MACHINE LEARNING REPORT

KNN, Ensemble Techniques, Text Mining

Contents

List of Figures:

List of Tables:

Problem 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 the overall win and seats covered by a particular party.

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.

Read Data first 5 rows:

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 1. Read Data first 5 row.

Data information:

```
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(8), object(2)
memory usage: 119.3+ KB
```

Interpretation:

- ***We have dropped 'Unnamed: 0' is a variable that simply represents the index.***
- ***The data has 1525 rows and 9 columns with Float as the data type for each.***
- ***We have no non-null data with 9 variables. Ordinal 7 variables and 2 object types.***

Describe the data: Numeric data:

	Count	**Mean**	**STD**	**MIN**	**25.00%**	**50.00%**	**75.00%**	**MAX**
Age	1525	54.182295	15.711209	24	41	53	67	93
Economic.cond. national	1525	3.245902	0.880969	1	3	3	4	5
Economic.cond. household	1525	3.140328	0.929951	1	3	3	4	5
Blair	1525	3.334426	1.174824	1	2	4	4	5
Hague	1525	2.746885	1.230703	1	2	2	4	5
Europe	1525	6.728525	3.297538	1	4	6	10	11
Political.knowledge	1525	1.542295	1.083315	0	0	2	2	3

Table2. Describe the data Numeric data

- ***The summary stats: average age is approximate 50-60.***
- ***The current national economic condition average is 3.***
- ***The current national economic condition average is 3.***
- ***The assessment of people on Labour leader is 3 and above on a scale of 1-5.***
- ***The assessment of people on Labour leader is 2 and above on a scale of 1-5.***
- ***6-point that is above 50% is the power of opposition towards the European Union.***

- ***The Political Knowledge of 50% of people is 2 and above on a scale of 1-3.***

Describe the data: Categorical data:

	Count	Unique	Top	Freq
Vote	1525	2	Labour	1063
Gender	1525	2	female	812

Table 3. 5-Point summary stat of Categorical data

- ***Vote is our Dependent variable:***
  - ***We see the majority of voters is 'Labour' and its frequency is 1057.***
- ***Most voters are females: 808.***
- ***There are 8 duplicate rows. I will be dropping them, there is a possibility that the data would be from different customers.***
- ***The data shape is now:*** `(1517, 9)`.

Skewness:

age	***0.139800***
economic.cond.national	***-0.238474***
economic.cond.household	***-0.144148***
Blair	***-0.539514***
Hague	***0.146191***
Europe	***-0.141891***
political.knowledge	***-0.422928***

dtype: float64

- ***Skewness refers to a distortion or asymmetry that deviates from the mean.***
- ***Two variables are positively skewed and the rest negatively skewed.***

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 plot (histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Outliers proportion should be discussed, and inferences from the 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.

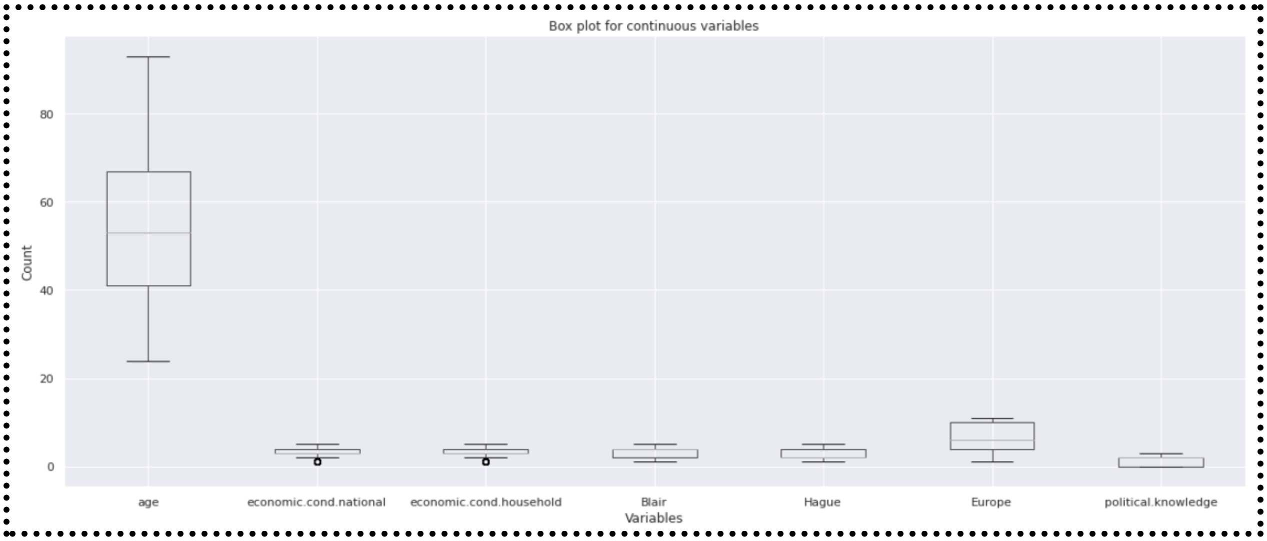
Pre_check_eda Table:

	Null values	Data types
vote	0	object
age	0	int64
economic.cond.national	0	int64
economic.cond.household	0	int64
Blair	0	int64
Hague	0	int64
Europe	0	int64
political.knowledge	0	int64
gender	0	object

Table 4. Pre_check_eda Table

- ***The Shape of the data:***
  - ***Rows: 1517***
  - ***Columns: 9***

Outliers: All the four continuous variables have outliers:



- ***Would not be treating the outliers for this case, as they are on ordinal variables.***
- ***Only two variables have***
  - **`Economic.cond.national`**
  - **`Economic.cond.household`**
- ***Value counts of ordinal Variables***
  - ***ECONOMIC.COND.NATIONAL : 5***

```
1    37
5    82
2   256
4   538
3   604
Name: economic.cond.national, dtype: int64
```

  - ***ECONOMIC.COND.HOUSEHOLD : 5***

```
1    65
5    92
2   280
4   435
3   645
Name: economic.cond.household, dtype: int64
```

  - ***BLAIR : 5***

```
3     1
1    97
5   152
2   434
4   833
Name: Blair, dtype: int64
```

  - ***HAGUE : 5***

```
3    37
5    73
1   233
4   557
```

```
    2    617
Name: Hague, dtype: int64
```

- ***EUROPE : 11***

```
    2      77
    7      86
    10    101
    1     109
    9     111
    8     111
    5     123
    4     126
    3     128
    6     207
    11    338
Name: Europe, dtype: int64
```

- ***POLITICAL.KNOWLEDGE : 4***

```
    1     38
    3    249
    0    454
    2    776
Name: political.knowledge, dtype: int64
```

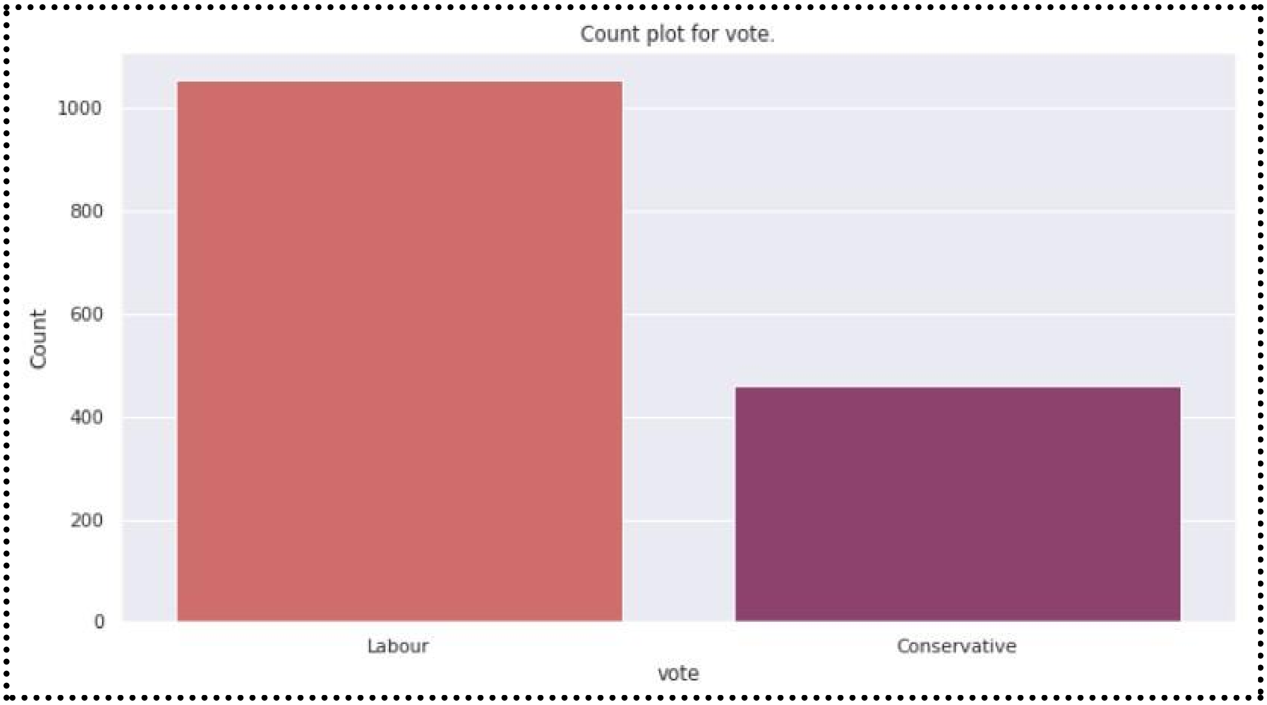
- ***Value counts of Categorical Variables***
  - ***VOTE : 2***

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

  - ***GENDER : 2***

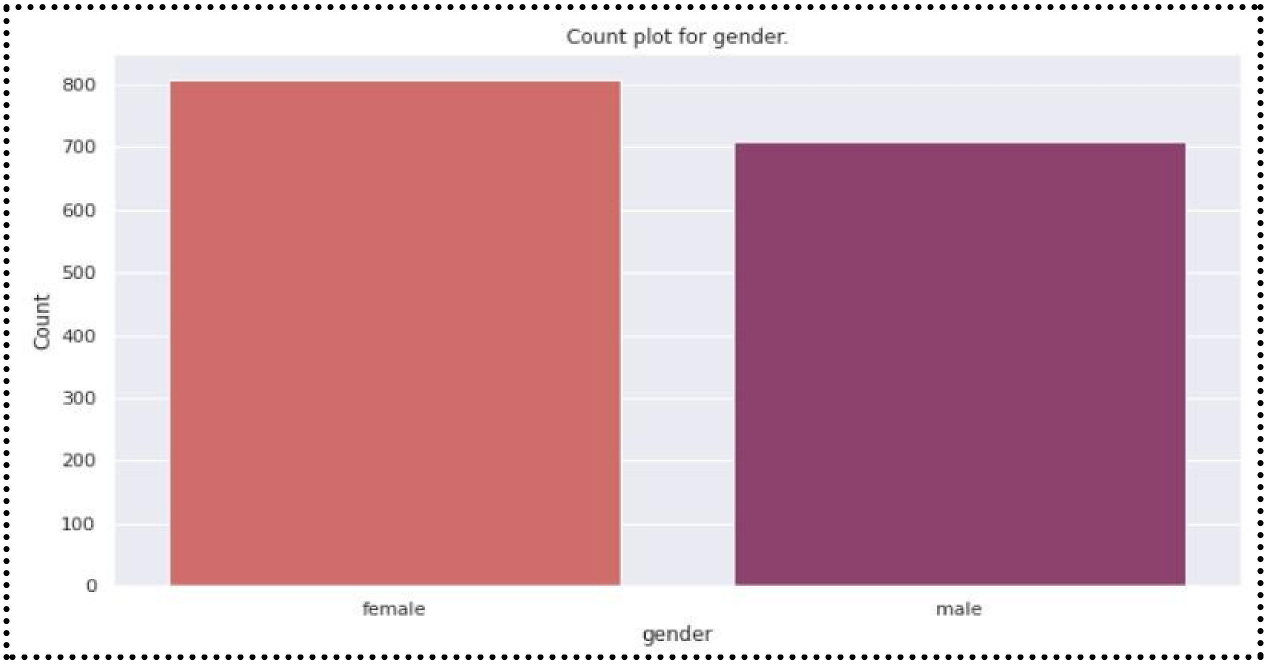
```
            male      709
            female    808
            Name: gender, dtype: int64
```

EDA - Univariate Analysis: Categorical variables:



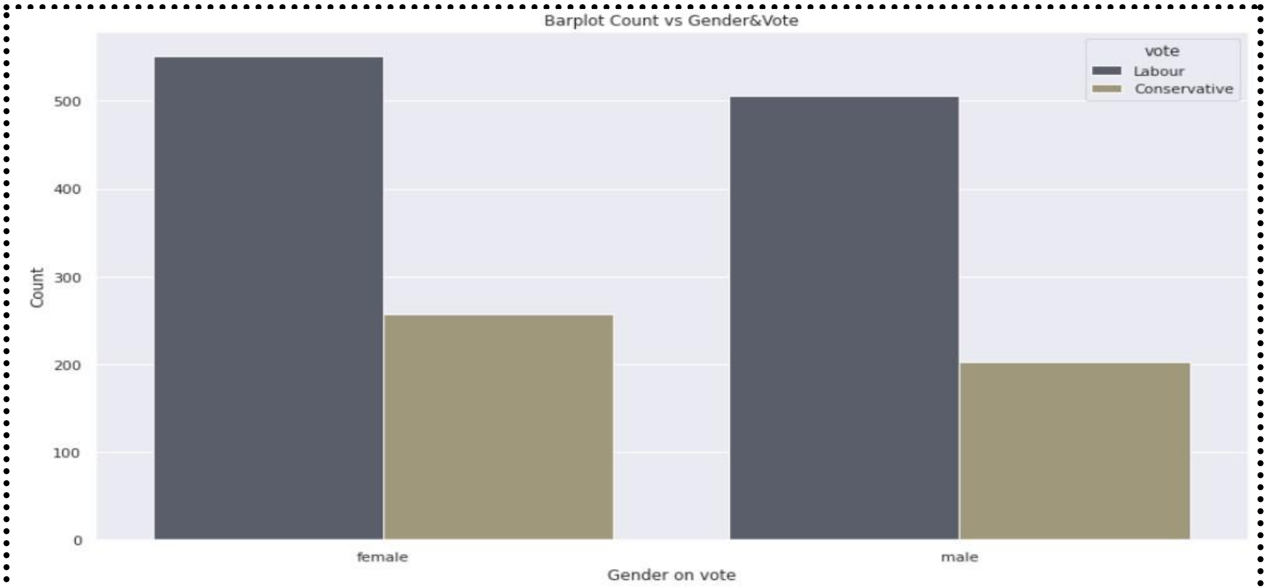
- ➢ ***From the above countplot it is clear that the Labour votes are above 50% more compared to Conservative.***

Gender:



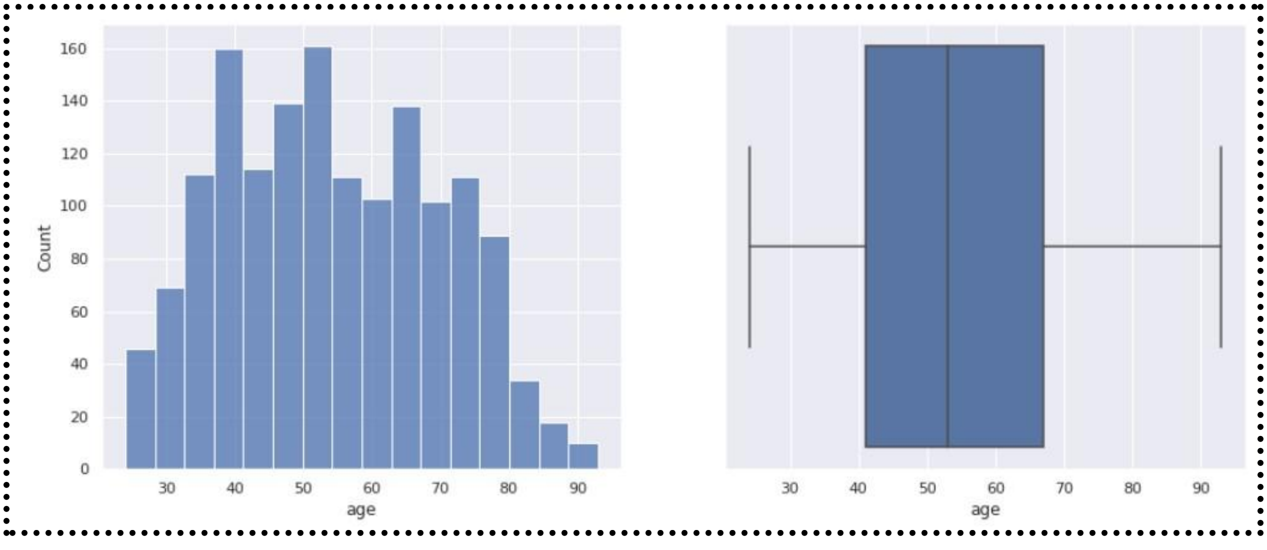
- ➢ ***From the above countplot it is clear that the Female voters are above 50% more compared to Male voters.***

Bivariate Analysis: Categorical variables:



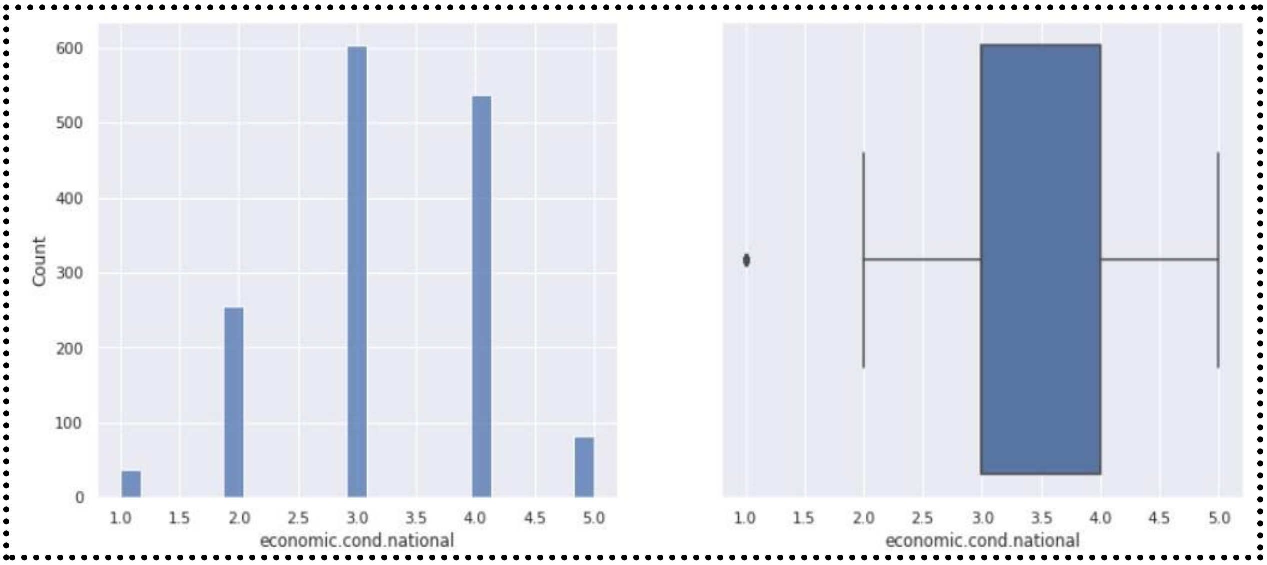
- ❖ ***Based on the figure we can see that both Labour and Conservative have Female voters more than males.***

Numerical Variable: Univariate Analysis
Dist and boxplot of all variables: Age



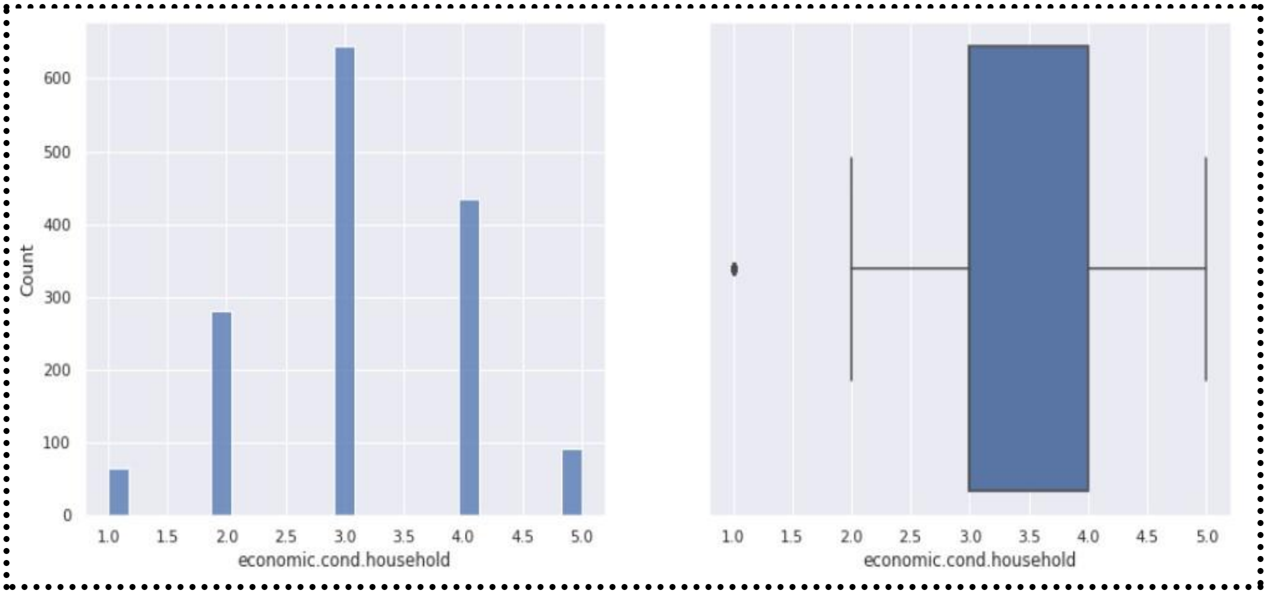
- ➢ ***The Boxplot tells us there are no outliers Age distribution.***
- ➢ ***The distplot distribution can be said to be a mostly normal distribution. Skewness(age) is 0.140. The distribution ranges between 24 to 90.***

