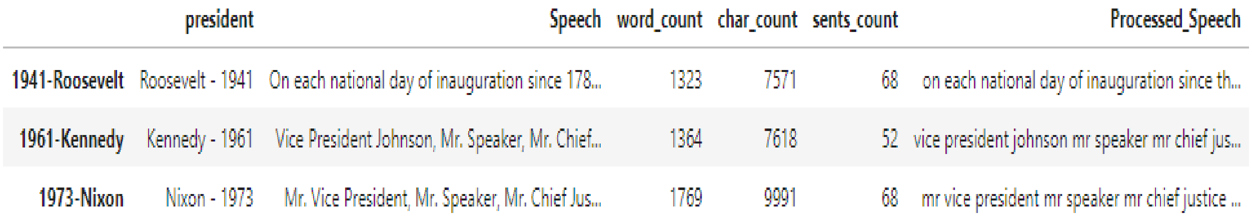
- President Franklin D. Roosevelt’s speech have **7571** Characters (including spaces) and **1360** words.
- President John F. Kennedy’s Speech have **7618** Characters (including spaces) and 1390 words.
- President Richard Nixon’s Speech have **9991** Characters (including spaces) and **1819** words.

Number of sentences:

- Number of sentence in Nixon - **68**
- Number of sentence in Kennedy - **52**
- Number of sentence in Roosevelt – **67**

2.2 Remove all the stopwords from all three speeches.

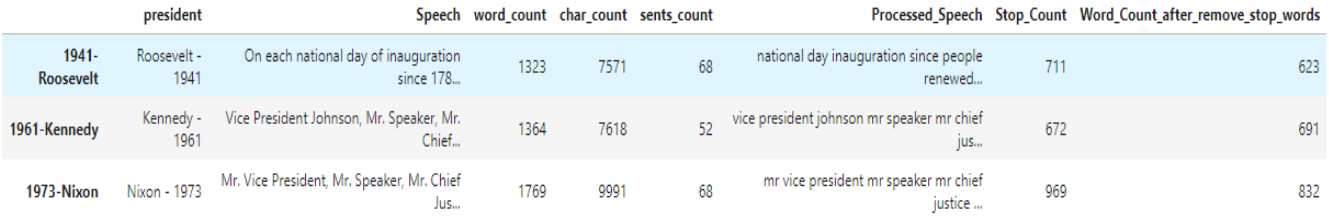
Converting all the character to lower case and removing all the punctuations.



	president	Speech	word_count	char_count	sents_count	Processed_Speech
1941-Roosevelt	Roosevelt - 1941	On each national day of inauguration since 178...	1323	7571	68	on each national day of inauguration since th...
1961-Kennedy	Kennedy - 1961	Vice President Johnson, Mr. Speaker, Mr. Chief...	1364	7618	52	vice president johnson mr speaker mr chief jus...
1973-Nixon	Nixon - 1973	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...	1769	9991	68	mr vice president mr speaker mr chief justice ...

Figure no: 36– Converted character in lower case

Counting the number of stop words and removing them.



	president	Speech	word_count	char_count	sents_count	Processed_Speech	Stop_Count	Word_Count_after_remove_stop_words
1941-Roosevelt	Roosevelt - 1941	On each national day of inauguration since 178...	1323	7571	68	national day inauguration since people renewed...	711	623
1961-Kennedy	Kennedy - 1961	Vice President Johnson, Mr. Speaker, Mr. Chief...	1364	7618	52	vice president johnson mr speaker mr chief jus...	672	691
1973-Nixon	Nixon - 1973	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...	1769	9991	68	mr vice president mr speaker mr chief justice ...	969	832

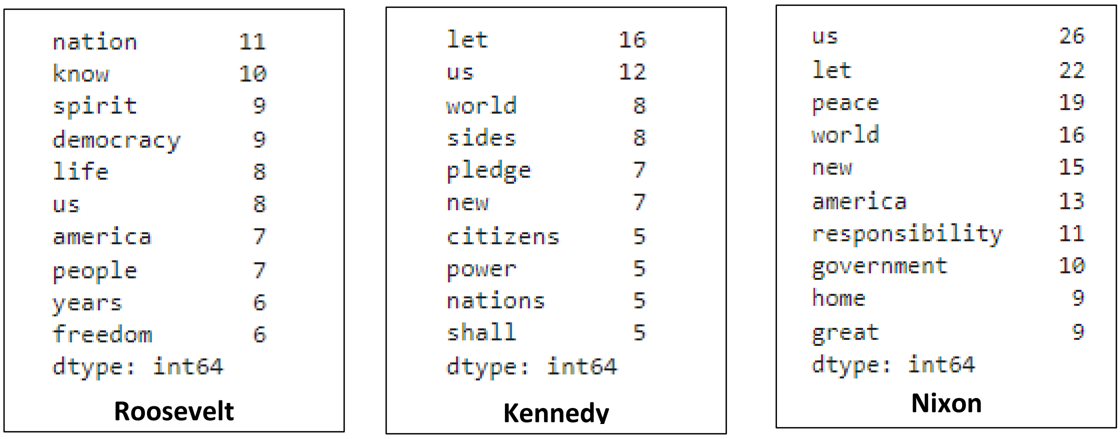
Figure no: 37– Counting the number of stop words and removing them

Inference:

All the stop words have been removed from all the three speeches.

