**Problem Statement 1*:***

**You are hired by one of the leading news channel 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.**

**Dataset for Problem: Election\_Data.xlsx**

* **Read the dataset. Do the descriptive statistics and do null value condition check.**
* First of all, the data Election\_Data.xlsx is read in python. Top 5 values of the matrix (df1) are shown below

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

* The dataset has 1525 rows and 9 columns (8 independent variables and 1 dependent)
* Following is the info report of the dataset specifying the data types of each column.

# 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

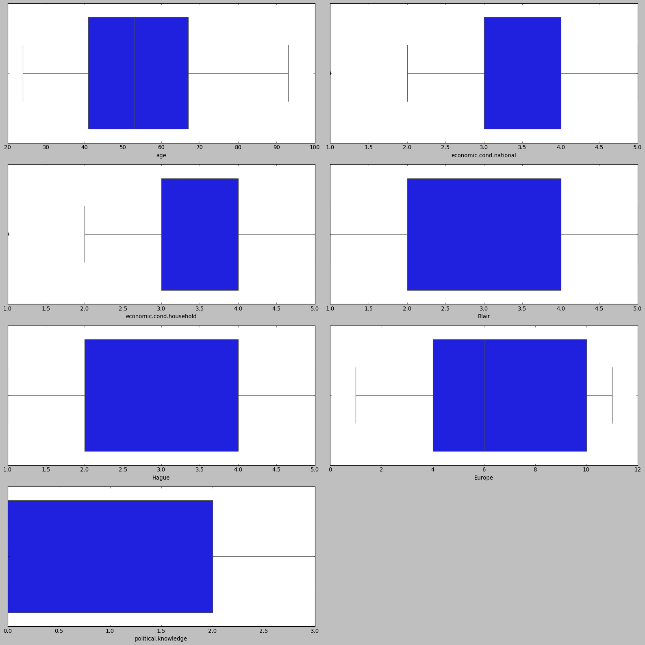
7 political.knowledge 1525 non-null int64

8 gender 1525 non-null object

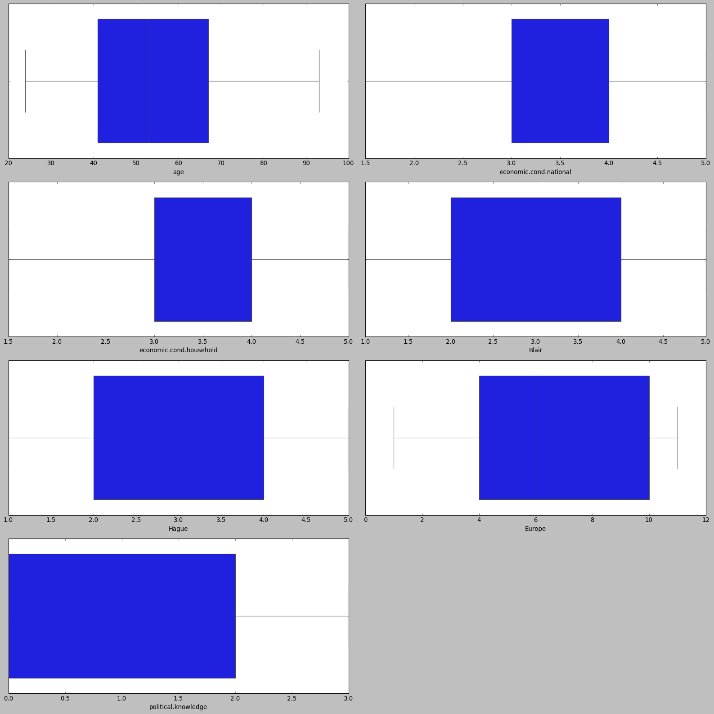
* Following is the descriptive matrix of the dataset df1

|  | **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 | NaN | NaN | NaN | 54.1823 | 15.7112 | 24 | 41 | 53 | 67 | 93 |
| **economic.cond.national** | 1525 | NaN | NaN | NaN | 3.2459 | 0.880969 | 1 | 3 | 3 | 4 | 5 |
| **economic.cond.household** | 1525 | NaN | NaN | NaN | 3.14033 | 0.929951 | 1 | 3 | 3 | 4 | 5 |
| **Blair** | 1525 | NaN | NaN | NaN | 3.33443 | 1.17482 | 1 | 2 | 4 | 4 | 5 |
| **Hague** | 1525 | NaN | NaN | NaN | 2.74689 | 1.2307 | 1 | 2 | 2 | 4 | 5 |
| **Europe** | 1525 | NaN | NaN | NaN | 6.72852 | 3.29754 | 1 | 4 | 6 | 10 | 11 |
| **political.knowledge** | 1525 | NaN | NaN | NaN | 1.5423 | 1.08331 | 0 | 0 | 2 | 2 | 3 |
| **gender** | 1525 | 2 | female | 812 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