Economic.cond.national



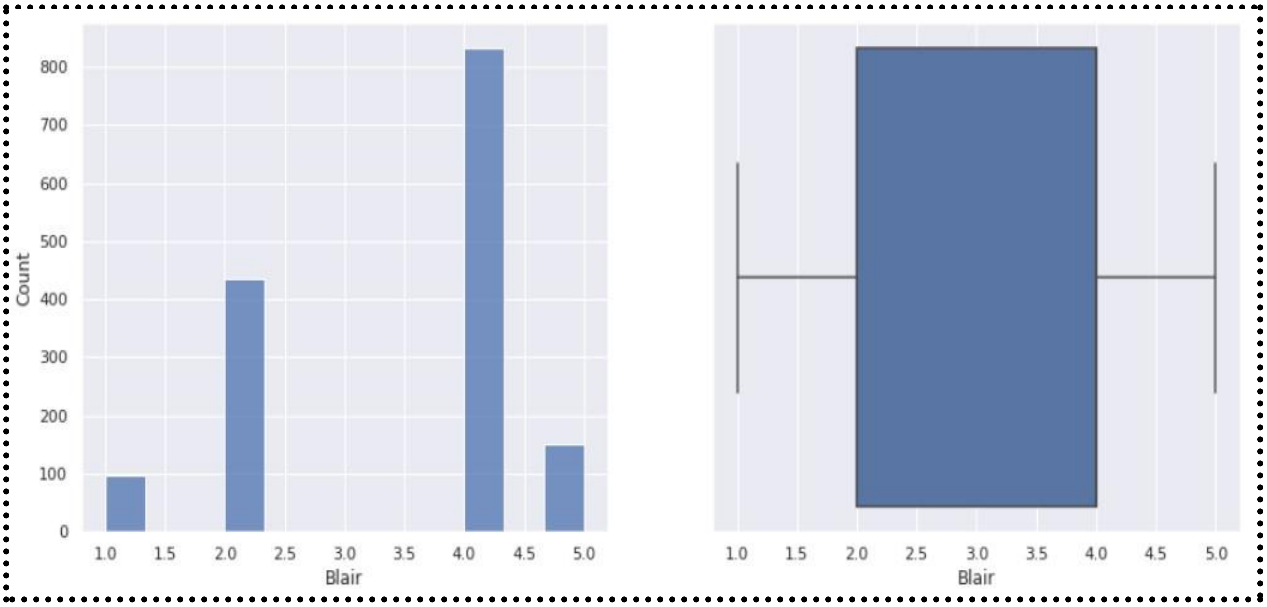
- ➢ ***The Boxplot tells us there are few outliers in Economic.cond.national distribution.***
- ➢ ***The distplot distribution can be said to be slightly right skewed. Skewness() is negative -0.238. The distribution ranges between 1 to 5.***

Economic.cond.household



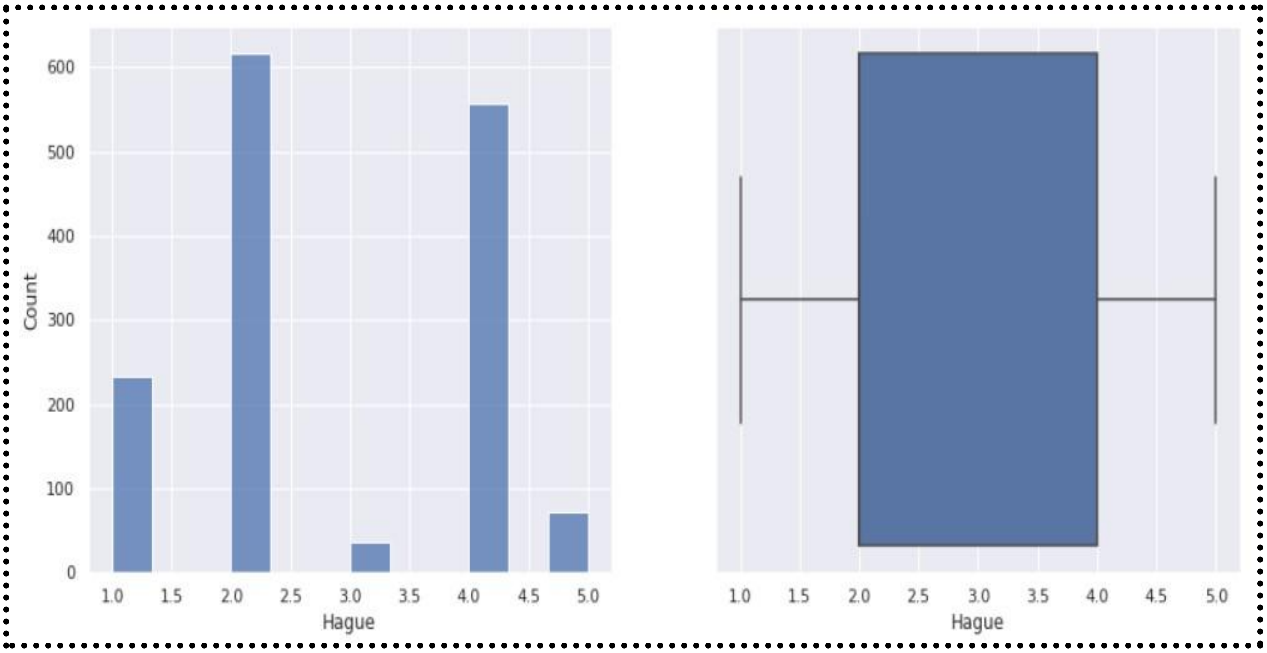
- ➢ ***The Boxplot tells us there are few outliers for Economic.cond.household distribution.***
- ➢ ***The distplot distribution can be said to be slightly right skewed. Skewness () is -0.144. The distribution ranges between 1 to 5.***

Blair



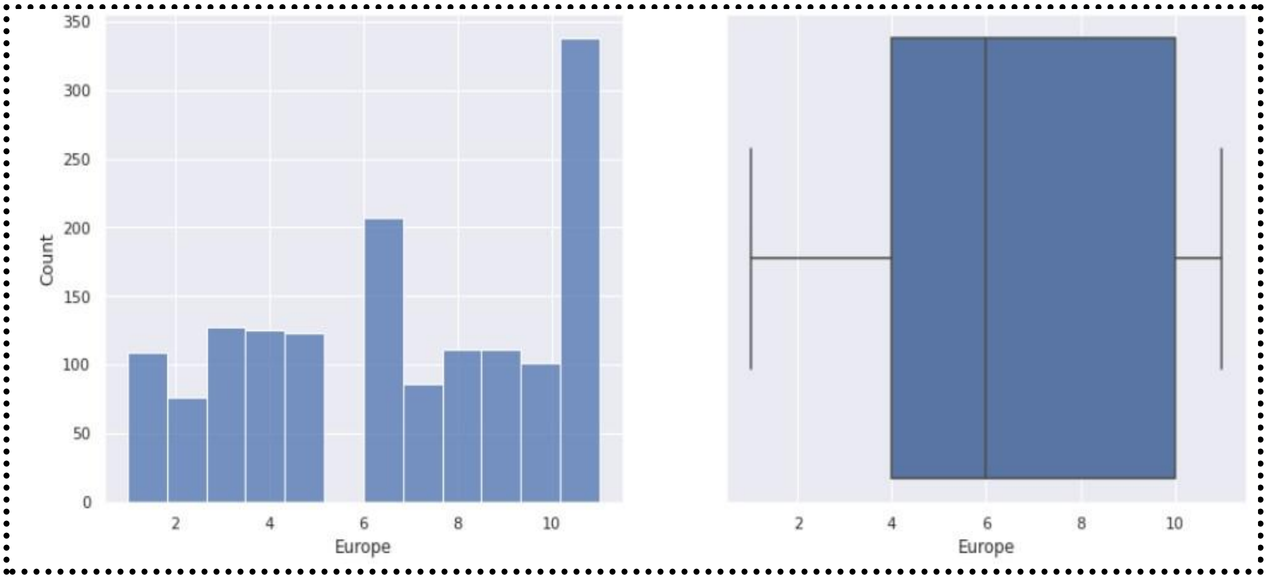
- ➢ ***The Boxplot tells us there are no outliers for Blair distribution.***
- ➢ ***The distplot distribution can be said to be slightly right-skewed. Skewness () is -0.540. The distribution ranges between 1 to 5.***

Hague



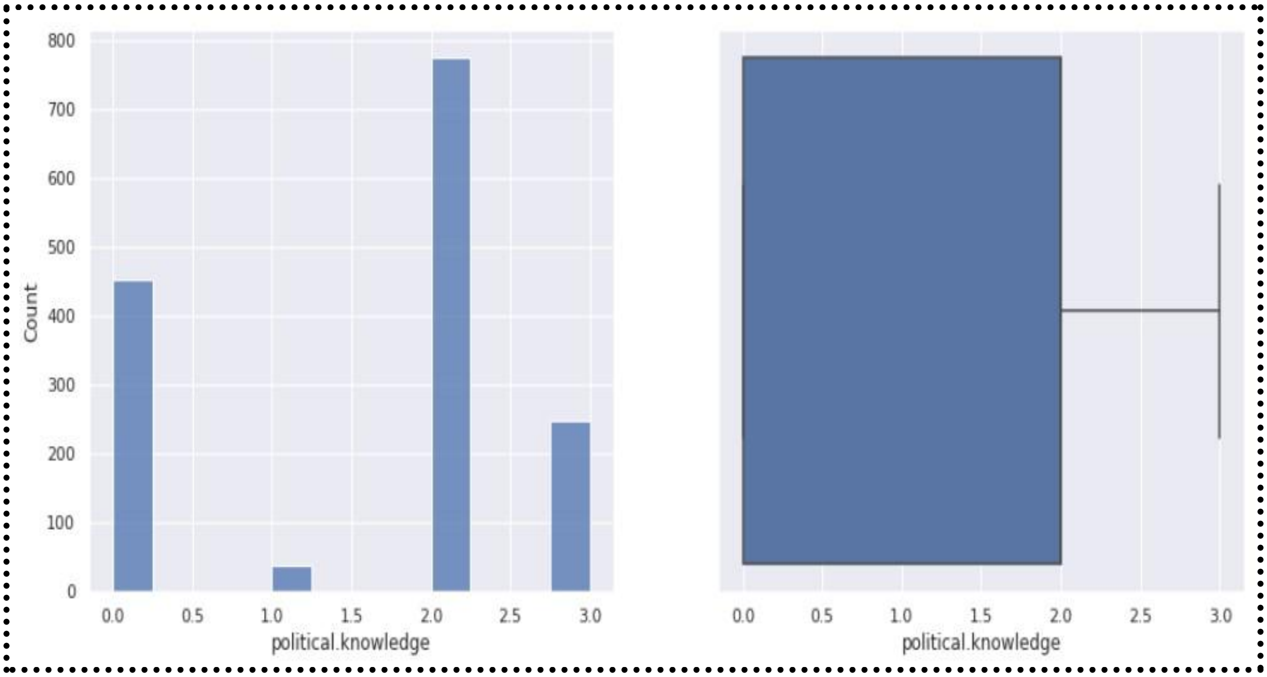
- ➢ ***The Boxplot tells us there are no outliers for Hague distribution.***
- ➢ ***The distribution can be said to be slightly right-skewed. Skewness () is 0.145. The distribution ranges between 1 to 5.***

Europe



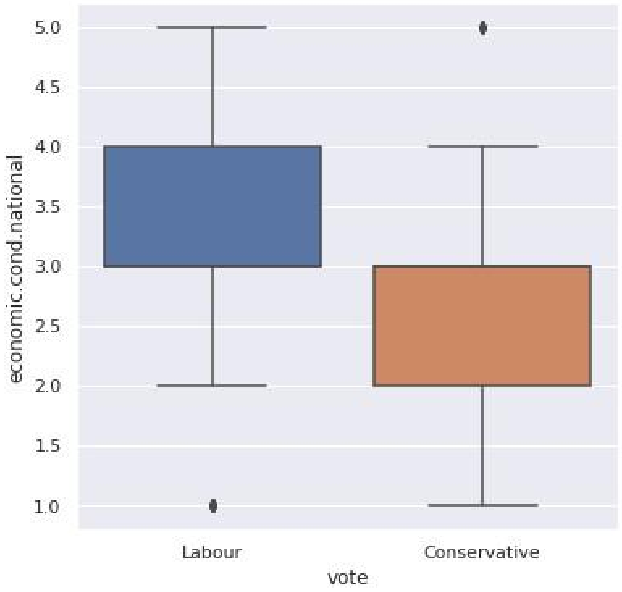
- ➢ ***The Boxplot tells us there are no outliers for Europe distribution.***
- ➢ ***The distribution can be said to be right-skewed. Skewness() is 0.142. The distribution ranges between 1 to 11.***

Political Knowledge

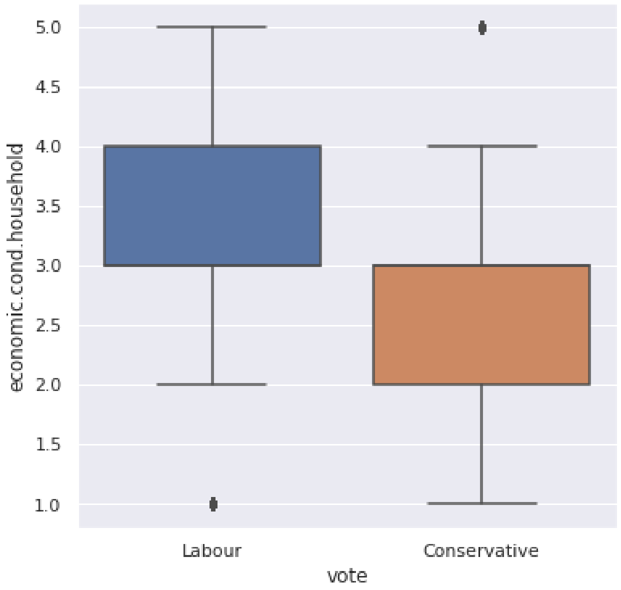


- ➢ ***The Boxplot tells us there are no outliers for Political Knowledge distribution.***
- ➢ ***The distribution can be said to be right-skewed. Skewness() is 0.142. The distribution ranges between 1 to 11.***

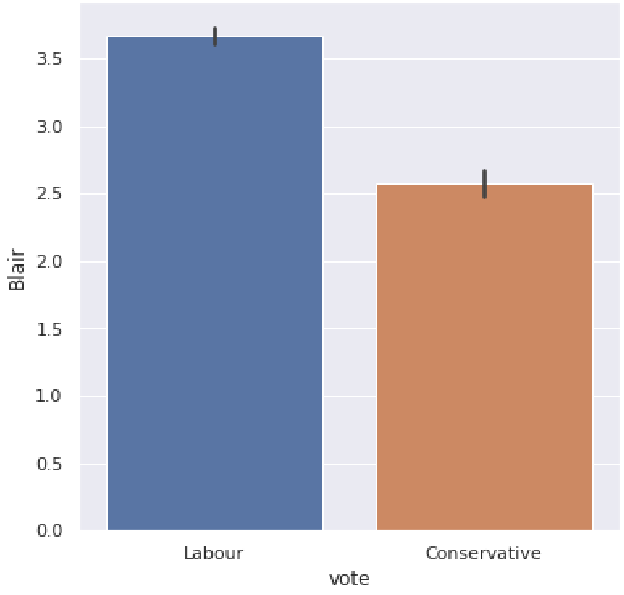
Bivariate Analysis:



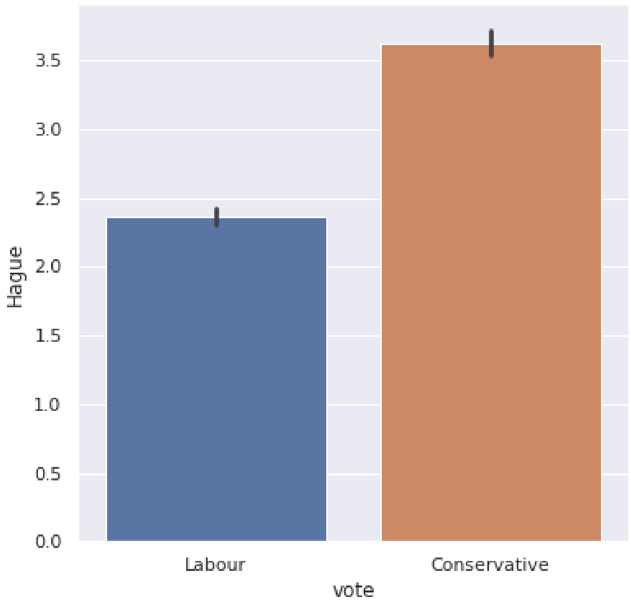
- ***We can see based on the assessment of current economic.cond.national the voters prefer Labour.***



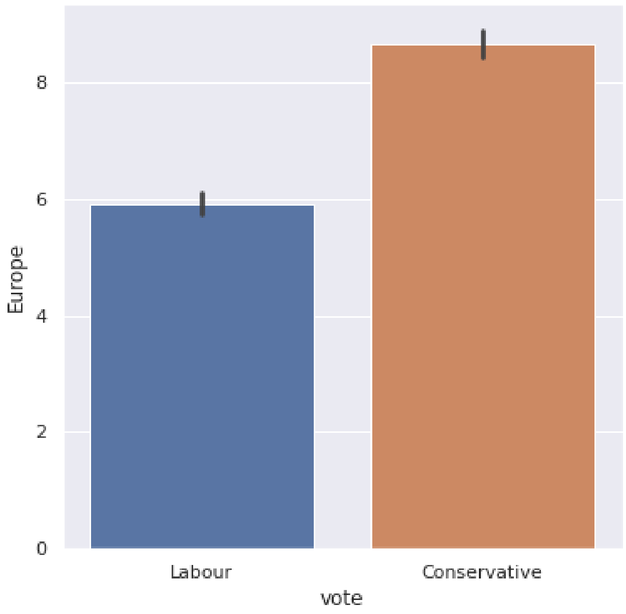
- ***We can see based on the assessment of current economic.cond.household the voters prefer Labour.***



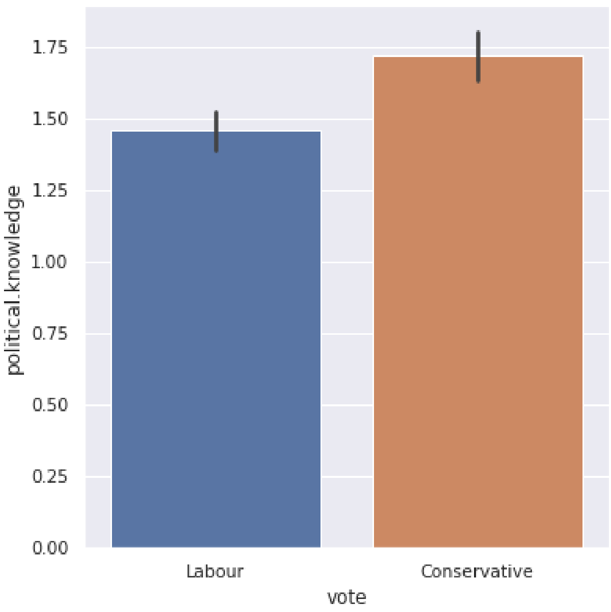
- ***Based on the assessment of Labour leader Blair, the voters prefer Labour.***



- ***Based on the assessment of Conservative leader Hague, the voters prefer the Conservative party.***

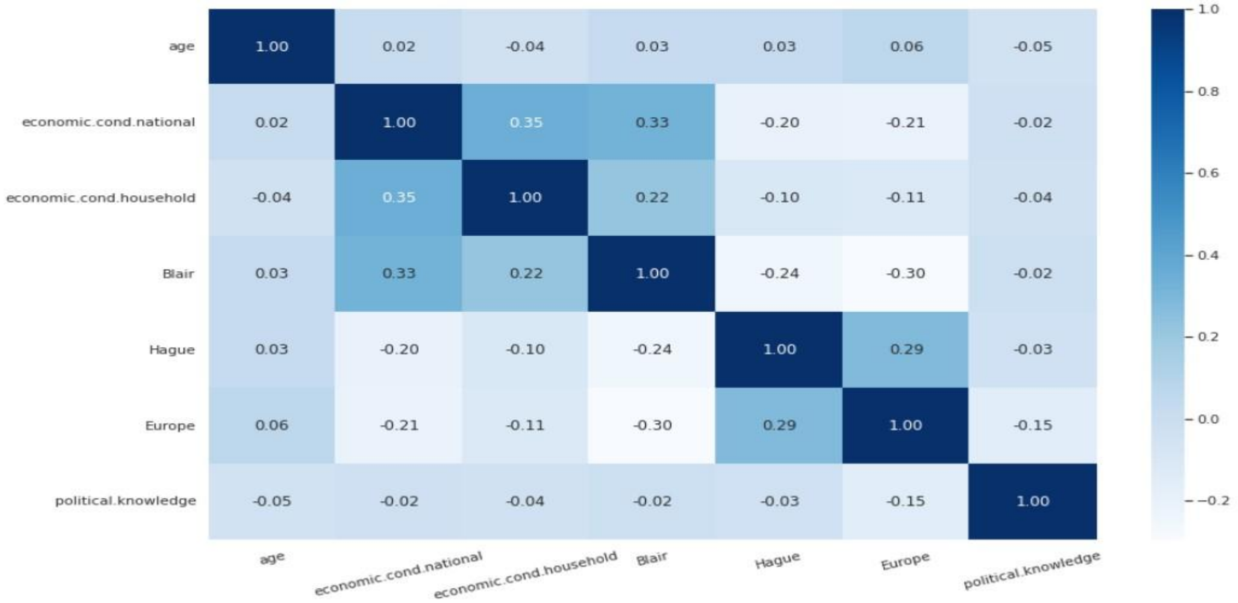


❖ ***Based on the Voter's view on European integration the more they oppose it, the voters prefer the Conservative party.***



❖ ***Based on the Voter's Knowledge of European integration, the voters prefer the Conservative party.***

Multivariate Analysis - Correlation Heatmap:



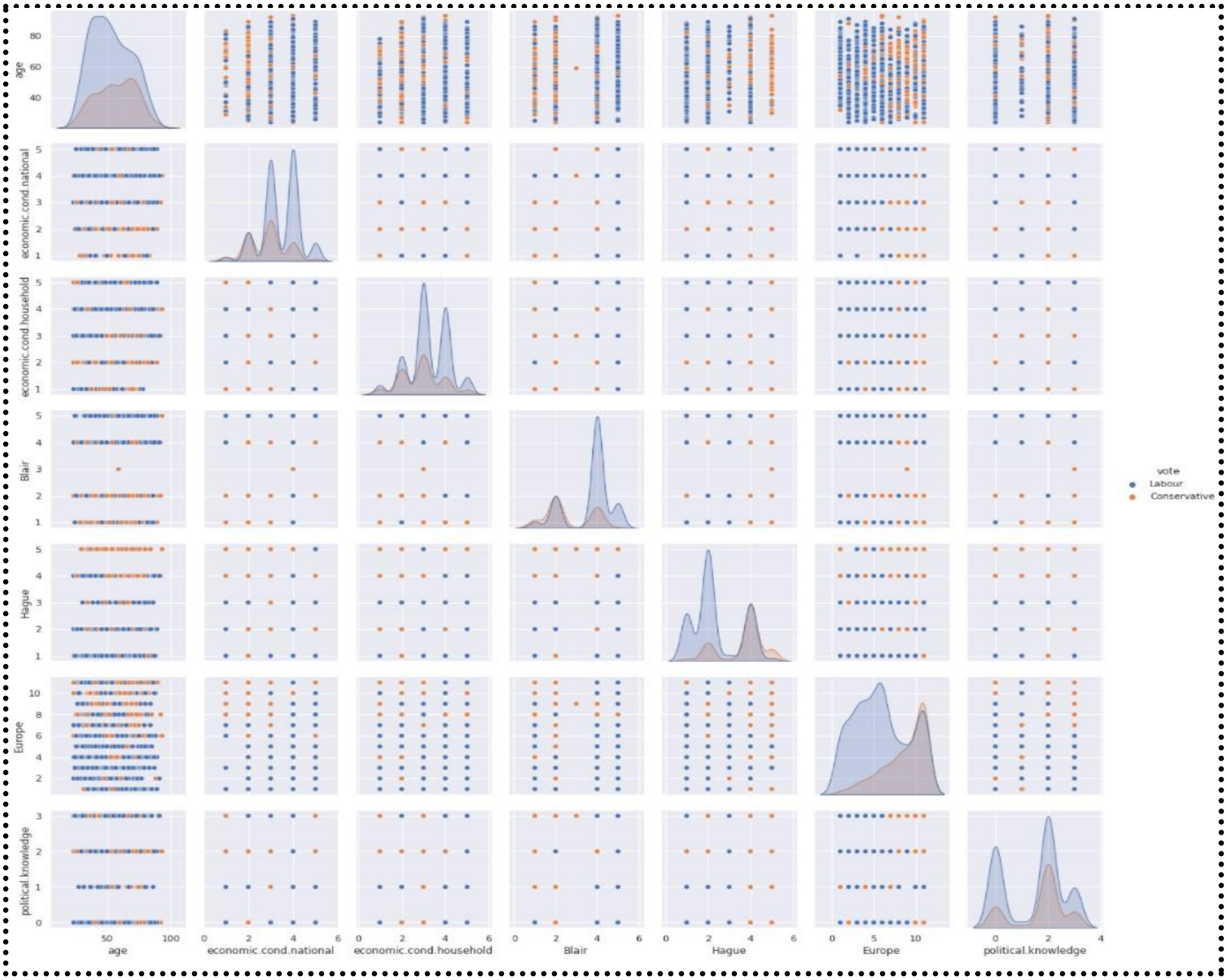
- ➢ ***The relation between pairs of numeric&ordinal variables is given by the heatmap.***
- ➢ ***The correlation between the following variables are positive: (variable are directly proportional)***

- ***Economic.cond.national and Economic.cond.household***
- ***Blair and Economic.cond.national***
- ***Blair and Economic.cond.household***
- ***Spending and Current Balance***

➢ ***The correlation between the following variables are negative: (variable are inversely proportional)***

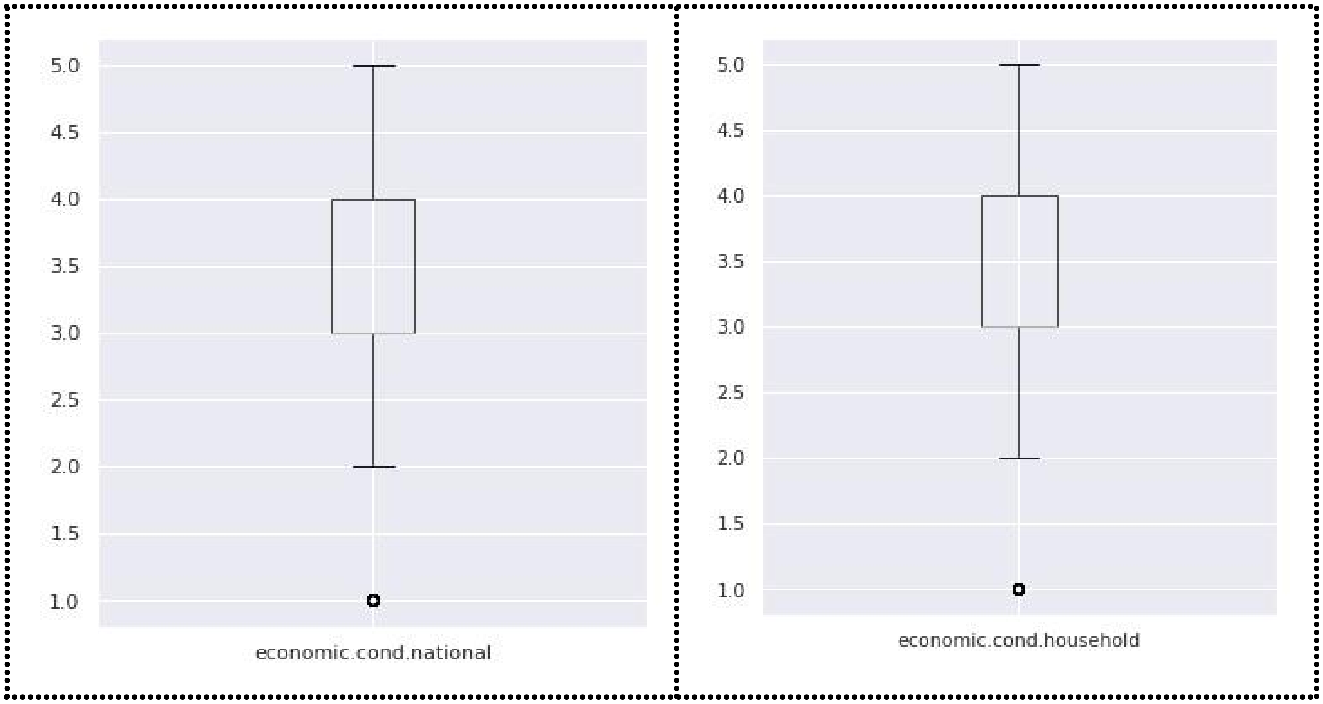
- ***Blair and Europe***
- ***Blair and Hague***
- ***Economic.cond.national and Hague***
- ***Economic.cond.national and Europe***

Pair Plot:



- ***We can see the same relation pattern(correlation) between the variables in the Pair Plot above with Two parties as hue.***
- ***Blue is Labour and Organe is Conservatives.***
- ***A voter with a lower assessment of the labour leader votes for the Conservative party and wise versa.***
- ***A voter that is aware of current household and national economic votes for the Labour party.***

Economic.cond.national and Economic.cond.household with Outliers:



- ***As we have few outliers in Economic.cond.national and Economic.cond.household.***

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 basis of appropriate measures 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.

Variables Encoded using dummy variable encoding:

age	Economic Cond National	Economic Cond Household	Blair	Hague	Europe	Political Knowledge	Labour =1 Conservative=0	Male=1 Female =0
43	3	3	4	1	2	2	1	0
36	4	4	4	4	5	2	1	1
35	4	4	5	2	3	2	1	1
24	4	2	2	1	4	0	1	0
41	2	2	1	1	6	2	1	1

Table5. Variables Encoded using dummy variable encoding

- ***All the variables are now encoded and ready for model implementation.***
- ***We have an object data type, which we will convert to categorical.***
- ***Data info after conversion:***

```
#  Column                    Non-Null Count   Dtype
---  ------                  --------------   -----
0  age                       1517 non-null    int64
1  economic.cond.national    1517 non-null    int64
2  economic.cond.household   1517 non-null    int64
3  Blair                     1517 non-null    int64
4  Hague                     1517 non-null    int64
5  Europe                    1517 non-null    int64
6  political.knowledge       1517 non-null    int64
7  Labour=1_Conservative=0   1517 non-null    uint8
8  Male=1_Female=0           1517 non-null    uint8
dtypes: int64(7), uint8(2)
memory usage: 130.1 KB
```

Describe the data: ***Summary***

	Count	Mean	STD	MIN	25.00%	50.00%	75.00%	MAX
age	1517	54.24	15.7	24	41	53	67	93
Economic.cond.national	1517	3.25	0.88	1	3	3	4	5
Economic.cond.household	1517	3.14	0.93	1	3	3	4	5
Blair	1517	3.34	1.17	1	2	4	4	5
Hague	1517	2.75	1.23	1	2	2	4	5
Europe	1517	6.74	3.3	1	4	6	10	11
Political.knowledge	1517	1.54	1.08	0	0	2	2	3
Labour=1 Conservative=0	1517	0.7	0.46	0	0	1	1	1
Male=1_Female=0	1517	0.47	0.5	0	0	0	1	1

Table 6. Describe the data: Summary

- ***We have variables with different scales, only Age is a continuous variable.***
- ***Scaling is used to eliminate redundancy and can be done on ordinal and continuous variable, in this case all except for: Labour=1_Conservative=0 and Male=1_Female=0.***
- ***Our variables vary in dimension (having different weight), which would lead some variables to have more weightage on the outcome than others.***
- ***For example, if we look at our data summary we see:***
  - ***Standard Deviation(std) of 'Age' is the highest(15.7) followed by 'Europe', (3.3) when compared to others.***
- ***Same if we look at the variance:***

➢ ***age***	***246.38***
➢ ***economic.cond.national***	***0.78***
➢ ***economic.cond.household***	***0.87***
➢ ***Blair***	***1.38***
➢ ***Hague***	***1.52***
➢ ***Europe***	***10.88***

➢ *political.knowledge*	***1.18***
➢ *Labour=1_Conservative=0*	***0.21***
➢ *Male=1_Female=0*	***0.25***

- ***As we do have large variance data, we do scale our variables. So as to avoid(creating a bias). Can be done by:***
  - ***Z-score method***
  - ***Min-Max method***
- *Using* **Min-Max** *method method to scale the 7 variables here:*
- ***Min-Max method has now brought the data closer and decreased the variance between them: We can see the change in std and variance after scaling. (between 0 & 1)***
- ***We look at the variance after scaling:***

➢ *age*	***0.05***
➢ *economic.cond.national*	***0.05***
➢ *economic.cond.household*	***0.05***
➢ *Blair*	***0.09***
➢ *Hague*	***0.09***
➢ *Europe*	***0.11***
➢ *political.knowledge*	***0.13***
➢ *Labour=1_Conservative=0*	***0.21***
➢ *Male=1_Female=0*	***0.25***

5-Point summary stat of scaled data:

	Count	Mean	STD	MIN	25.00%	50.00%	75.00%	MAX
age	1517	0.44	0.23	0	0.25	0.42	0.62	1
Economic.cond.national	1517	0.56	0.22	0	0.5	0.5	0.75	1
Economic.cond.household	1517	0.53	0.23	0	0.5	0.5	0.75	1
Blair	1517	0.58	0.29	0	0.25	0.75	0.75	1
Hague	1517	0.44	0.31	0	0.25	0.25	0.75	1

Europe	1517	0.57	0.33	0	0.3	0.5	0.9	1
Political.knowledge	1517	0.51	0.36	0	0	0.67	0.67	1
Labour=1 Conservative=0	1517	0.7	0.46	0	0	1	1	1
Male=1_Female=0	1517	0.47	0.5	0	0	0	1	1

Table 7. 5-Point summary stat of scaled data

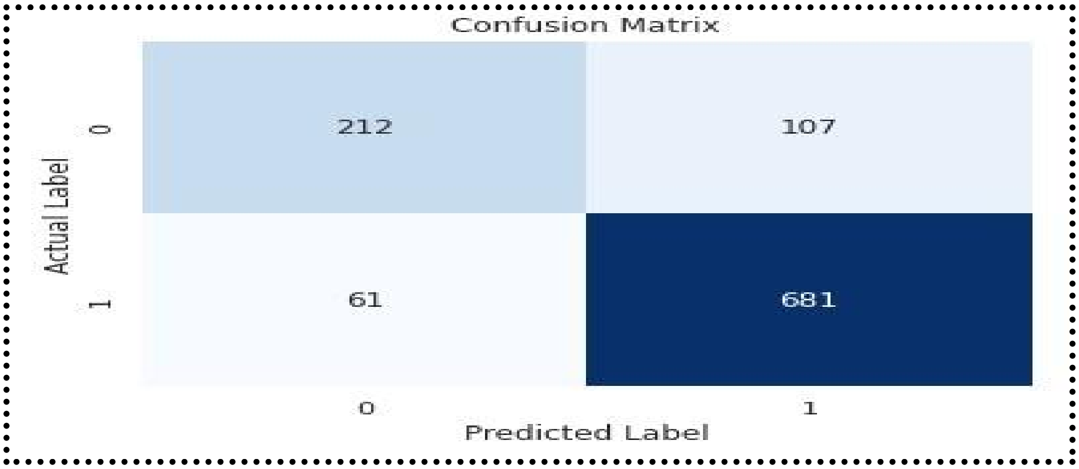
- ***Splitting data into Train and Test at the default radio of 30:70% with the random state as 123.***
- ***After splitting the data into test and train, distribution is:***
  - ***`X_train (1061, 8)`***
  - ***`X_test (456, 9)`***
  - ***`train_labels (1061)`***
  - ***`test_labels (456)`***
- ***Target variable class distribution, we can see that the proportion of Ones and Zeroes in the training and test set is the same as the proportion of Ones and Zeroes that were present in the whole dataset.***
  - ***Train_set:***
    - ***1: 69.93%***
    - ***0: 30.06%***
  - ***Test_set:***
    - ***1: 69.07***
    - ***0: 30.92***

1.4 Apply Logistic Regression and LDA (Linear Discriminant Analysis) (2 pts). Interpret the inferences of both models (2 pts). Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (overfitting or underfitting)

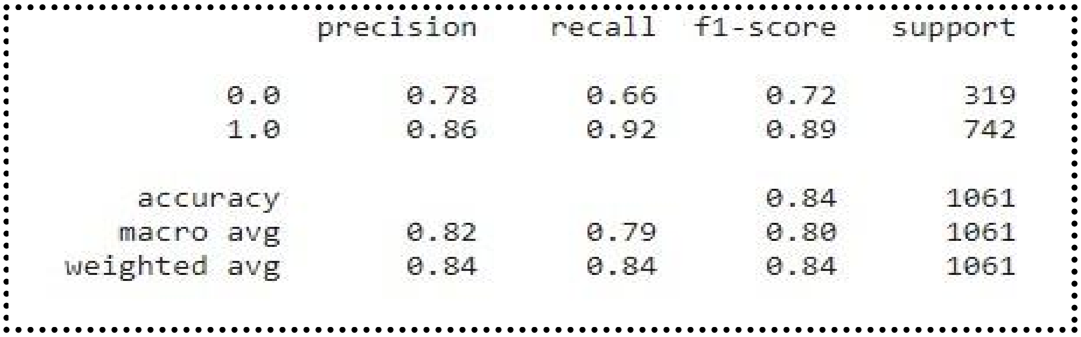
Logistic Regression model applied.

- ***Model_train Score: 0.8416588124410933***
- ***Model_test Score: 0.8179824561403509***

❖ ***Logistic Regression: Confusion Matrix Train data:***

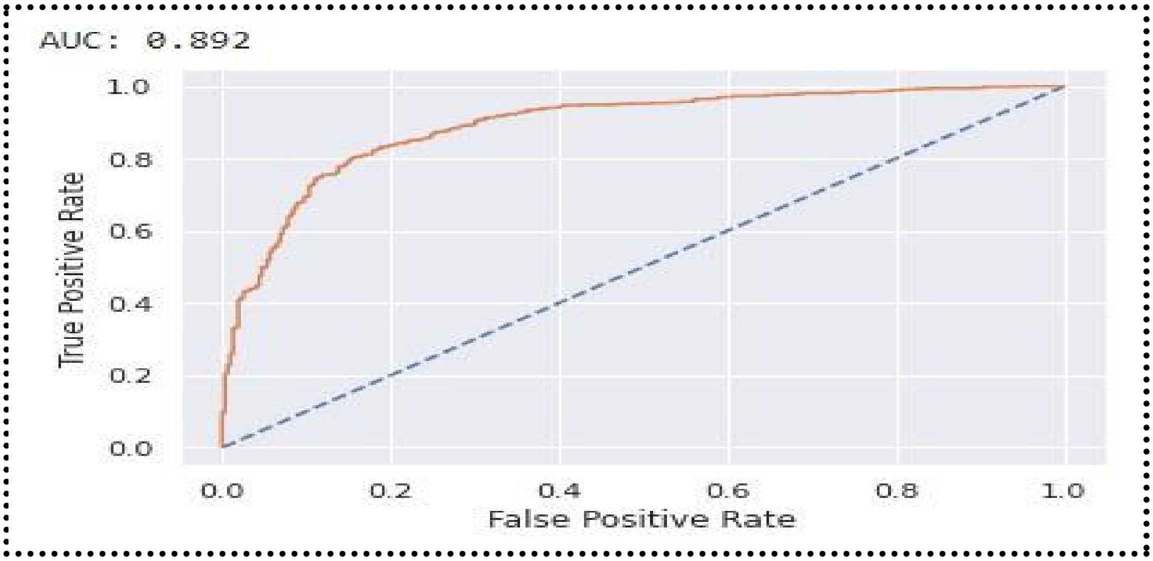


❖ ***Classification_report:***

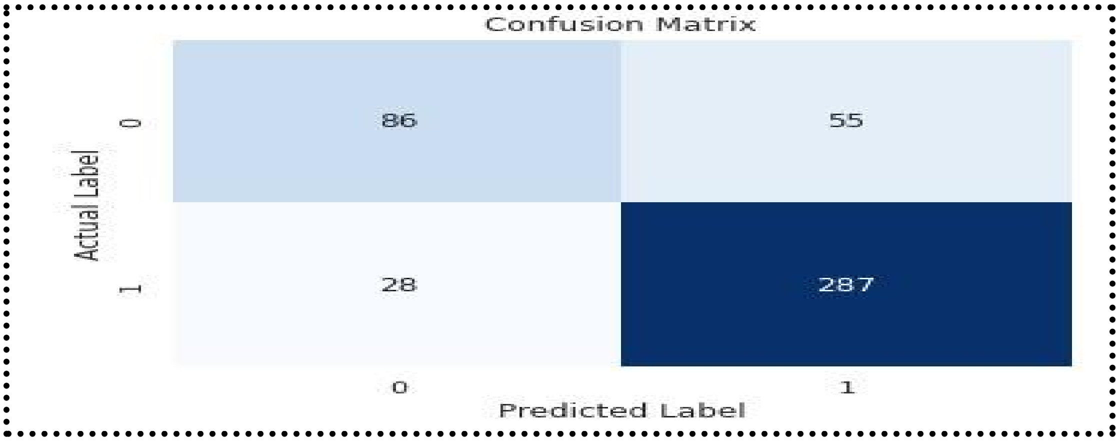


```
              precision    recall  f1-score   support

         0.0       0.78      0.66      0.72       319
         1.0       0.86      0.92      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
```

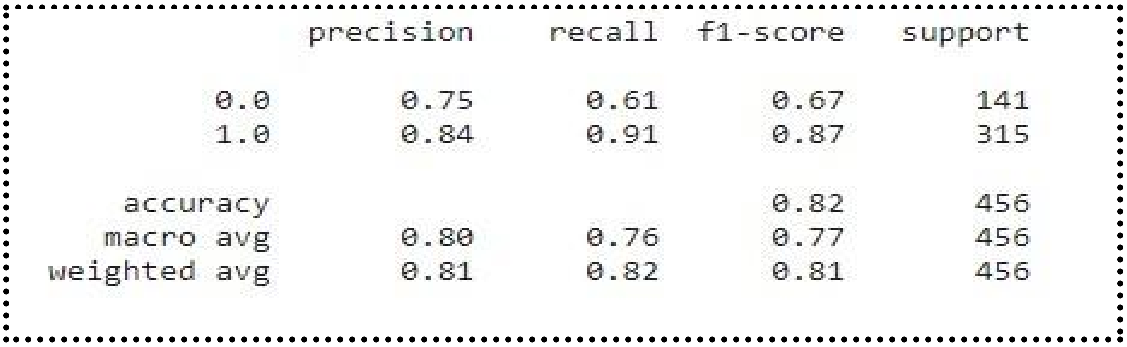
❖ ***AUC and ROC curve for the training data:***



- ***Confusion Matrix Test data:***

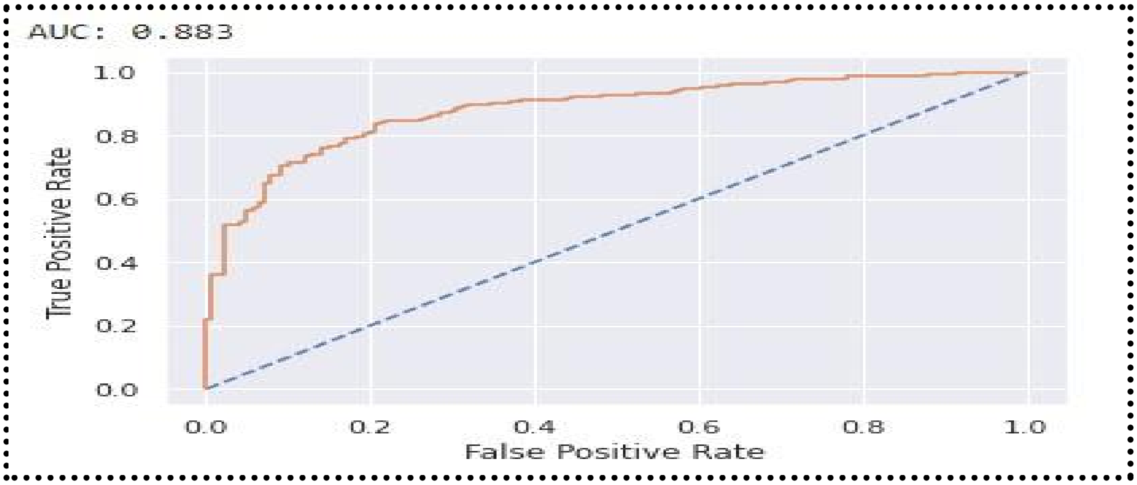


- ***Classification_report:***



```
              precision    recall  f1-score   support

         0.0       0.75      0.61      0.67       141
         1.0       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
```