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 stopwords)

In the below snippets we could see the words that occurred most number of times in their inaugural address.



```
nation        11
know          10
spirit         9
democracy      9
life           8
us             8
america        7
people         7
years          6
freedom        6
dtype: int64
```

Roosevelt

```
let          16
us           12
world         8
sides         8
pledge        7
new           7
citizens      5
power         5
nations       5
shall         5
dtype: int64
```

Kennedy

```
us                26
let               22
peace             19
world             16
new               15
america           13
responsibility    11
government        10
home               9
great              9
dtype: int64
```

Nixon

Figure no: 38– words that occurred most number of times

Top three words that occurs more times:

President Franklin D. Roosevelt's speech are

- nation
- know
- spirit

President John F. Kennedy's Speech are

- let
- us
- world

President Richard Nixon's Speech are

- us
- let
- peace

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

Word Cloud for President Franklin D. Roosevelt's speech (after cleaning)!!

Word Cloud for Roosewelt after cleaning

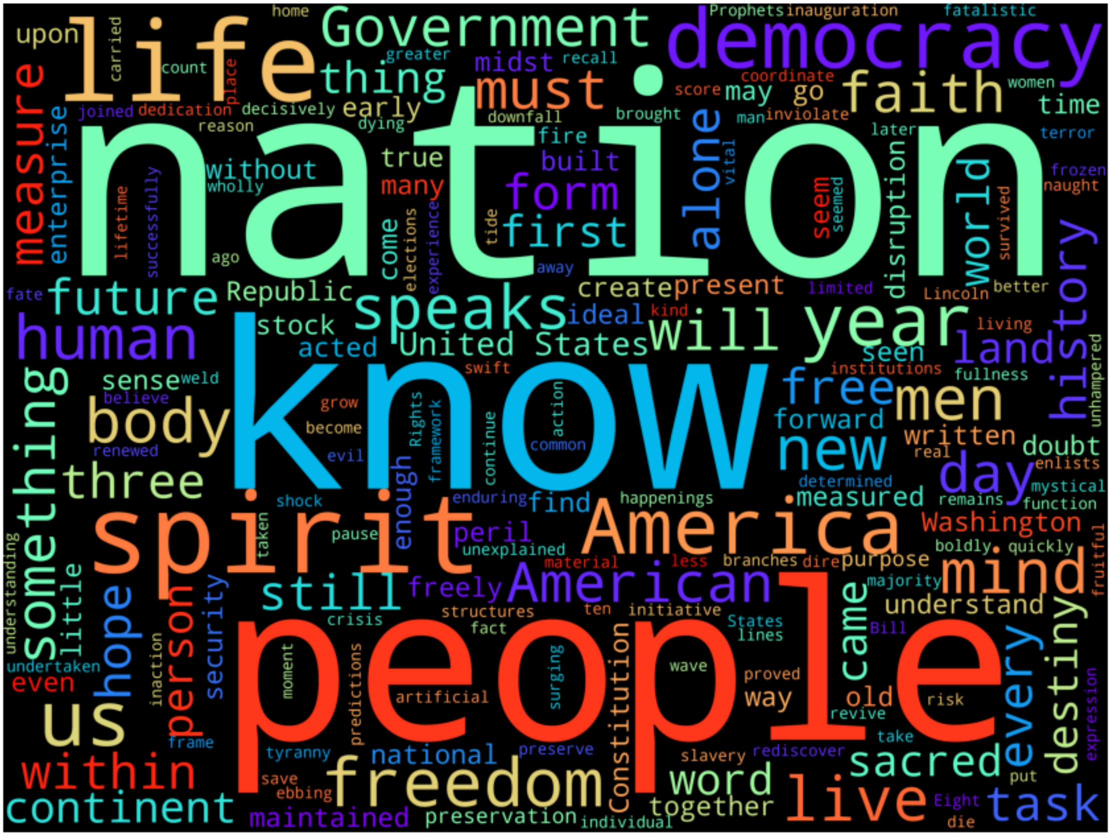


Figure no: 39– Word Cloud for President Franklin D. Roosevelt's speech (after cleaning)!!

Word Cloud for President John F. Kennedy's Speech (after cleaning)!!