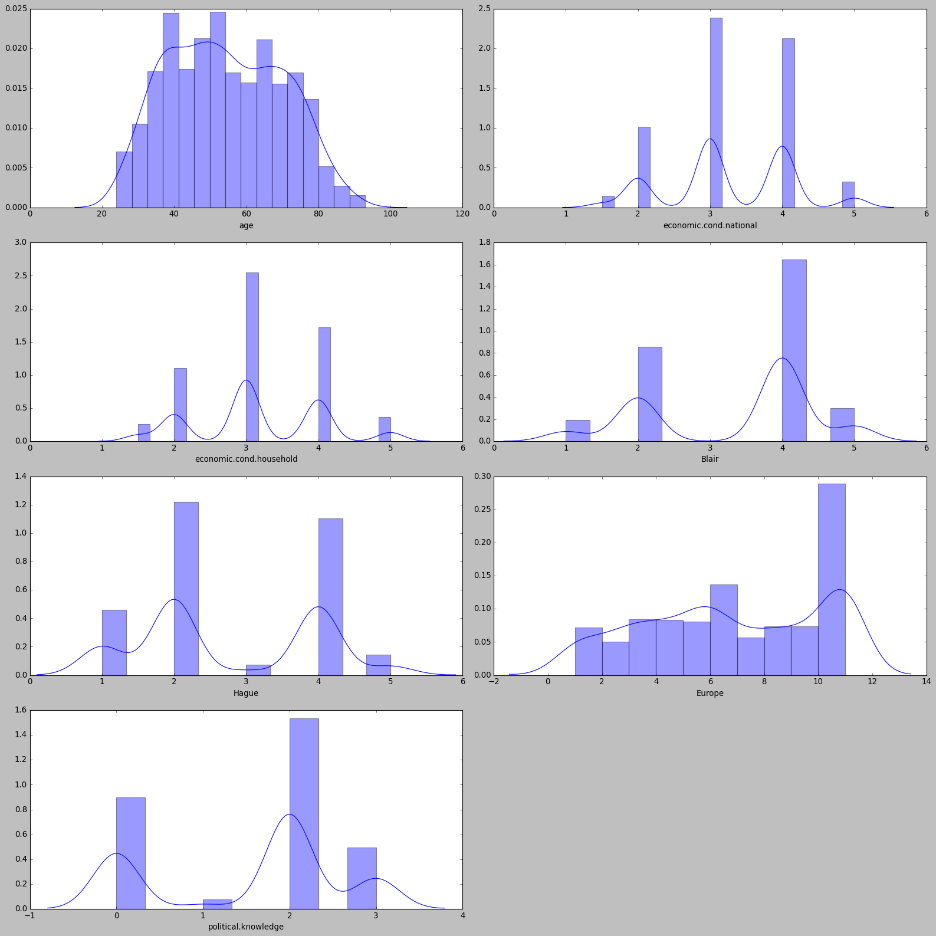
* There are no null values present in dataset
* Number of duplicate rows present in dataset is 8. We remove the duplicate rows
  1. **Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also check for outliers (4 pts). Interpret the inferences for each (3 pts)**
* Plotting a boxplot, we see that outliers are present in all numerical variables, which we can replace with lower/upper limit.