- ***AUC and ROC curve for the testing data:***

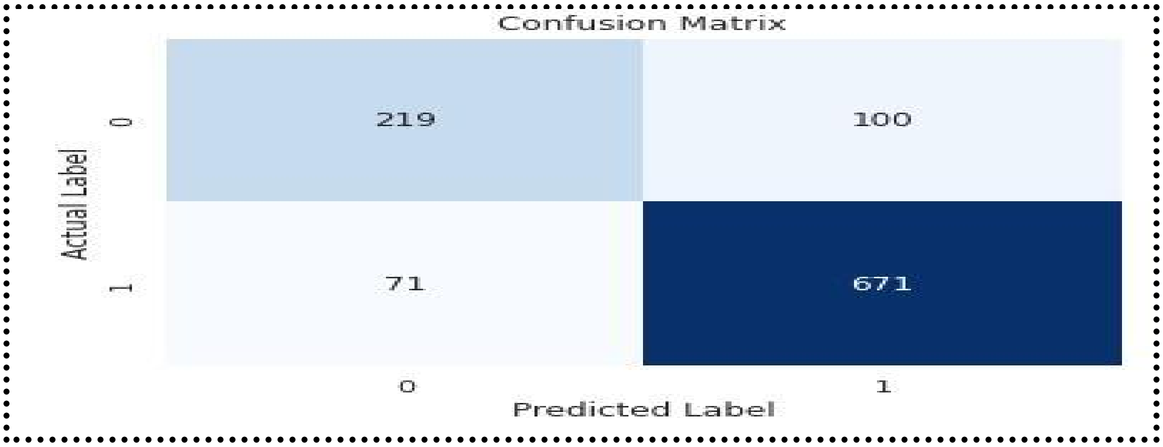


- ***From the data above we can say that the model with default values:***
  - ***Accuracy from the Training data is: 0.892***
  - ***Accuracy from the Test data is: 0.883***
- ***As we see, the accuracy of training and test results are almost the same. That the model is not*** **Overfitted, we have a good model.**

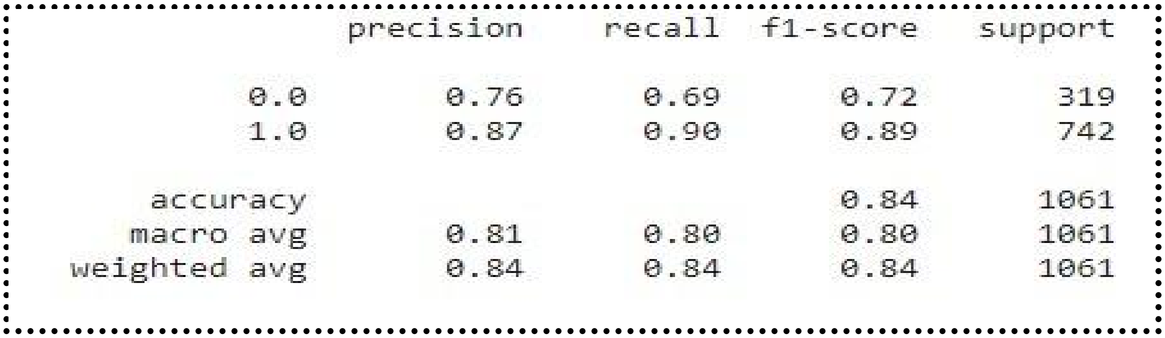
Linear discriminant analysis (LDA):

- ❖ *We start with Default parameters to implement* **LDA** *to our split data of 30:70*
    - ➢ ***Model_train Score: 0.8388312912346843***
    - ➢ ***Model_test Score: 0.8245614035087719***

LDA: Confusion Matrix Train data:

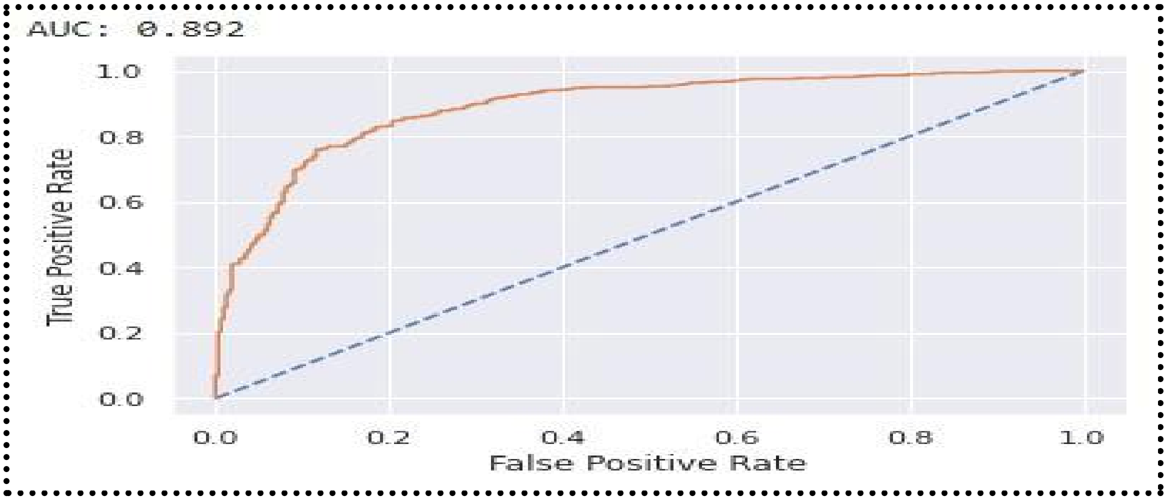


- ❖ ***Classification_report:***

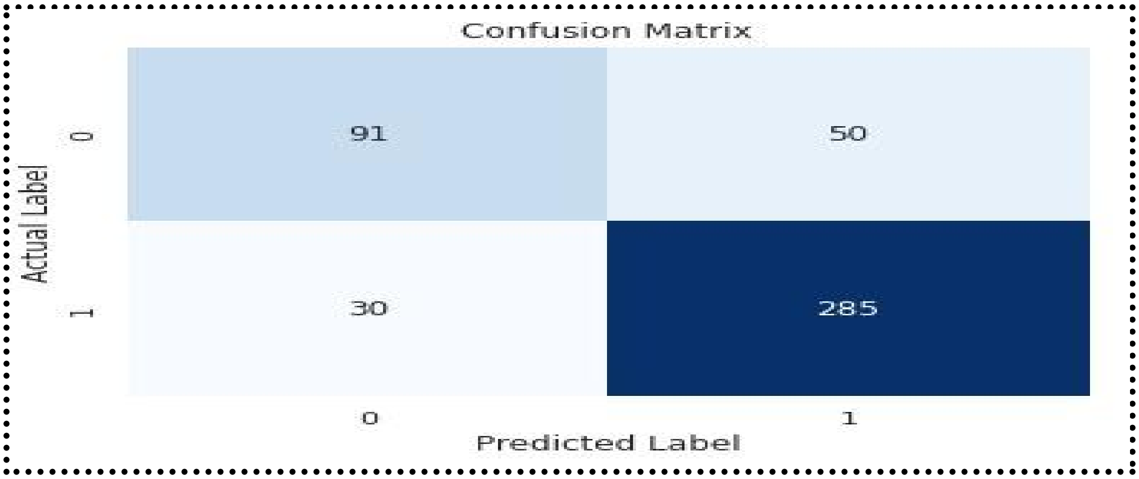


```
              precision    recall  f1-score   support

         0.0       0.76      0.69      0.72       319
         1.0       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
```

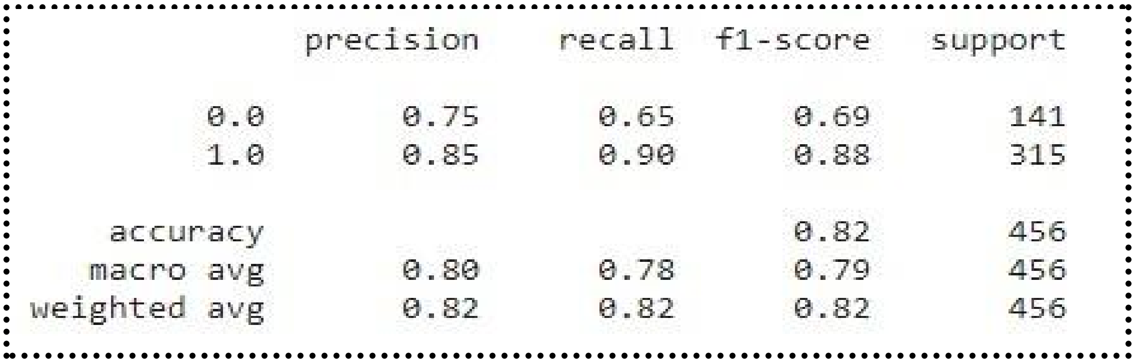
- ❖ ***AUC and ROC curve for the training data:***



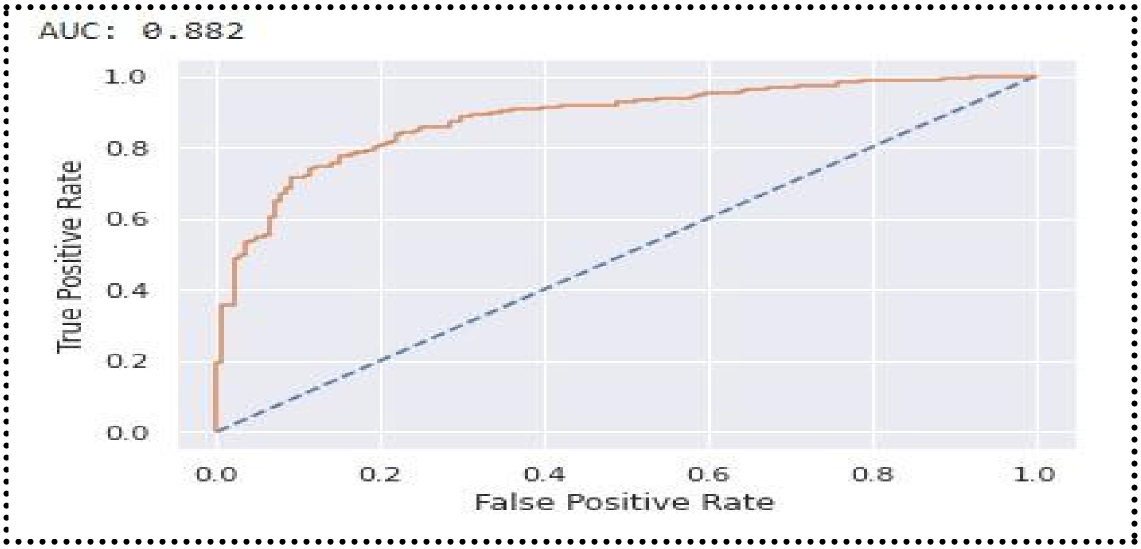
- ***Confusion Matrix Test data:***



- ***Classification_report:***


```
              precision    recall  f1-score   support

         0.0       0.75      0.65      0.69       141
         1.0       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
```

- ***AUC and ROC curve for the testing data:***



- ***From the data above we can say that the model with default values.***
  - ***Accuracy from the Training data is: 0.892***

- ➢ *Accuracy from the Test data is: 0.882*

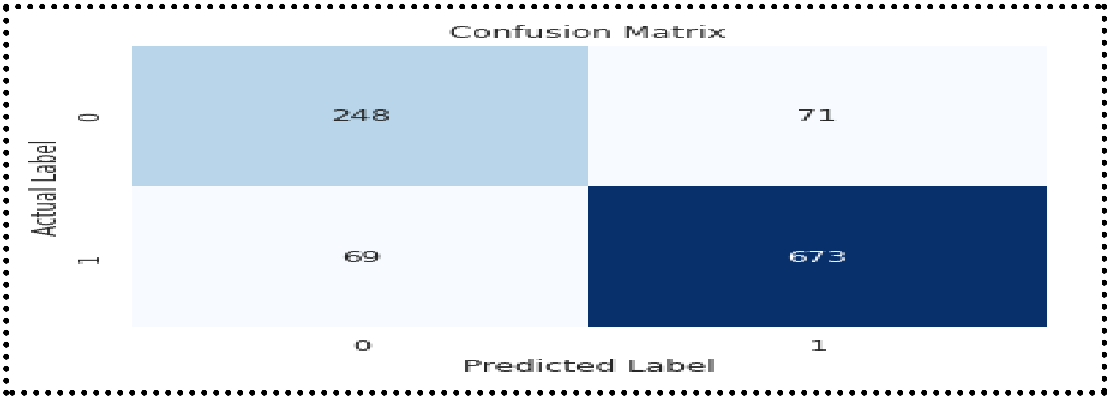
- ❖ ***As we see, the accuracy of training and test results are very close. Which is good for our model. There is no overfitting.***
- ❖ ***As for comparison, the models(score and accuracy) with Logistic Regression and LDA and very close to each other, when compared to the Logistic regression model is slightly better.***

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 behind the selection of different values for the parameters involved in each model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (overfitting or underfitting)

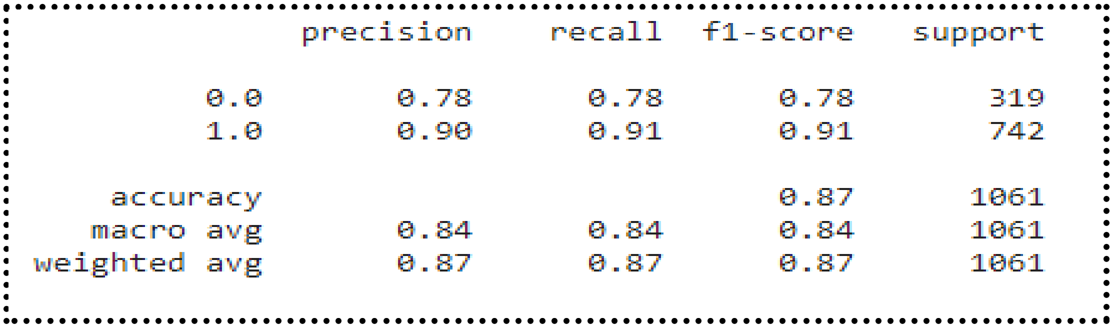
KNN model applied.

- ➢ ***Model_train Score: 0.8680490103675778***
- ➢ ***Model_test Score: 0.7982456140350878***

- ❖ *KNN : Confusion Matrix Train data:*

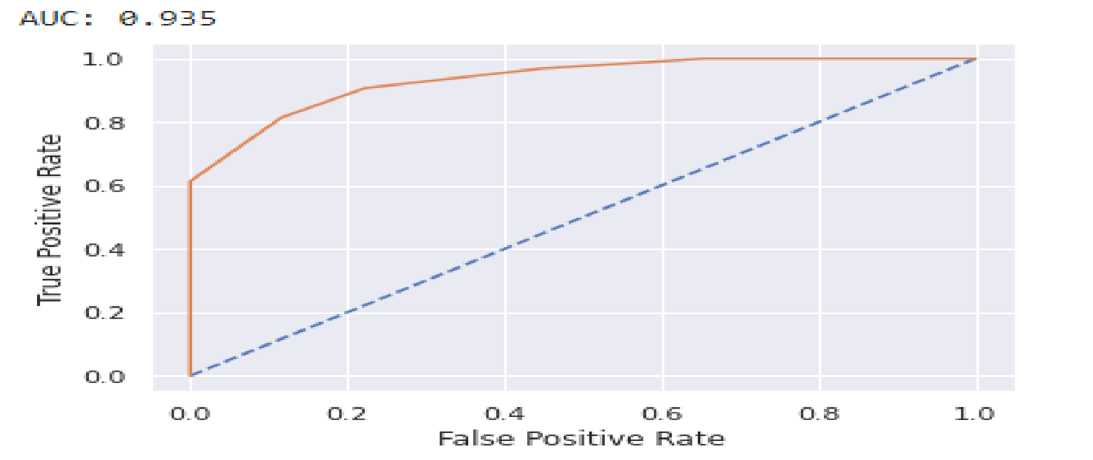


- ❖ *Classification_report:*



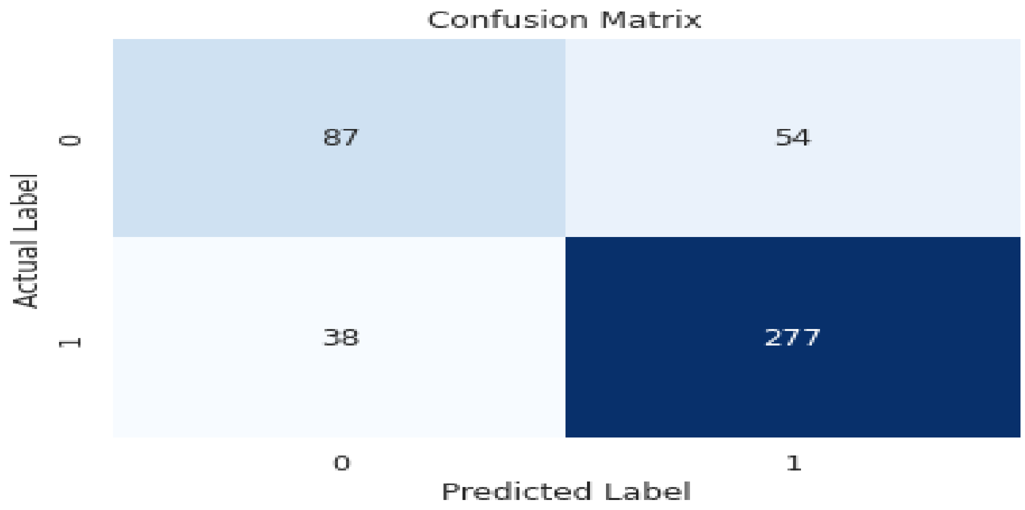
```
              precision    recall  f1-score   support

         0.0       0.78      0.78      0.78       319
         1.0       0.90      0.91      0.91       742

    accuracy                           0.87      1061
   macro avg       0.84      0.84      0.84      1061
weighted avg       0.87      0.87      0.87      1061
```

❖ *AUC and ROC curve for the training data:*

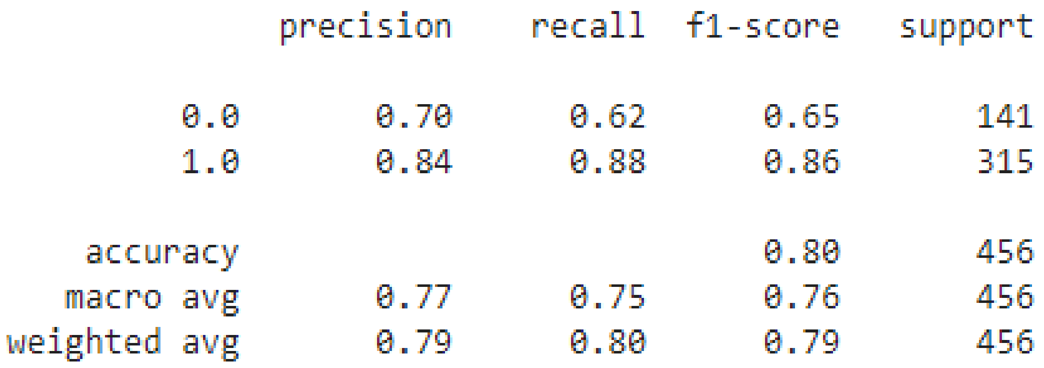


```
AUC: 0.935
```

❖ *Confusion Matrix Test data:*

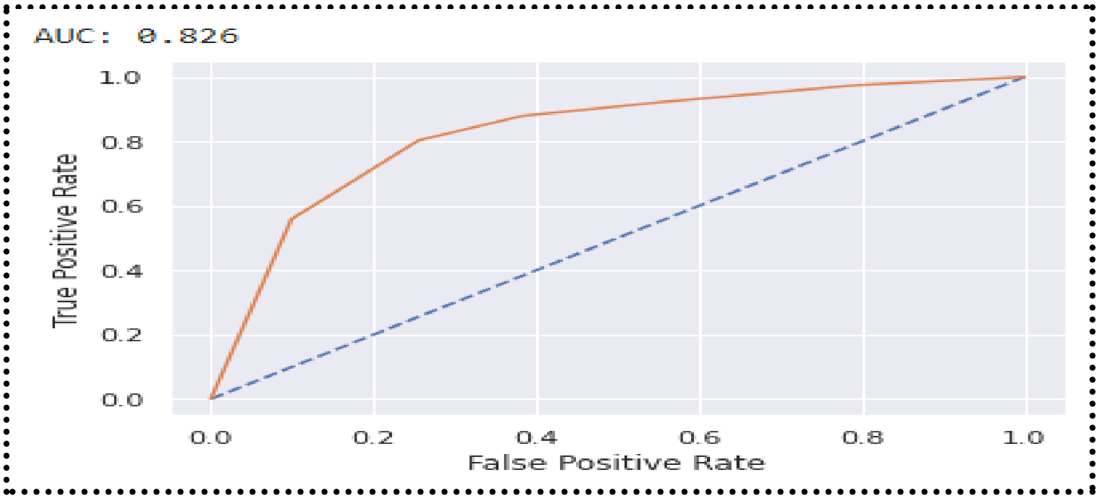


❖ *Classification_report:*



```
              precision    recall  f1-score   support

         0.0       0.70      0.62      0.65       141
         1.0       0.84      0.88      0.86       315

    accuracy                           0.80       456
   macro avg       0.77      0.75      0.76       456
weighted avg       0.79      0.80      0.79       456
```

AUC and ROC curve for the testing data:

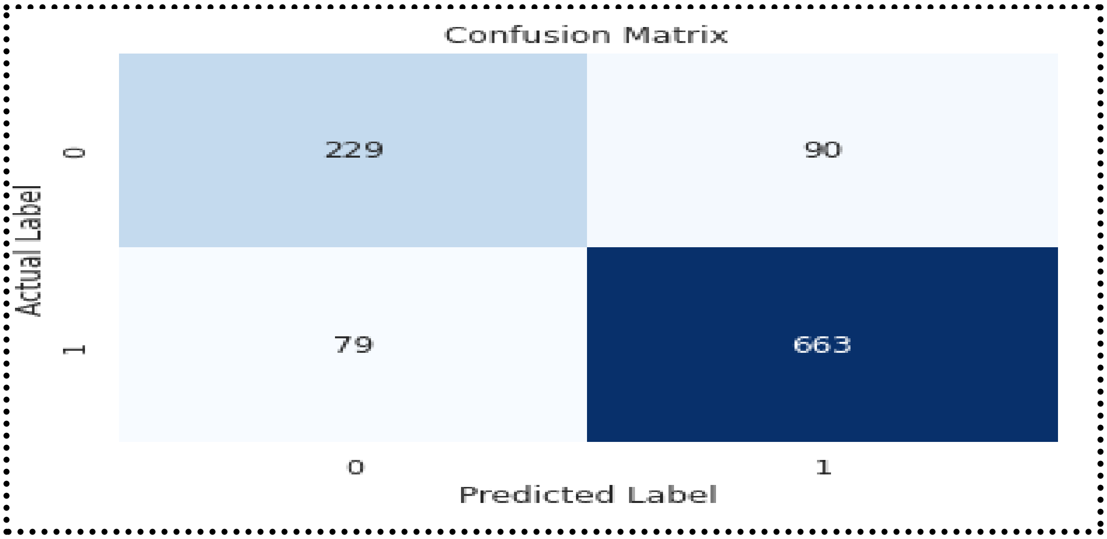


- ***From the data above we can say that the model with default values:***
  - ***Accuracy from the Training data is: 0.935***
  - ***Accuracy from the Test data is: 0.826***
- ***As we see, the accuracy of training and test results have a 10% drop. That the model is* Overfitted*. We can chose another K value to fix this problem.***