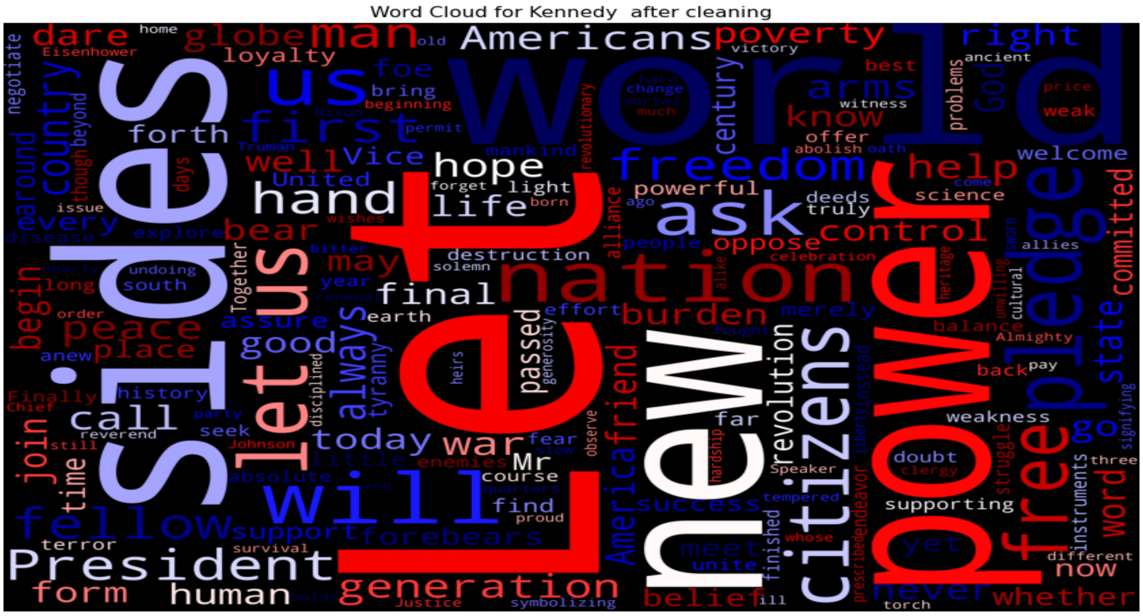


Figure no: 40– Word Cloud for President John F. Kennedy's Speech (after cleaning)!!

Word Cloud for President Richard Nixon's Speech (after cleaning)!!

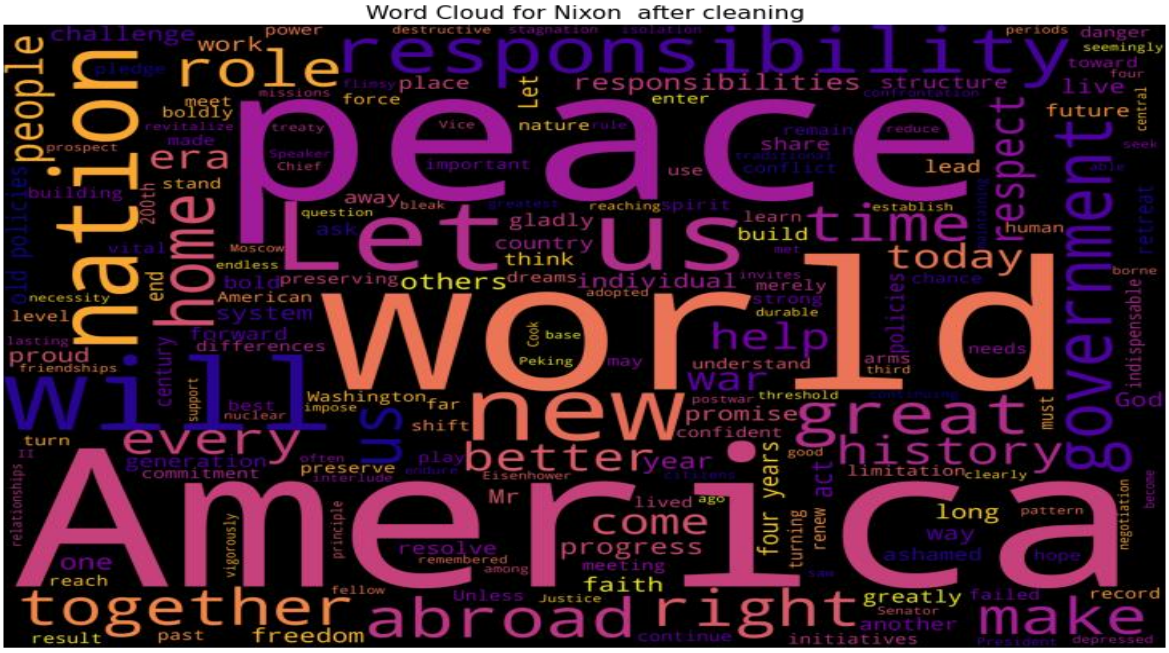


Figure no: 41– Word Cloud for President Richard Nixon's Speech (after cleaning)!!

End of Problem2