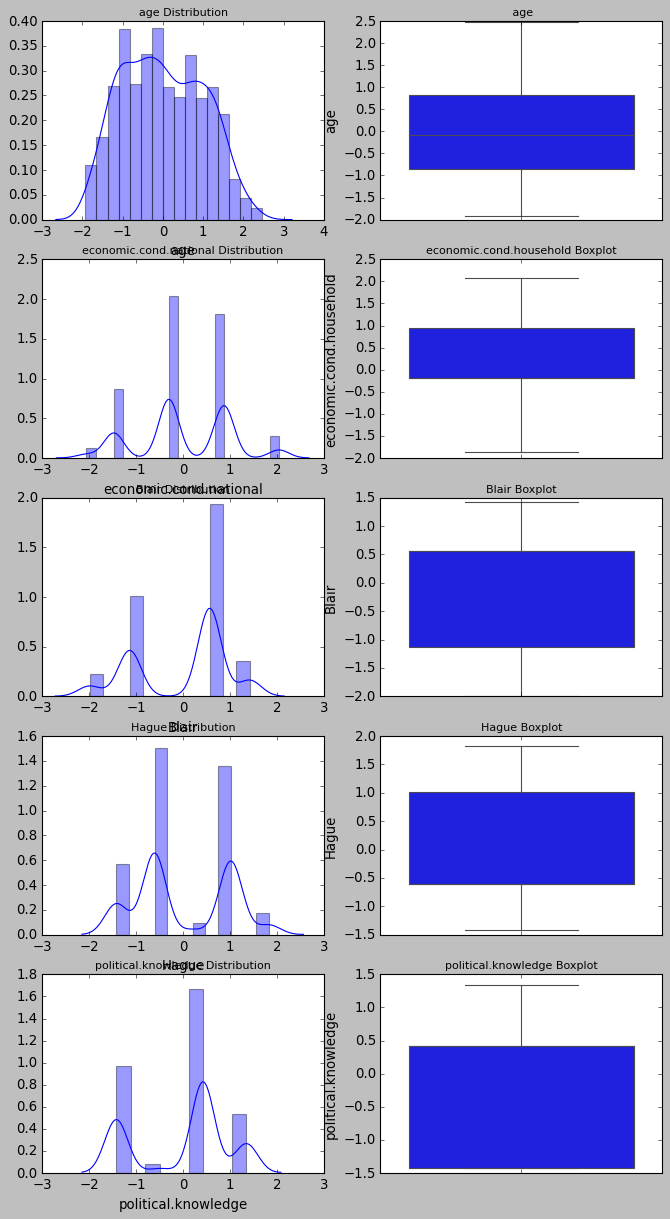
* Very small no. of Outliers are present, which we can replace by max/min limit. Following is the boxplot, in which replaced the outliers with upper/lower limit of respective variables.



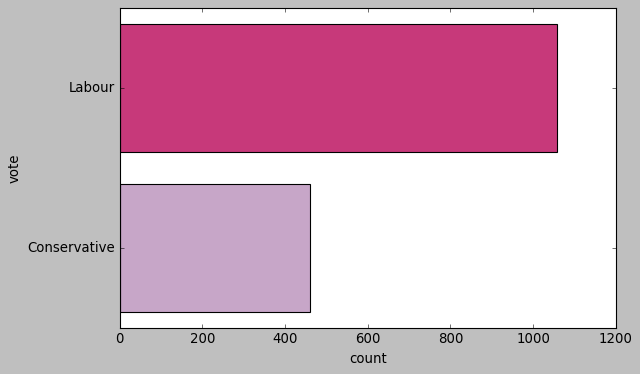
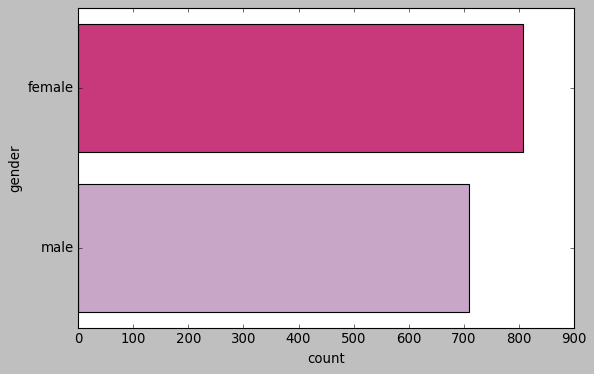
* Following is the histogram of df1