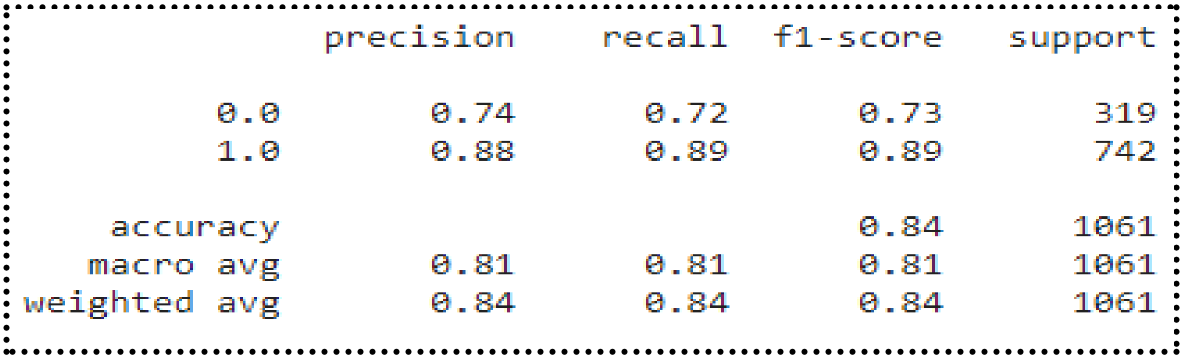
Naive Bayes Model:

- ***We with Default parameters to implement* Naive Bayes *to our split data of 30:70***
  - ***`Model_train Score: 0.8407163053722903`***
  - ***`Model_test Score: 0.8157894736842105`***

NB: Confusion Matrix Train data:



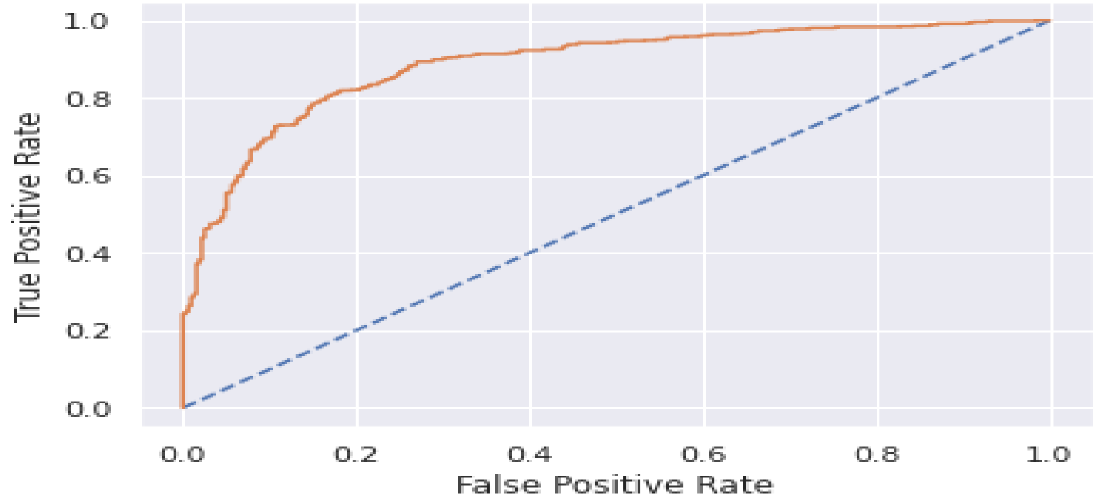
- ***Classification_report:***



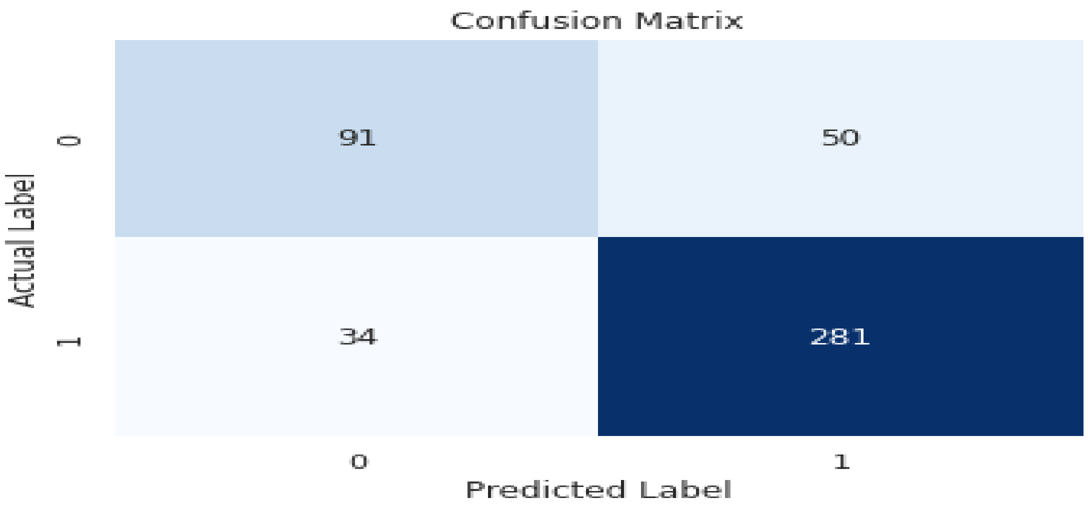
```
              precision    recall  f1-score   support

         0.0       0.74      0.72      0.73       319
         1.0       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
```

- ***AUC and ROC curve for the training data:***

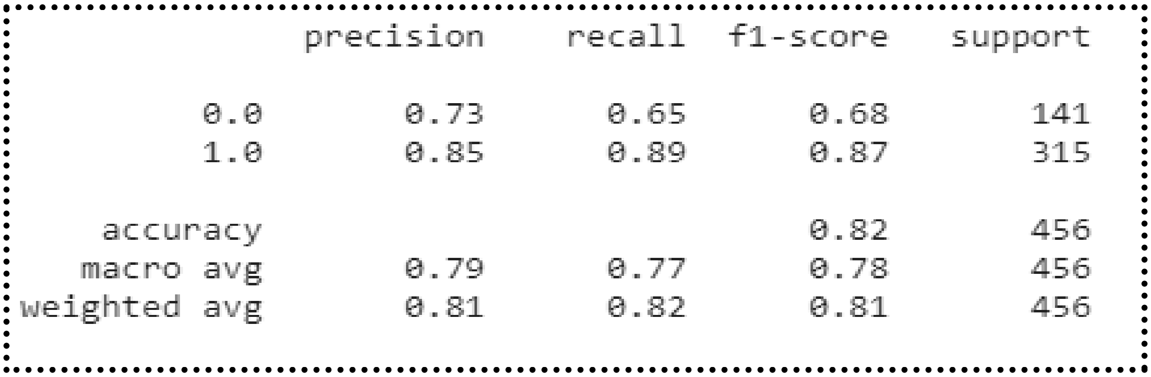
```
AUC: 0.888
```



- ***Confusion Matrix Test data:***

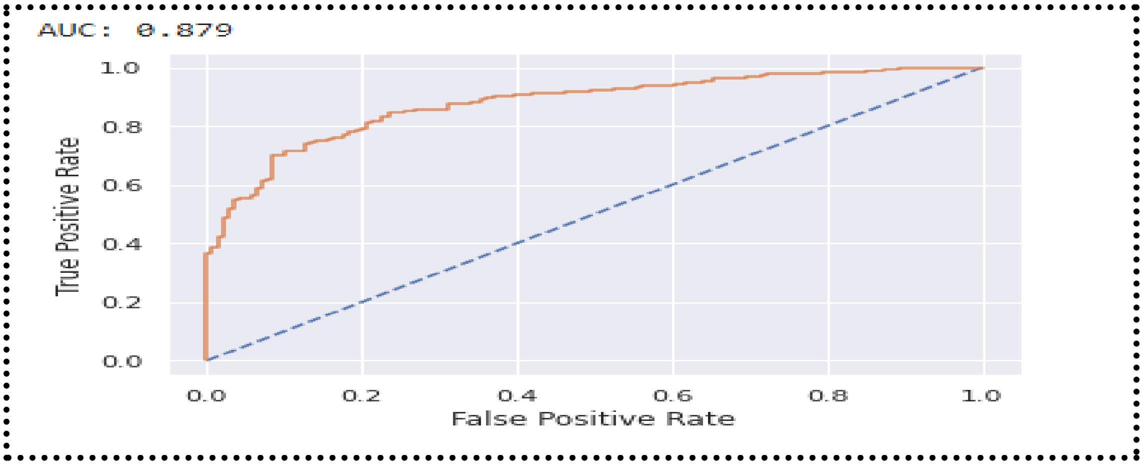


- *Classification_report:*



```
               precision    recall  f1-score   support

         0.0       0.73      0.65      0.68       141
         1.0       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
```

- *AUC and ROC curve for the testing data:*



```
AUC: 0.879
```

Inference:

- ***From the data above we can say that the model with default values.***
    - ***Accuracy from the Training data is: 0.888***
    - ***Accuracy from the Test data is: 0.879***
- ***As we see, the accuracy of training and test results are very close. Which is good for our model. There is no overfitting.***
- ***As for comparison, the models(score and accuracy) with KNN Model and Naive Bayes Model with default values, when compared to the KNN model, Naive Bayes is better.***

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. Define a logic behind choosing particular values for different hyper-parameters for a grid search. 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.

Model Tunning:

- **The model behavior is checked for the same with tuning. We can further work on it to improve.**
- ***The best parameters are:*** **Logistic Regression >> LogisticRegression('C': 10, 'penalty': 'l2', 'solver': 'saga', 'tol': 0.01)**

Logistic Regression :

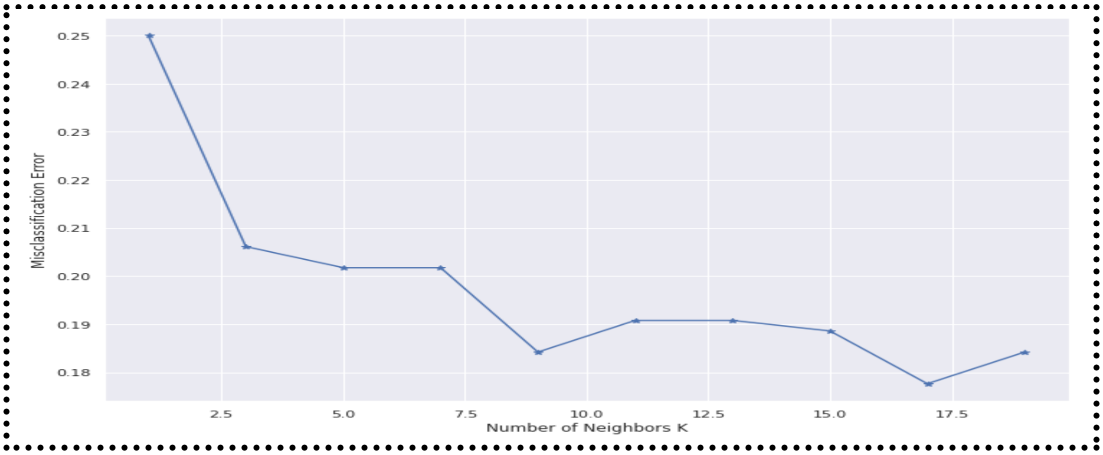
- *We with tunned parameters to implement* **Logistic Regression** *to our split data of 30:70*
  - *Model Score train: 0.8444863336475024*
  - *Model Score test: 0.8201754385964912*

Linear discriminant analysis (LDA):

- ***The best parameters are:*** **Linear discriminant analysis >> LinearDiscriminantAnalysis('shrinkage': 1, 'solver': 'lsqr', 'tol': 0.0001)**
- **Model Score after Tunning:**
  - *Model_train Score: 0.8388312912346843*
  - *Model_test Score: 0.8245614035087719*

KNN model applied.

- **The model behavior is checked for the same with tuning. We can further work on it to improve.**
- **KNN Misclassicification Plot:**



- **Different values of K accuracy with least Misclassification error:**
    - Accuracy Score for K=3 is 0.793859649122807
    - Accuracy Score for K=9 is 0.8157894736842105
    - Accuracy Score for K=10 is 0.8114035087719298
    - Accuracy Score for K=19 is 0.8157894736842105
    - Accuracy Score for K=17 is 0.8223684210526315
- *The best parameters are:* **KNN model >> K=17**
        - *Model_train Score: 0.8510838831291234*
        - *Model_test Score: 0.8223684210526315*

Inference:

- ***From the data above we can say that the KNN model with tunned values.***
    - ***Accuracy from the Training data is: 0.85***
    - ***Accuracy from the Test data is: 0.82***
- ***As we see, the accuracy of training and test results are very close. Which is good for our model. There is no overfitting.***
- ***As for comparison, the tunned models(score and accuracy) of KNN Model are now close with Naive Bayes Model default values, when compared to the KNN is now giving a slightly better result.***

Ada Boost model applied.

- *The parameters are:* **GradientBoostingClassifier(random_state=123)**
- **Model Score after Tunning:**
    - *Model_train Score: 0.8539114043355325*
    - *Model_test Score: 0.8070175438596491*

Gradient Boosting model applied.

- *The parameters are:* **AdaBoostClassifier(n_estimators=100, random_state=123)**
- **Model Score after Tunning:**
    - *Model_train Score: 0.8092105263157895*
    - *Model_test Score: 0.8092105263157895*

Bagging model DecisionTree:

- *The parameters are:* **BaggingClassifier(base_estimator=DecisionTreeClassifier(), n_estimators=100, random_state=123)**
- **Model Score after Tunning:**
  - *Model_train Score: 0.9990574929311969*
  - *Model_test Score: 0.8070175438596491*
- *The model is overfitting, but it's alright on ensemble models.*

Bagging KNN:

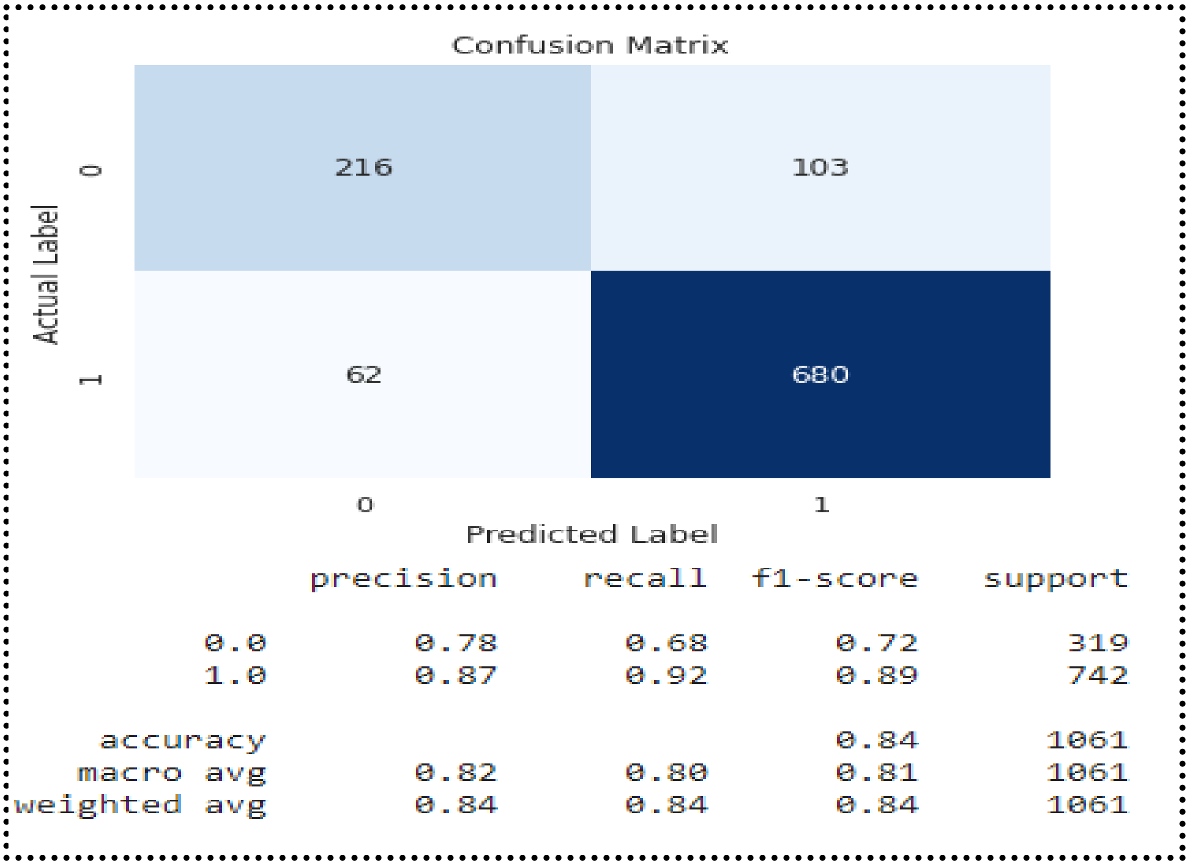
- *The parameters are:* **BaggingClassifier(base_estimator=KNeighborsClassifier(), n_estimators=100, random_state=123)**
- **Model Score after Tunning:**
  - *Model_train Score: 0.8774740810556079*
  - *Model_test Score: 0.7982456140350878*

Bagging Random Forest:

- *The parameters are:* **BaggingClassifier(base_estimator=RandomForestClassifier(), n_estimators=100, random_state=123)**
- **Model Score after Tunning:**
  - *Model_train Score: 0.9707822808671065*
  - *Model_test Score: 0.8092105263157895*
- ***We see Decision tree and Random Forest gives as the same outcome. We will go with Random forest in detail.***

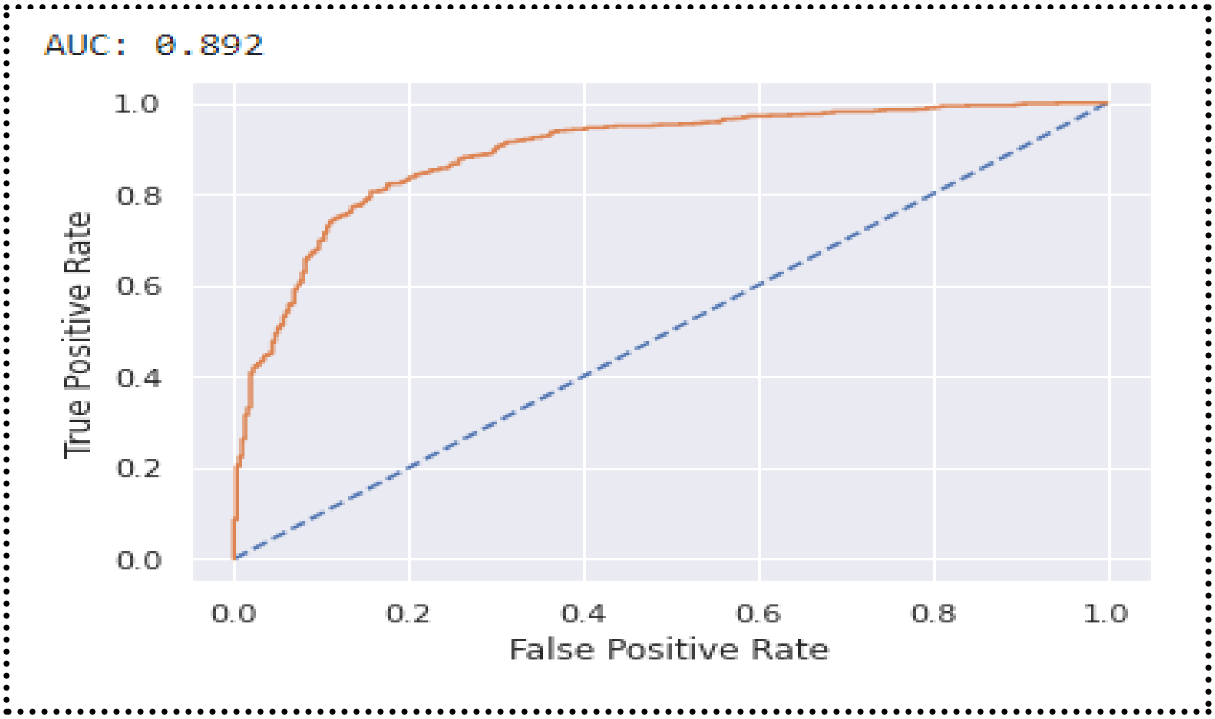
1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model, classification report (4 pts) Final Model - Compare and comment on all models on the basis of the performance metrics in a structured tabular manner. Describe 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.

Logistic Regression: Confusion Matrix & Classification_report Train data:



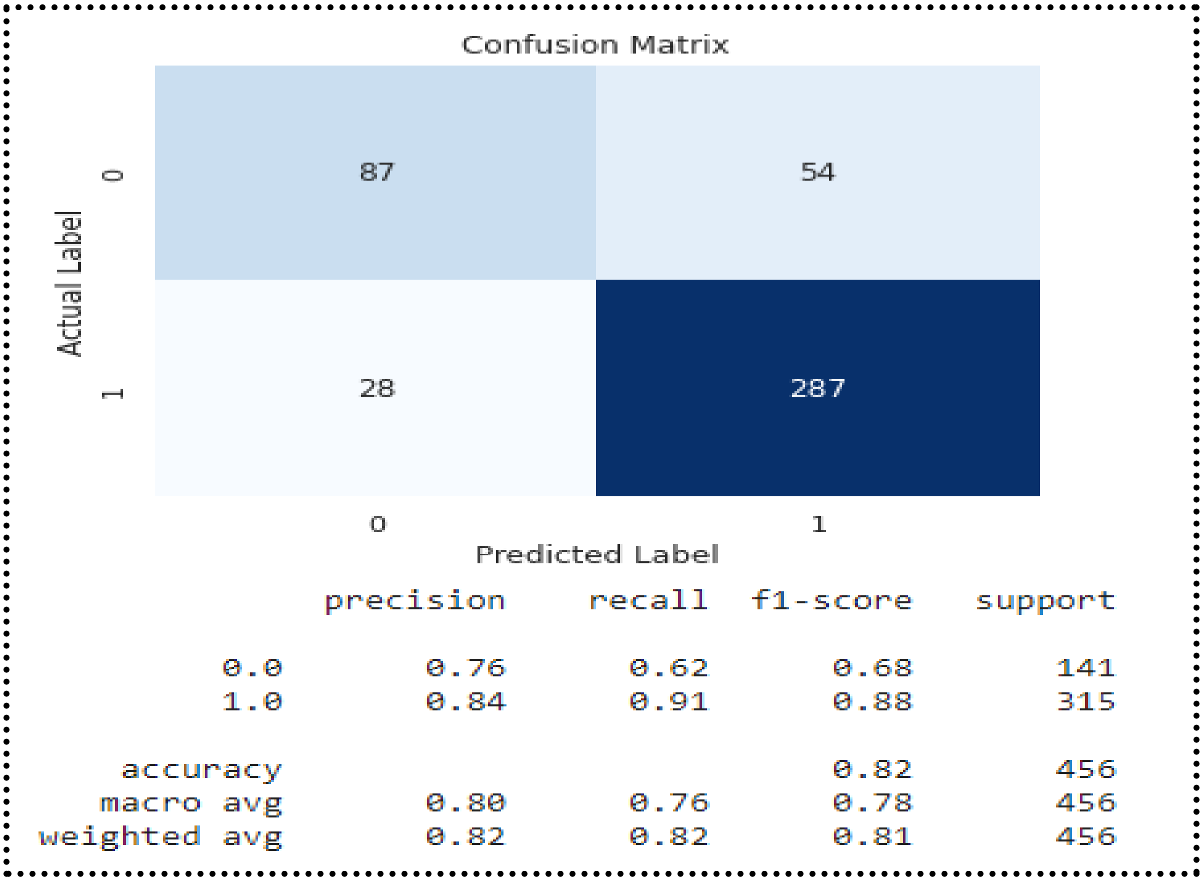
```
              precision    recall  f1-score   support

         0.0       0.78      0.68      0.72       319
         1.0       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
```

AUC and ROC curve for the training data:



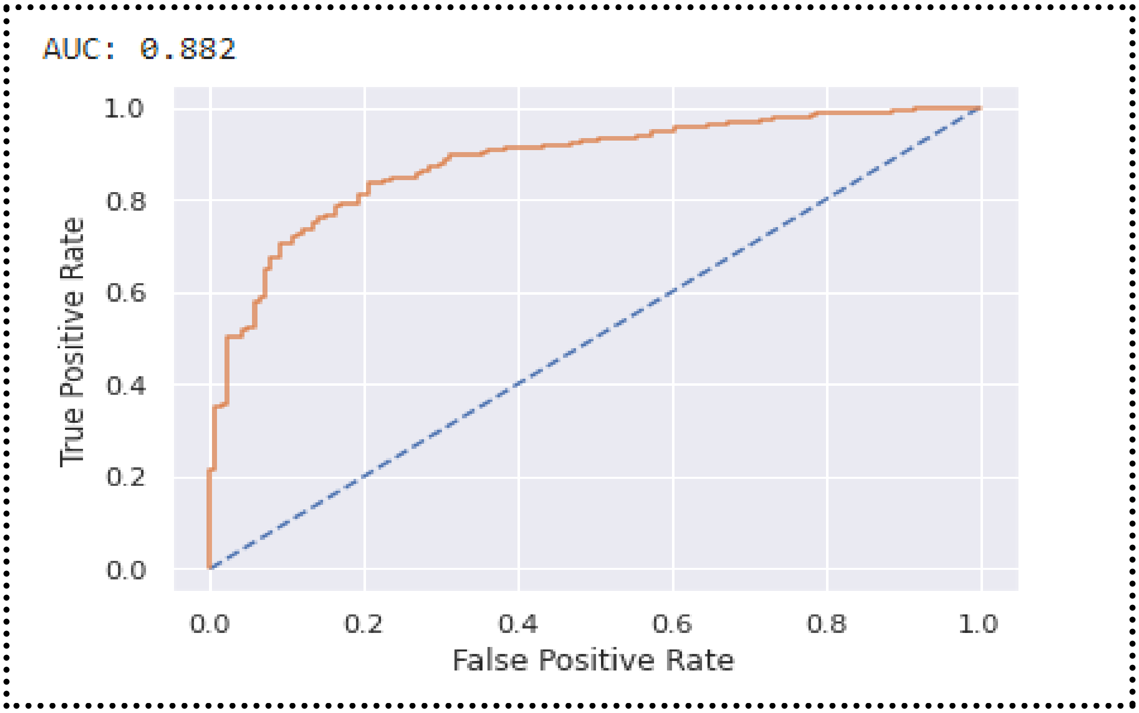
```
AUC: 0.892
```

Confusion Matrix & Classification_report Test data:



```
                  precision    recall  f1-score   support

         0.0         0.76      0.62      0.68       141
         1.0         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
```

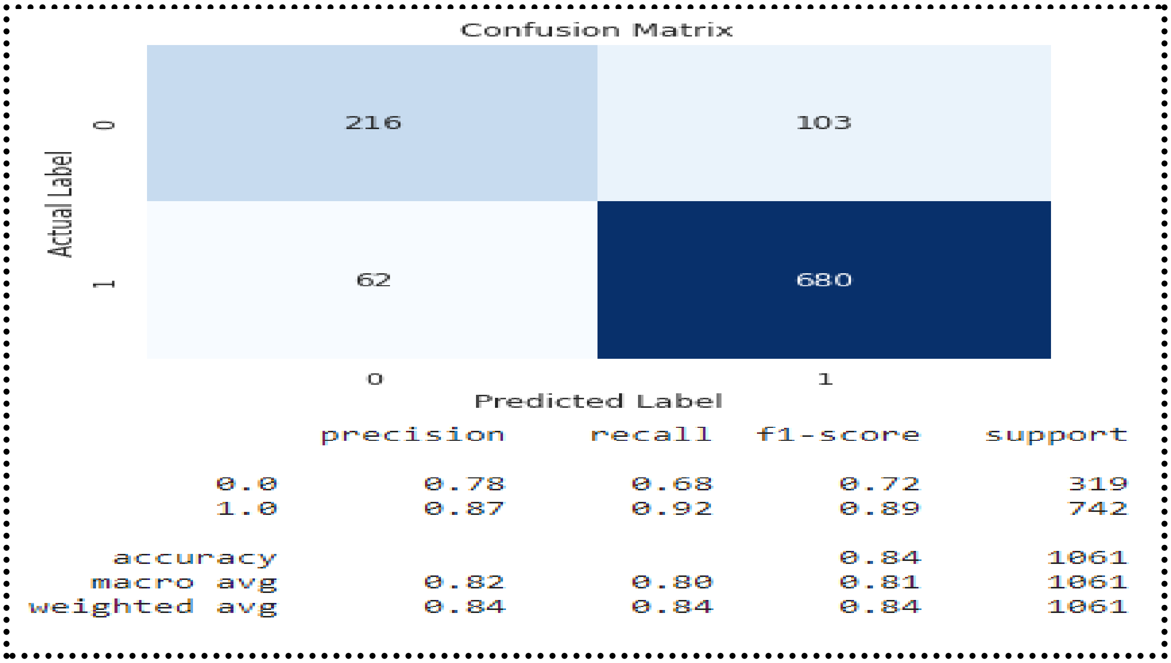
AUC and ROC curve for the testing data:



```
AUC: 0.882
```

- ***From the data above we can say that the model with default values:***
    - ***Accuracy from the Training data is: 0.892***
    - ***Accuracy from the Test data is: 0.882***
- ***As we see, the accuracy of training and test results are almost the same. That the model is not* Overfitted, we have a good model.**
- ***The model is the same as our default model for Logic Regression after tunning.***

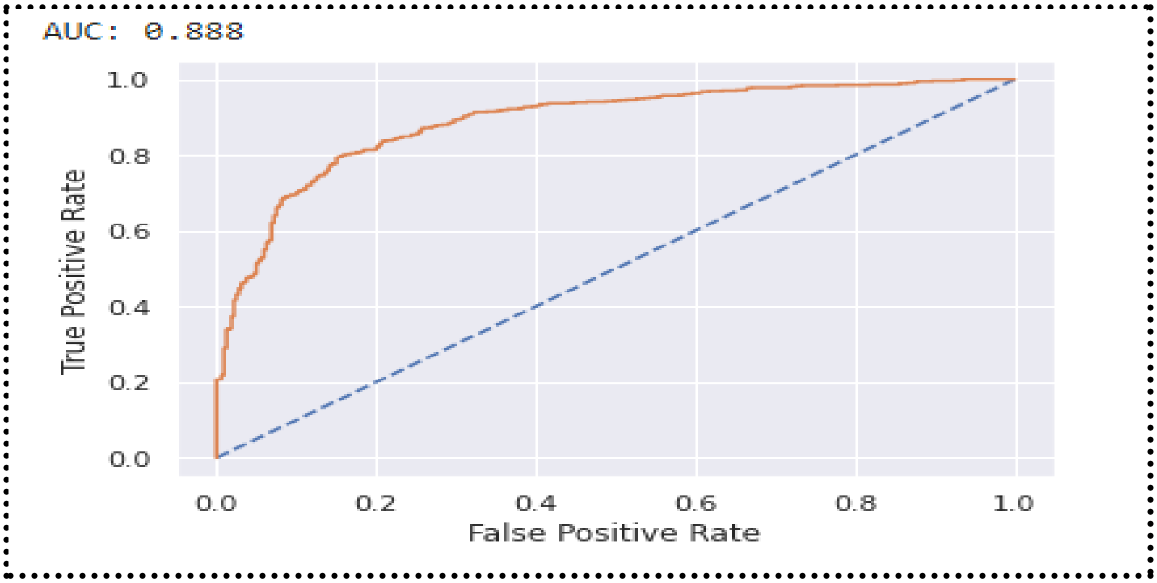
LDA: Confusion Matrix & Classification_report Train data:



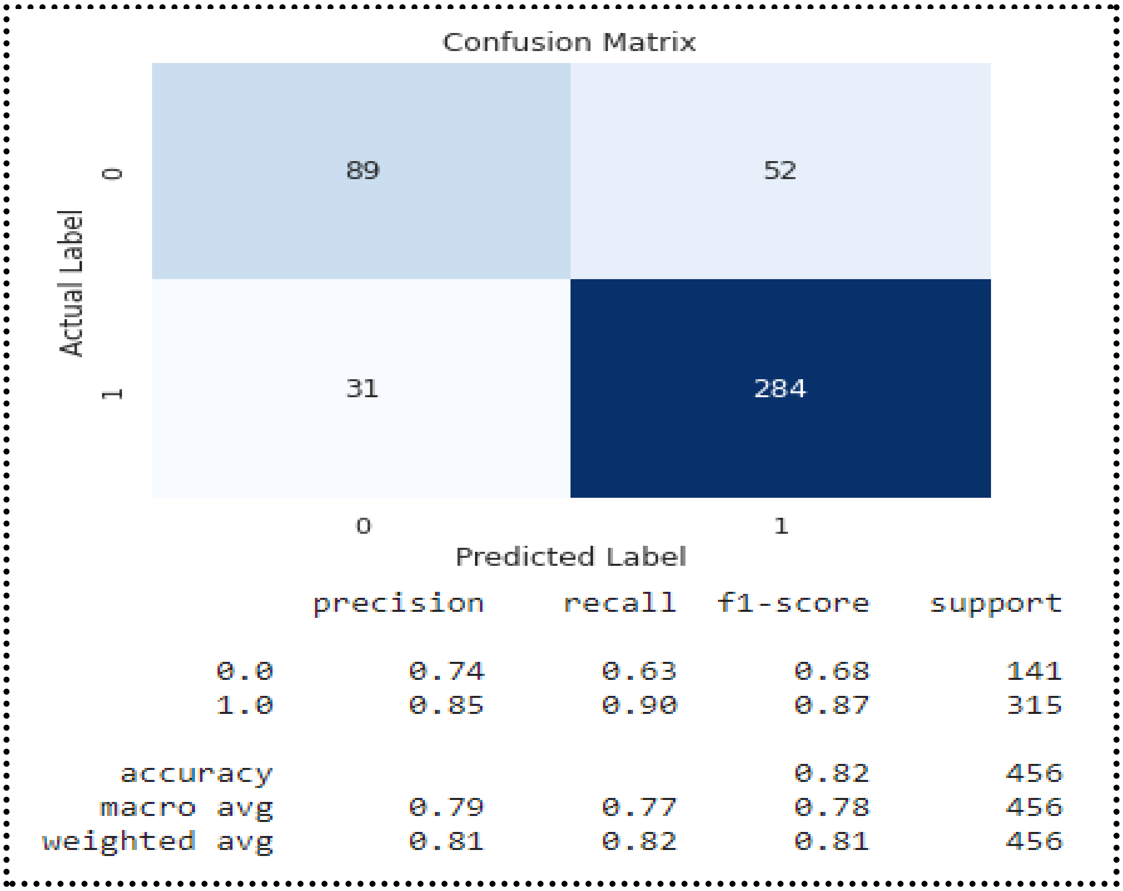
```
              precision    recall  f1-score   support

         0.0       0.78      0.68      0.72       319
         1.0       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
```

- *AUC and ROC curve for the training data:*

```
AUC: 0.888
```

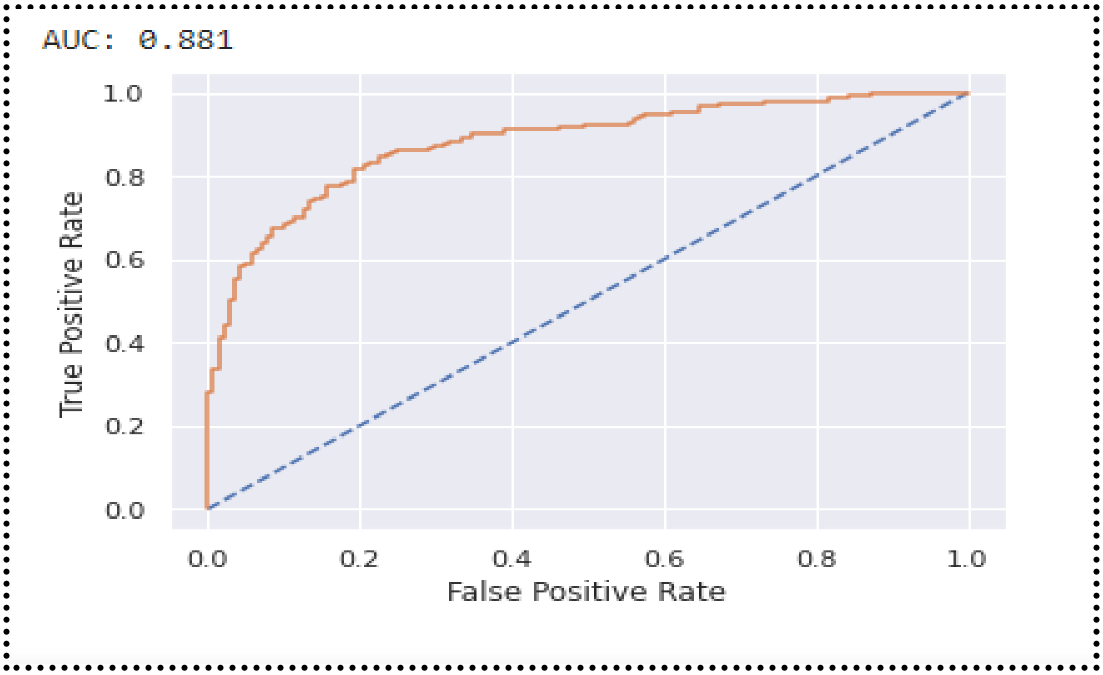


Confusion Matrix & Classification_report Test data:



```
              precision    recall  f1-score   support

         0.0       0.74      0.63      0.68       141
         1.0       0.85      0.90      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
```

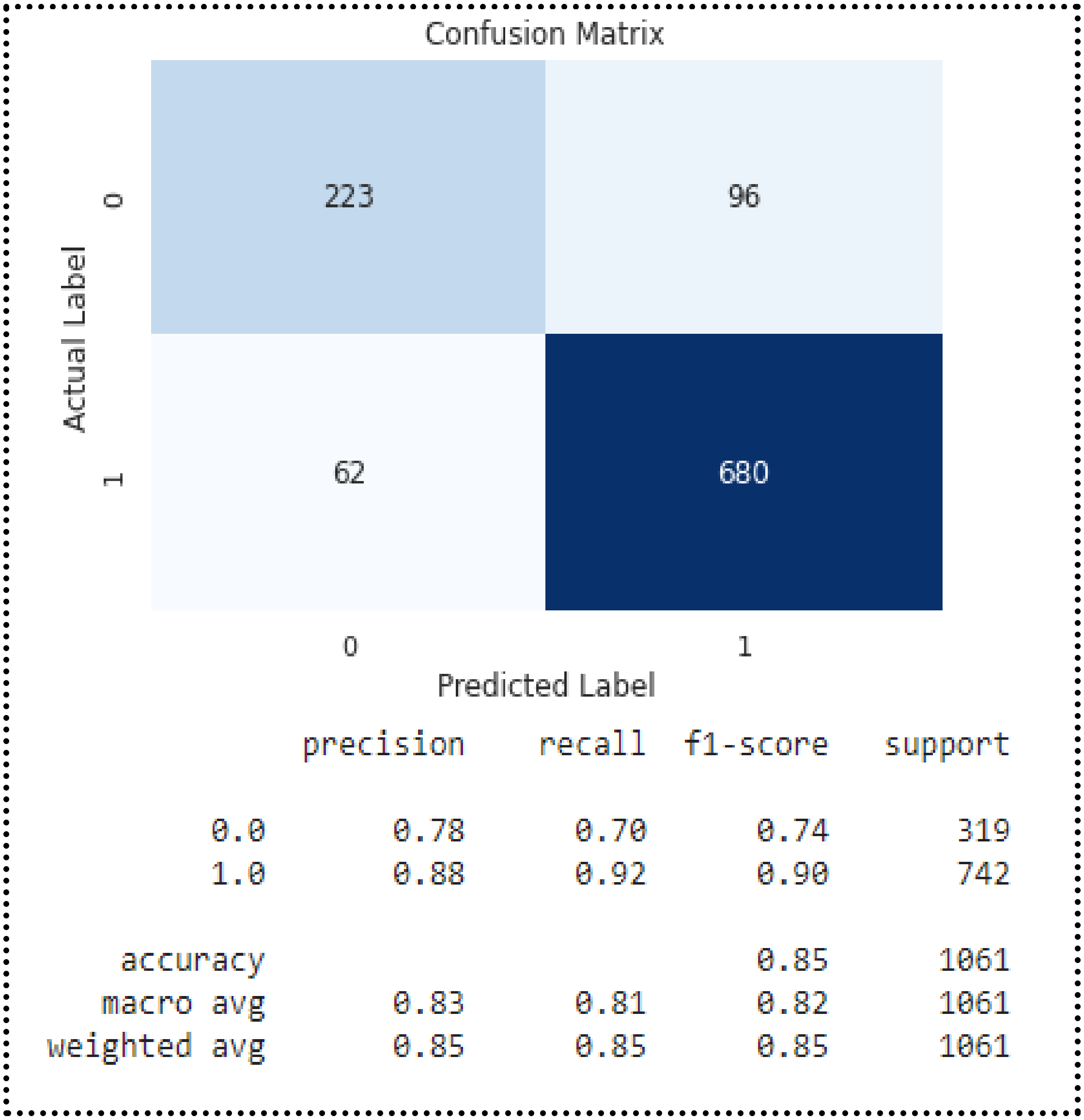
AUC and ROC curve for the testing data:



```
AUC: 0.881
```

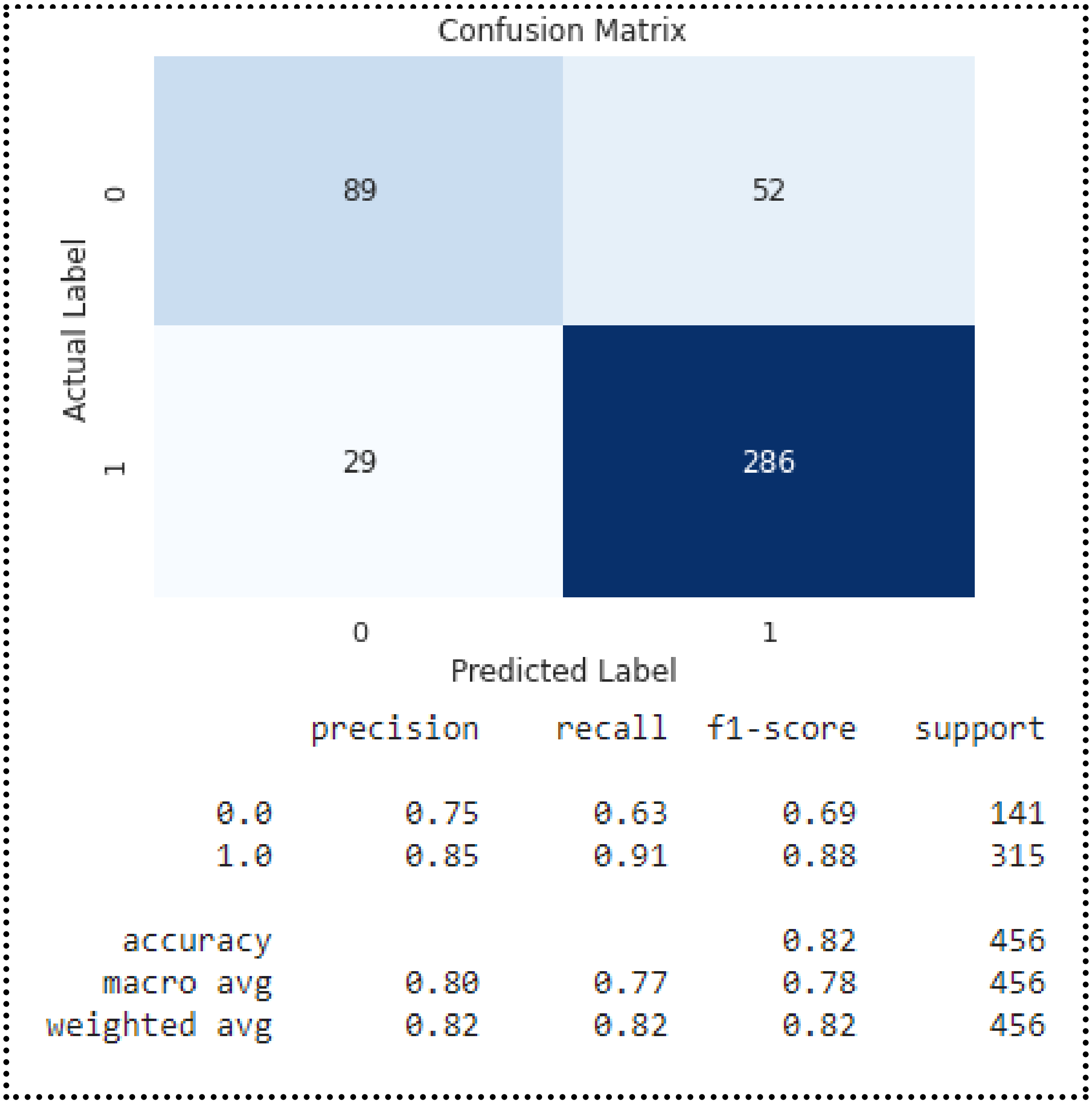
- ***From the data above we can say that the model with default values.***
    - ***Accuracy from the Training data is: 0.888***
    - ***Accuracy from the Test data is: 0.881***
- ***As we see, the accuracy of training and test results are very close. Which is good for our model. The overfitting has been fixed.***
- ***As for comparison, the models(score and accuracy) with Logistic Regression and LDA and very close(almost same) to each other, when compared to the Logistic regression model is slightly better.***