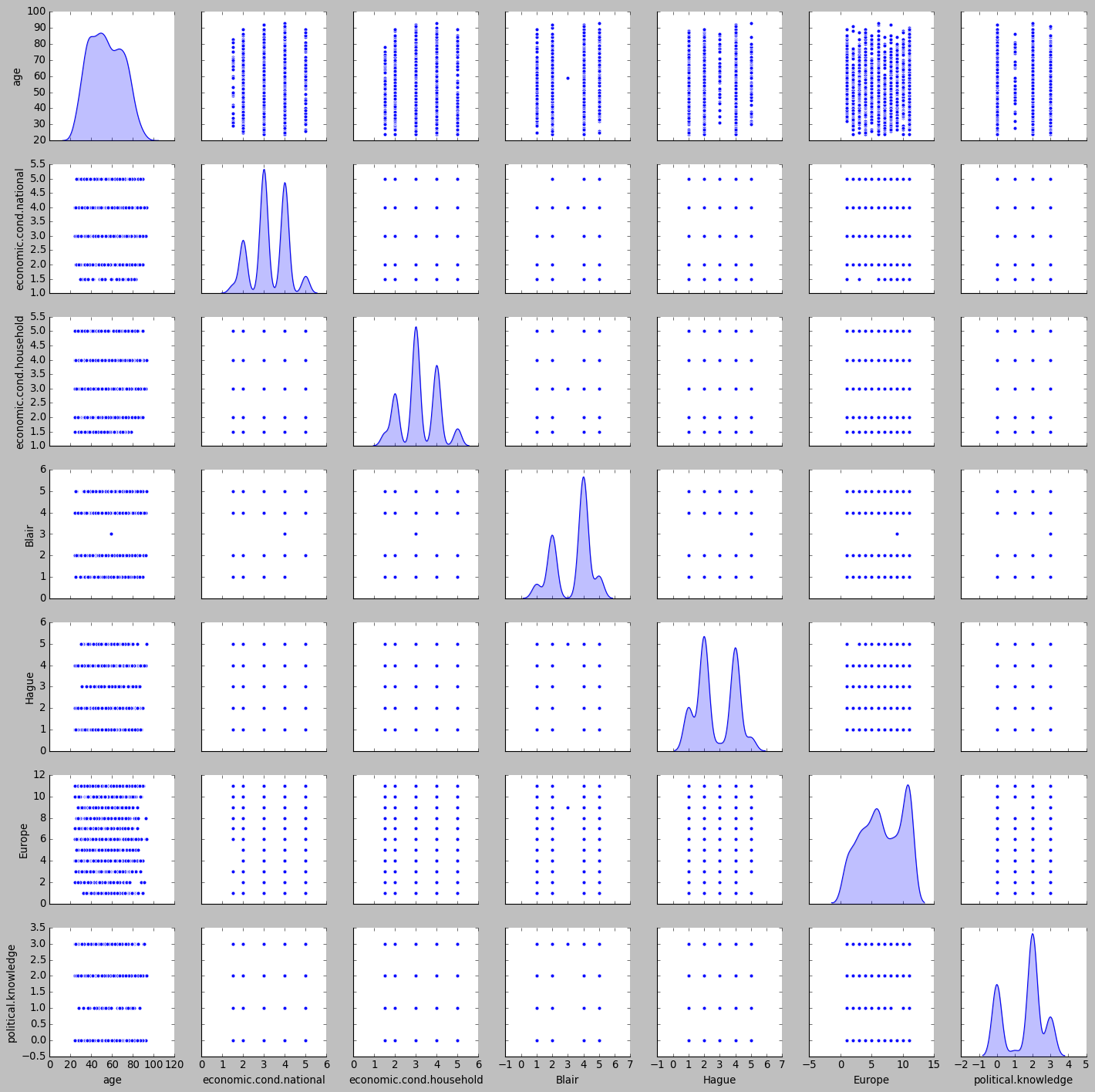
* Following is the univariate analysis of the numerical variables:



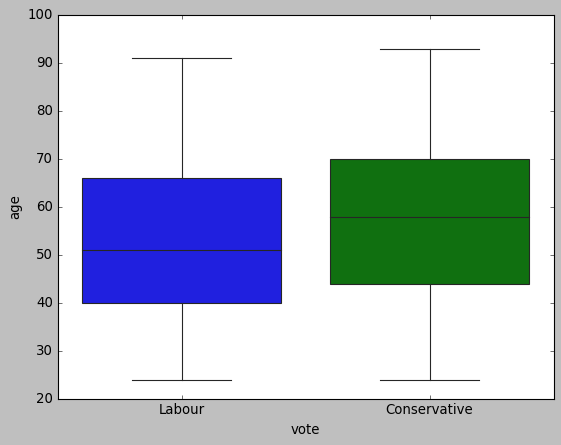
* From the above univariate analysis we can see that numerical plot of ‘age’ has more bins than other columns. So we can say that the other variables such as 'economic.cond.national', 'economic.cond.household', 'Blair', 'Hague', 'Europe', 'political.knowledge' are categorical variables with numeric labels.
* Following we can see the count of the categorical variables vote, gender