KNN, K=17 : Confusion Matrix & Classification_report Train data:



```
              precision    recall  f1-score   support

         0.0       0.78      0.70      0.74       319
         1.0       0.88      0.92      0.90       742

    accuracy                           0.85      1061
   macro avg       0.83      0.81      0.82      1061
weighted avg       0.85      0.85      0.85      1061
```

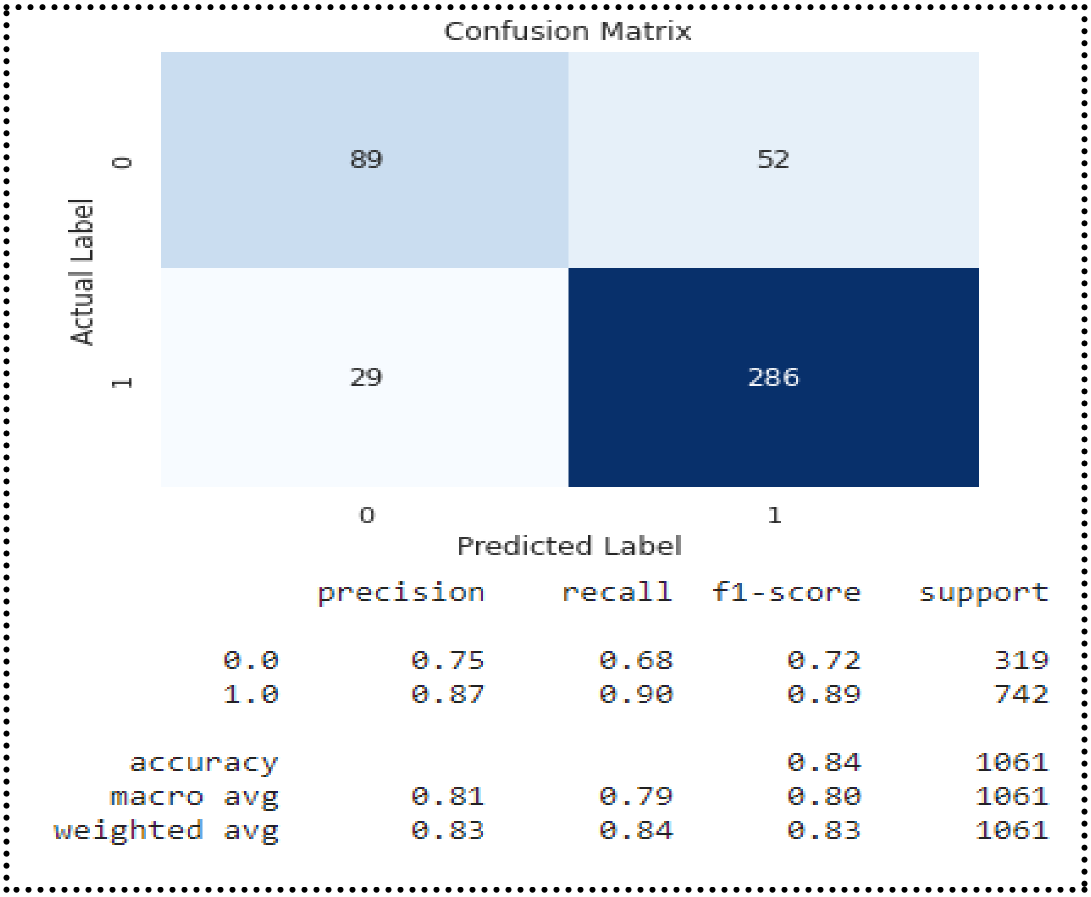
Confusion Matrix & Classification_report Test data:



```
              precision    recall  f1-score   support

         0.0       0.75      0.63      0.69       141
         1.0       0.85      0.91      0.88       315

    accuracy                           0.82       456
   macro avg       0.80      0.77      0.78       456
weighted avg       0.82      0.82      0.82       456
```

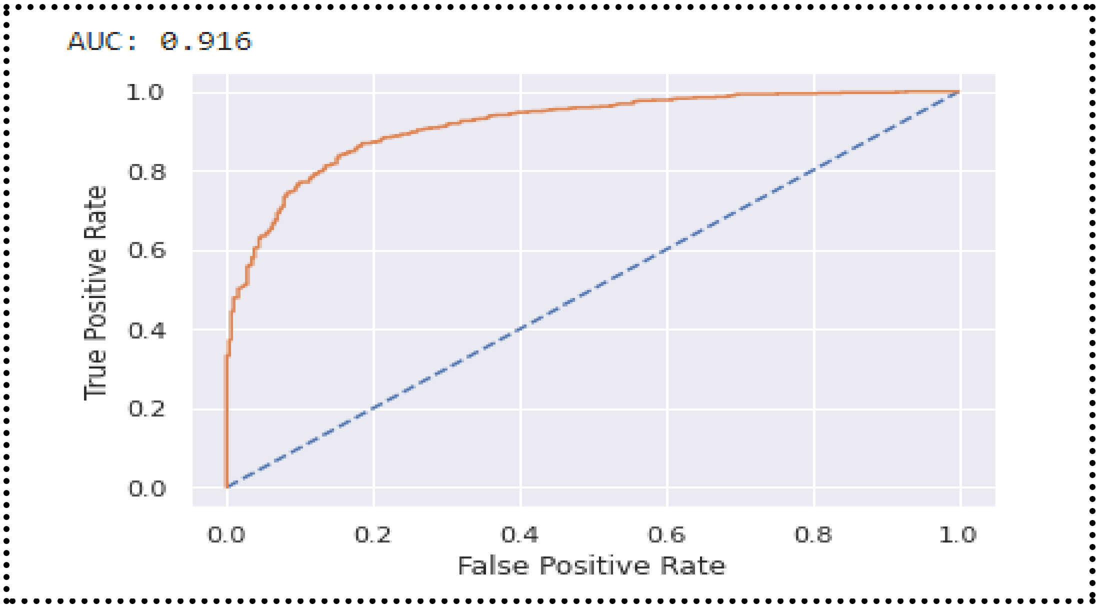
- ***From the data above we can say that the model with tunned values gives us a better result from default value:***
  - ***Accuracy from the Training data is: 0.918822719245621***
  - ***Accuracy from the Test data is: 0.8640999662276259***
- ***As we see, the accuracy of training and test results have a 5% drop, but they are closer when compared to default model. That the model's*** **Overfitting is addressed.** ***(before was10%)***

Ada Boost: Confusion Matrix & Classification_report Train data:



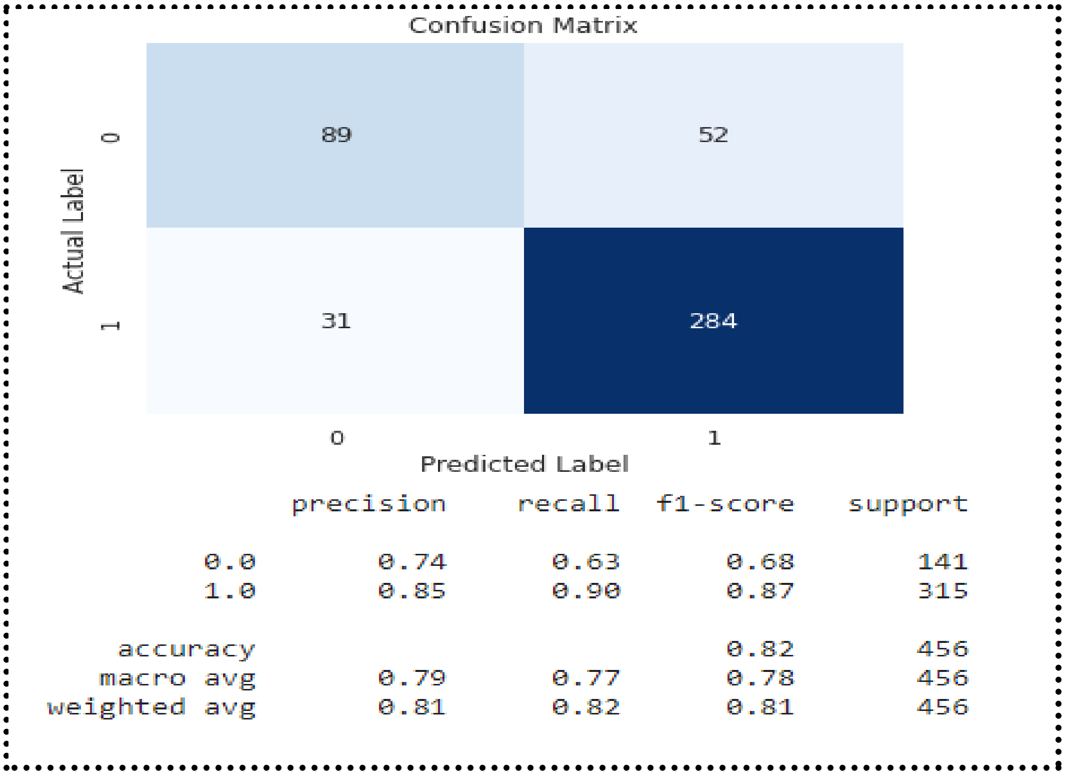
```
              precision    recall  f1-score   support

         0.0       0.75      0.68      0.72       319
         1.0       0.87      0.90      0.89       742

    accuracy                           0.84      1061
   macro avg       0.81      0.79      0.80      1061
weighted avg       0.83      0.84      0.83      1061
```

- *AUC and ROC curve for the training data:*



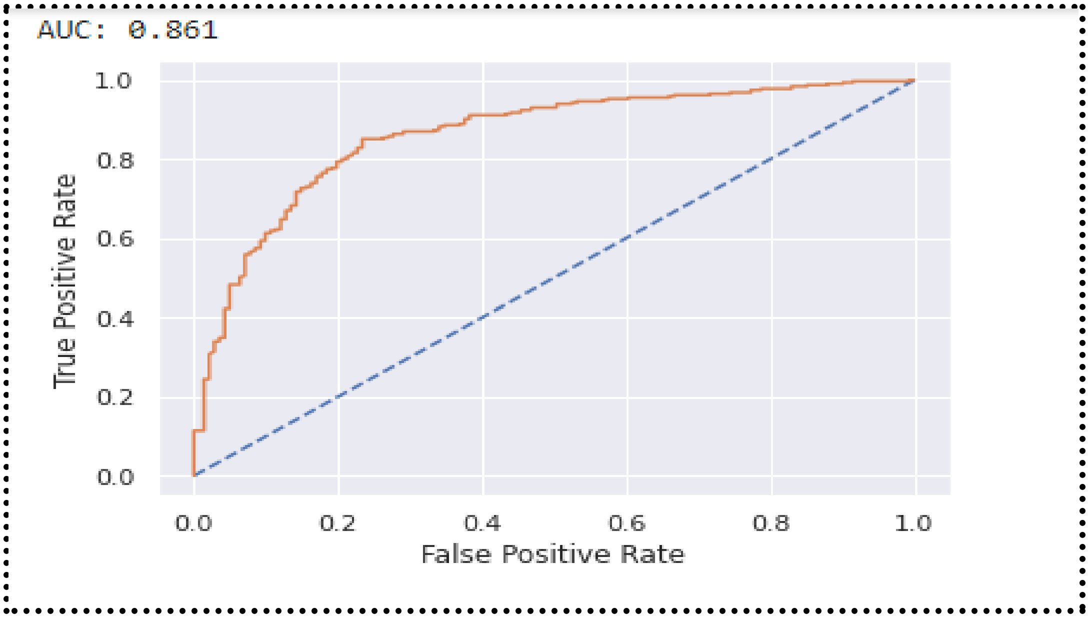
```
AUC: 0.916
```

Confusion Matrix & Classification_report Test data:



```
              precision    recall  f1-score   support

         0.0       0.74      0.63      0.68       141
         1.0       0.85      0.90      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
```

AUC and ROC curve for the testing data:



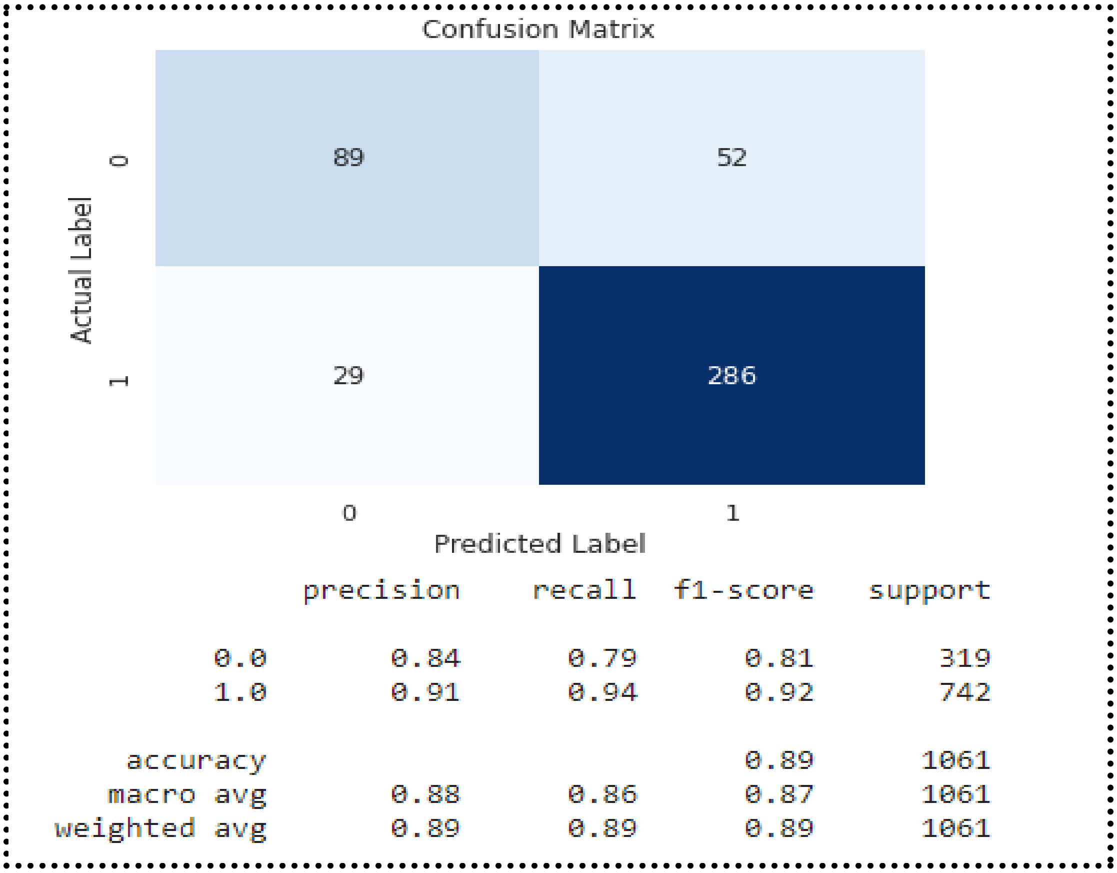
```
AUC: 0.861
```

➢ ***From the data above we can say that the model with default values.***

- ***Accuracy from the Training data is: 0.916***
- ***Accuracy from the Test data is: 0.861***

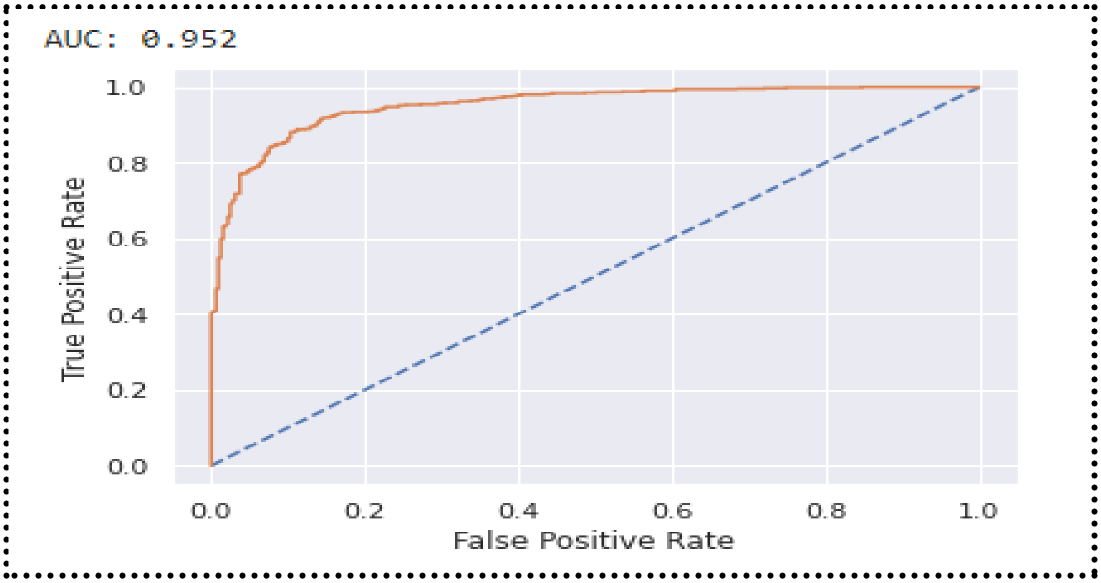
- *As we see, the accuracy of training and test results aren't very close. The mode is overfitting.*

Gradient Boosting : Confusion Matrix & Classification_report Train data:



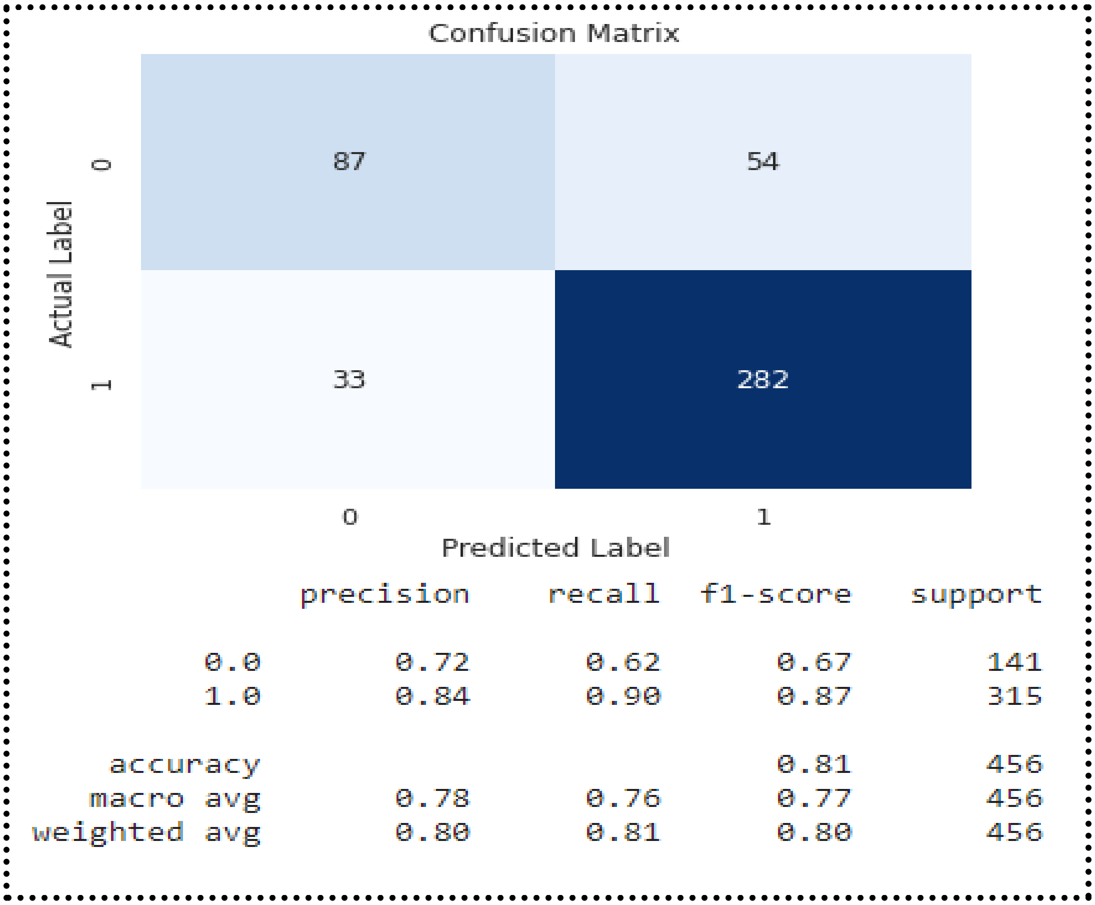
```
              precision    recall  f1-score   support

         0.0       0.84      0.79      0.81       319
         1.0       0.91      0.94      0.92       742

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

- *AUC and ROC curve for the training data:*



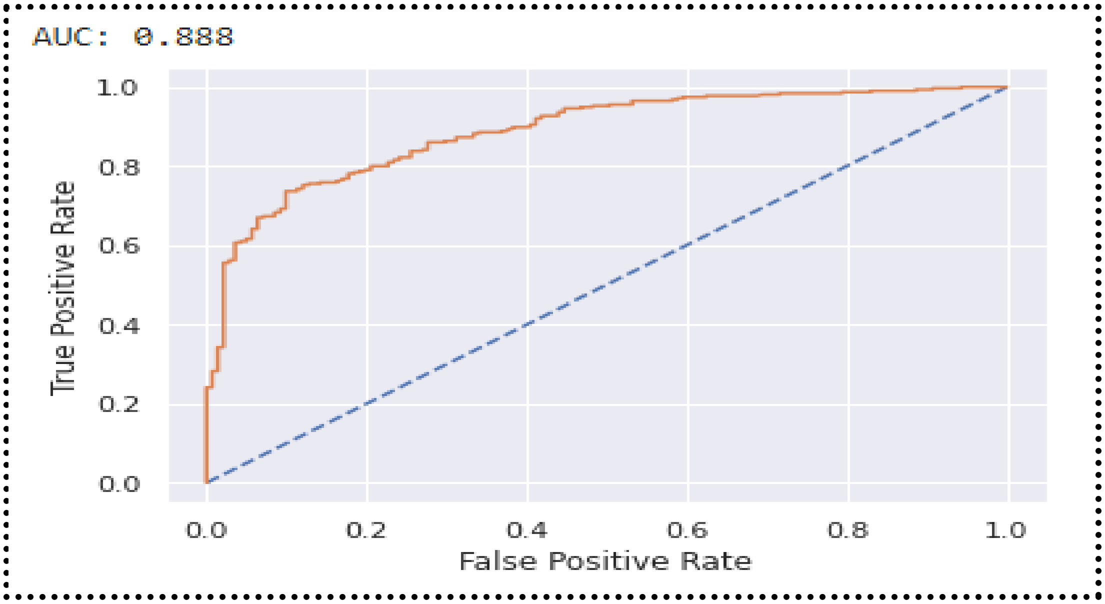
```
AUC: 0.952
```

Confusion Matrix & Classification_report Test data:



```
              precision    recall  f1-score   support

         0.0       0.72      0.62      0.67       141
         1.0       0.84      0.90      0.87       315

    accuracy                           0.81       456
   macro avg       0.78      0.76      0.77       456
weighted avg       0.80      0.81      0.80       456
```

AUC and ROC curve for the testing data:



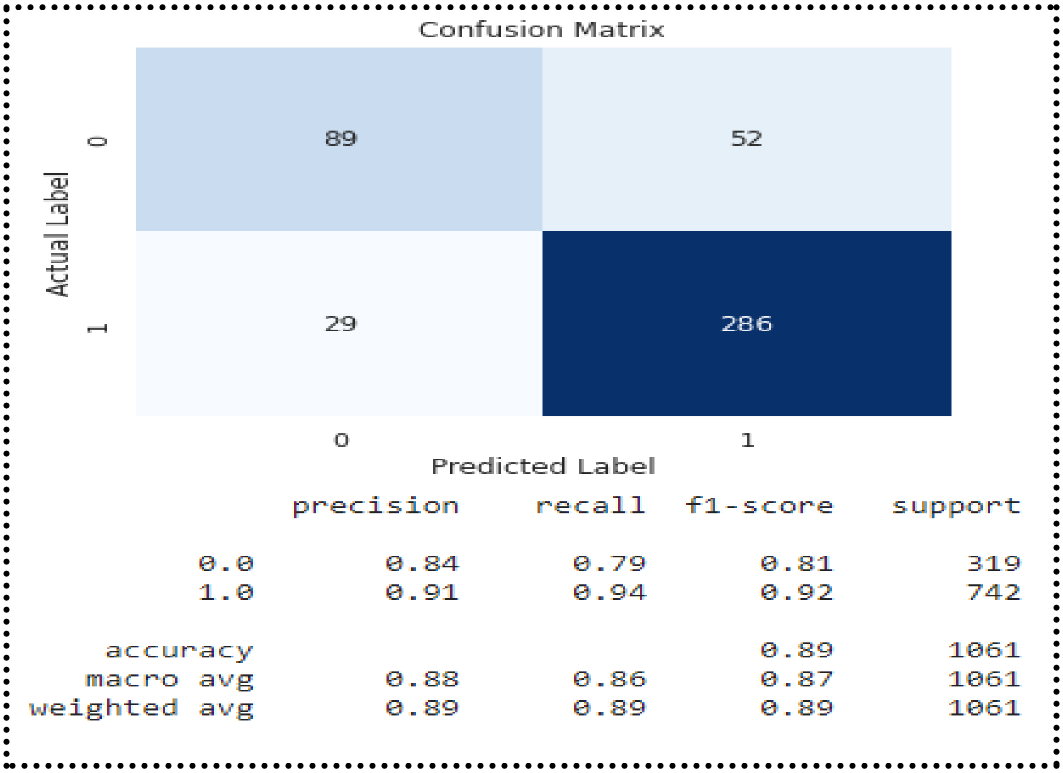
```
AUC: 0.888
```

- ***From the data above we can say that the model with default values.***

- *Accuracy from the Training data is: 0.952*
- *Accuracy from the Test data is: 0.888*

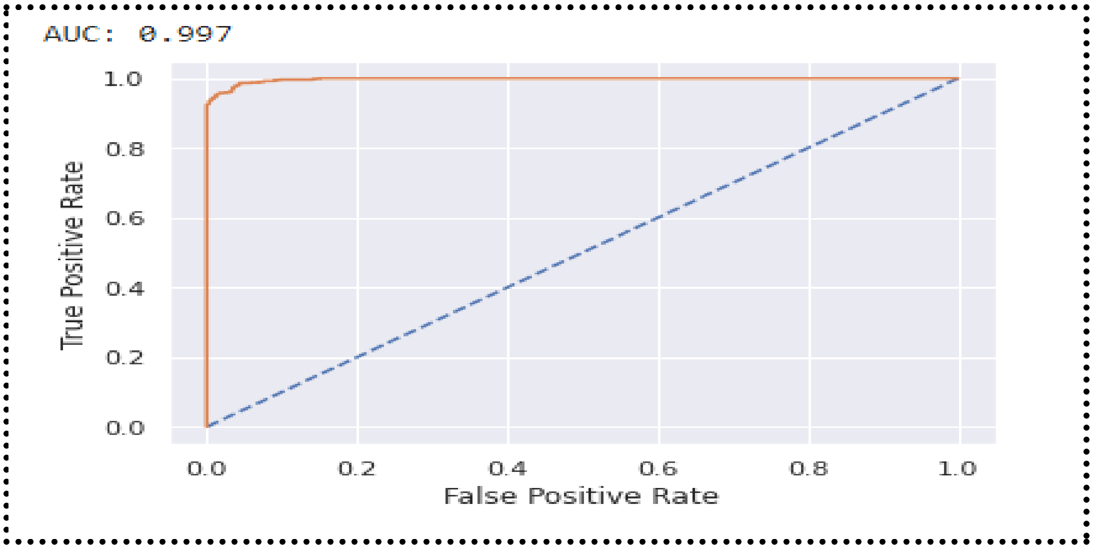
➢ *As we see, the accuracy of training and test results aren very close. The mode is not overfitting. It is a good model.*

Bagging with RF : Confusion Matrix & Classification_report Train data:



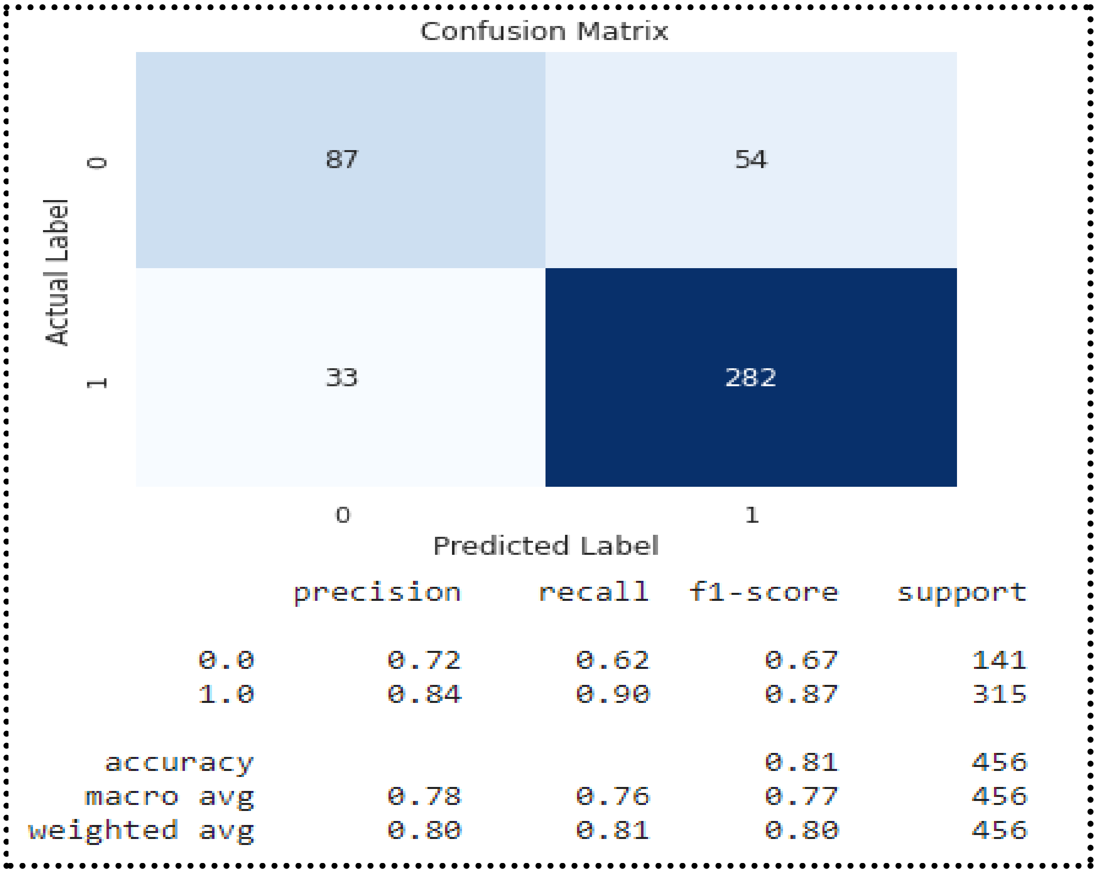
```
              precision    recall  f1-score   support

         0.0       0.84      0.79      0.81       319
         1.0       0.91      0.94      0.92       742

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

❖ *AUC and ROC curve for the training data:*



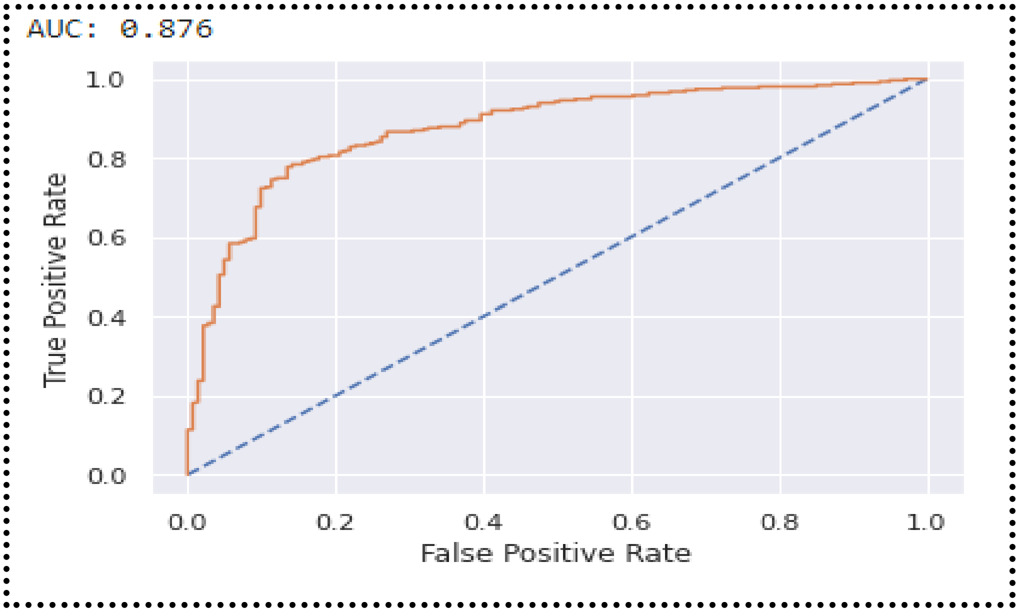
```
AUC: 0.997
```

Confusion Matrix & Classification_report Test data:



```
              precision    recall  f1-score   support

         0.0       0.72      0.62      0.67       141
         1.0       0.84      0.90      0.87       315

    accuracy                           0.81       456
   macro avg       0.78      0.76      0.77       456
weighted avg       0.80      0.81      0.80       456
```

AUC and ROC curve for the testing data:



```
AUC: 0.876
```

➢ ***From the data above we can say that the model with values.***

- ***Accuracy from the Training data is: 0.997***

- *Accuracy from the Test data is: 0.876*

➢ ***As we see, the accuracy of training and test results aren't very close. The mode is overfitting, but as it is ensemble. It is a good model.***

Table for Comparing Boost and Bagging model performance:

	Ada Boost Train	Ada Boost Test	Gradient Boosting Train	Gradient Boosting Test	Bagging model Train	Bagging model Test
Accuracy	0.84	0.82	0.95	0.81	0.89	0.81
AUC	0.82	0.91	0.89	0.88	0.997	0.876
Recall	0.90	0.90	0.94	0.90	0.91	0.90
Precision	0.87	0.85	0.91	0.84	0.94	0.84
F1 Score	0.89	0.87	0.92	0.87	0.92	0.87

Table 8. Table for Comparing Boost and Bagging model performance

➢ ***As we see, the model with the best result is:ADA Boost model. All the values in test and train are in the range of within(10%) variance and not more, the model can be called stable.***

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. Full marks should only be allotted if the recommendations are correct and business specific.

- ***Looking at the Logistic Regression and LDA model data we were able to predict better, 80% accuracy for test sets.***
- ***The models give us around 80 or close to 80% accuracy for both test and train model, though it still can to be improved and can be further tunned.***

- ***We see that more Femal voters happen to vote and support Party than Male voter in general.***
- ***In that Labour Party is more popular. The return point of the vote on the Conservative Party's favor is support for 'Eurosceptic sentiments'.***
- ***Party would benefit, using this view as the base of companion.***
- ***Male voters are few, regrading them and holding encouraging female target companion would be a good idea.***
- ***People who have more knowledge on Politics are voting when to compare.***
- ***Holding Party leader introduction and Parties' agenda would help gain trust and votes.***
- ***Currently, the Leading party is Labour.***
- *Need to focus more on age groups(20-40) and (80-90)*

Problem 2:

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**

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

- *Number of Characters:*
  - **President Franklin D. Roosevelt in 1941: 7571**
  - **President John F. Kennedy in 1961:** 7618
  - **President Richard Nixon in 1973:** 9991
- *Number of Characters:*
  - **President Franklin D. Roosevelt in 1941: 1536**
  - **President John F. Kennedy in 1961:** 1546
  - **President Richard Nixon in 1973:** 2028

- *Number of Lines:*
    - **President Franklin D. Roosevelt in 1941: 68**
    - **President John F. Kennedy in 1961:** 52
    - **President Richard Nixon in 1973:** 69

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

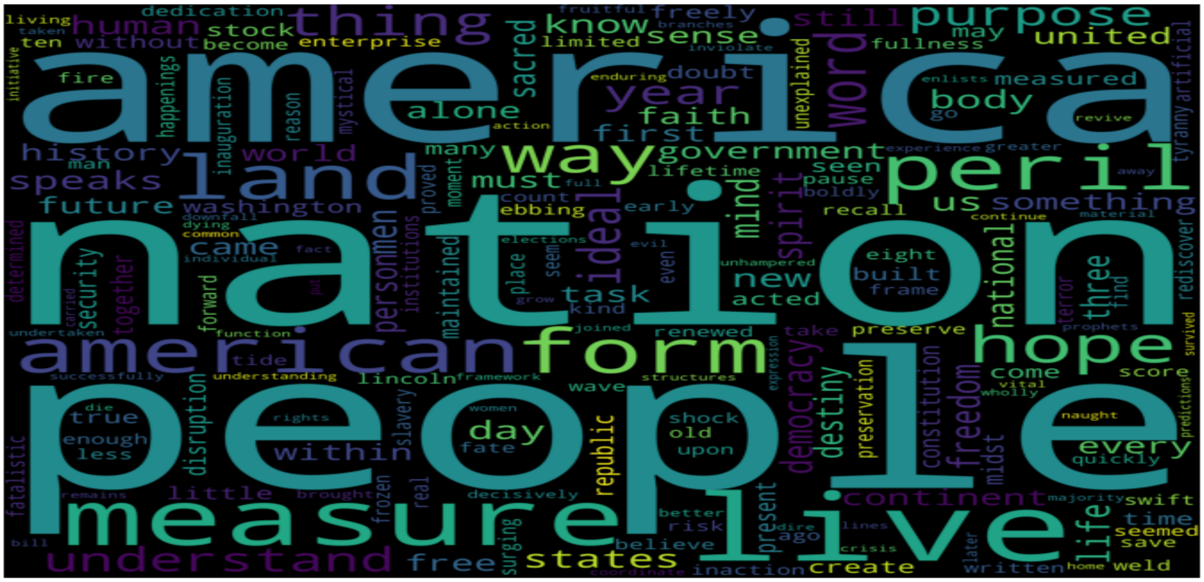
- *Frist 10 words before removing Stopwords:*
    - *['--', 'nation','know', 'spirit', 'life','democracy','us', 'people','america','years']*
- **First 10 words for 'President Franklin D. Roosevelt in 1941' after removing stopwards:**
    - *['nation','know', 'spirit', 'life','democracy','us', 'people','america','years', 'freedom']*
    - *On each national day of inauguration since 1789, the people have renewed their sense of dedication to the United States.\n\nIn Washington's day the task of the people was to create and weld together a nation*
- **First 10 words for 'President Franklin D. Roosevelt in 1941' after removing stopwards:**
    - *[us,'let','america','peace','world','new','nation','responsibility','government','great']*
    - *Mr. Vice President, Mr. Speaker, Mr. Chief Justice, Senator Cook, Mrs. Eisenhower, and my fellow citizens of this great and good country we share together: When we met here four years ago, America*
- **First 10 words for 'President John F. Kennedy in 1961' after removing stopwards:**
    - *['let','us','world','sides','new','pledge','citizens','power', 'shall','free']*
    - *Vice President Johnson, Mr. Speaker, Mr. Chief Justice, President Eisenhower, Vice President Nixon, President Truman, reverend clergy, fellow citizens, we observe today not a victory of party, but a celebration.*

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)

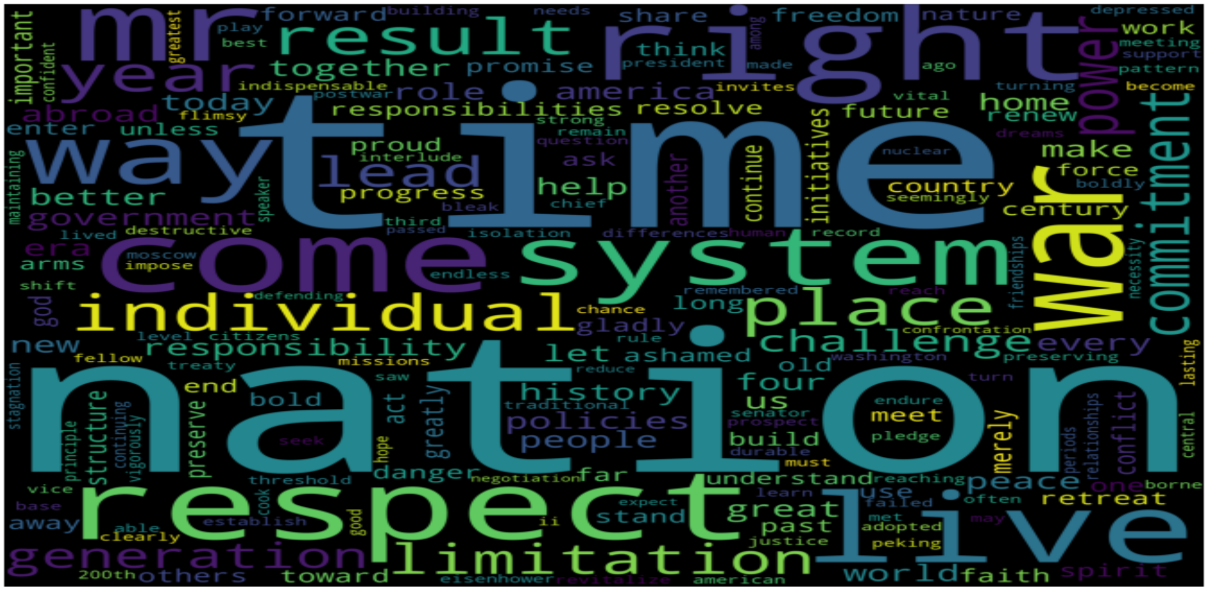
- **Top 3 words occurring the most in ‘President Franklin D. Roosevelt in 1941’ after removing stopwards:**
  - ➢ ***[('nation', 12), ('know', 10), ('spirit', 9)]***
- **Top 3 words occurring the most in ‘President Franklin D. Roosevelt in 1941’ after removing stopwords:**
  - ➢ ***[('us', 26), ('let', 22), ('america', 21)]***
- **Top 3 words occurring the most in ‘President John F. Kennedy in 1961’ after removing stopwords:**
  - ➢ ***[('let', 16), ('us', 12), ('world', 8)]***

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

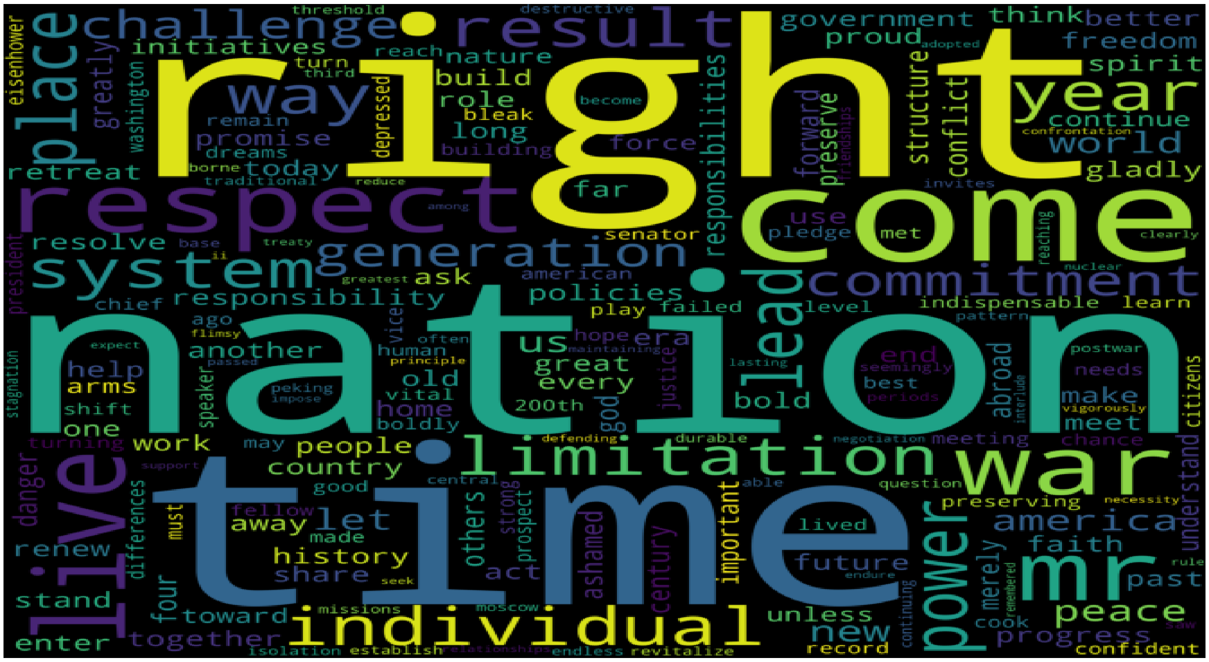
- **Word Cloud for ‘President Franklin D. Roosevelt in 1941’ after removing stopwords:**



- Word Cloud for ‘President Franklin D. Roosevelt in 1941’ after removing stopwords:



- Word Cloud for ‘President John F. Kennedy in 1961’ after removing stopwords:



END!