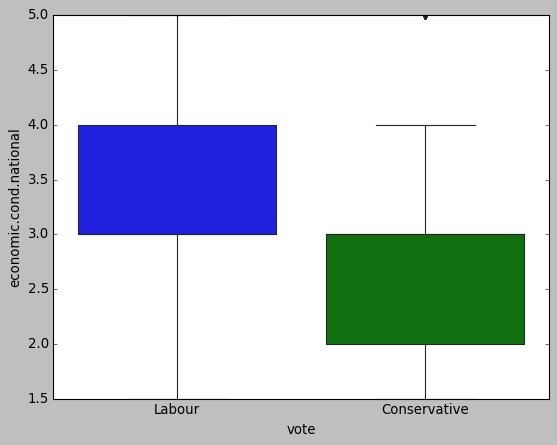
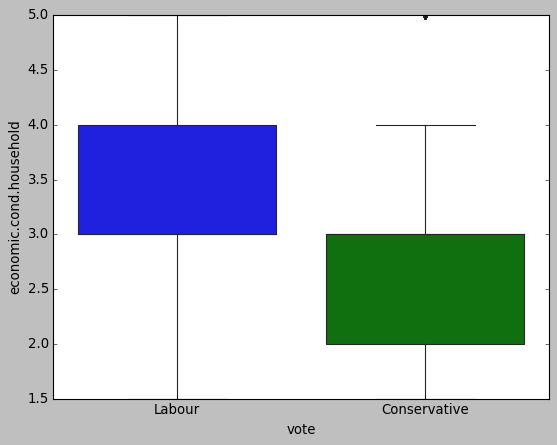
 

* From the following pair plot,

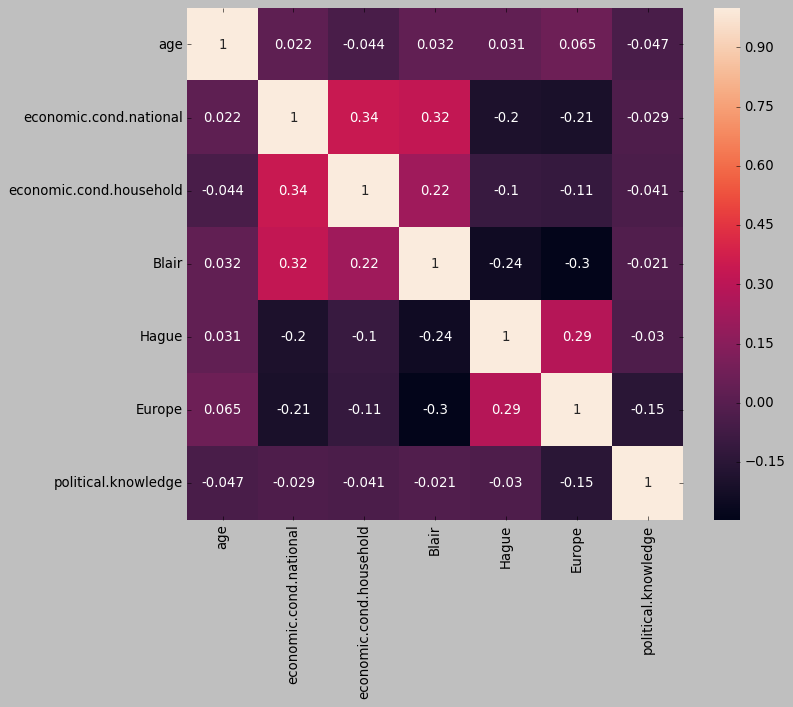


* From the multivariate analysis we can see that younger population prefers labour, while older population prefers conservative



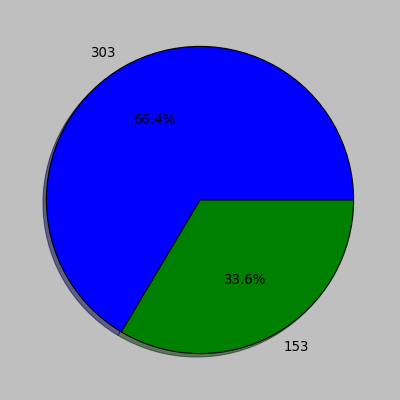
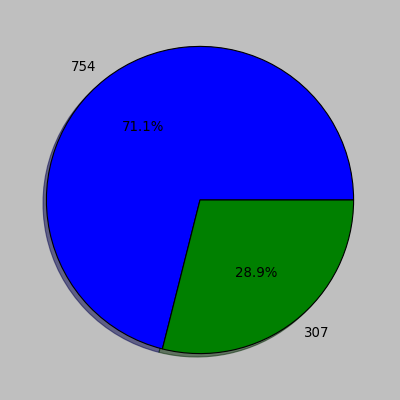
* Those with higher economic.cond. values tend to vote for Labour
* Following can be seen the heatmap the correlation values. There is a moderately high postive correlation among variables such as 'economic.cond.national', 'economic.cond.household','Blair' and small negative correlations among others



**1.3) Encode the data (having string values) for Modelling. Is Scaling necessary here or not? ( 3 pts), Data Split: Split the data into train and test (70:30) (2 pts).**

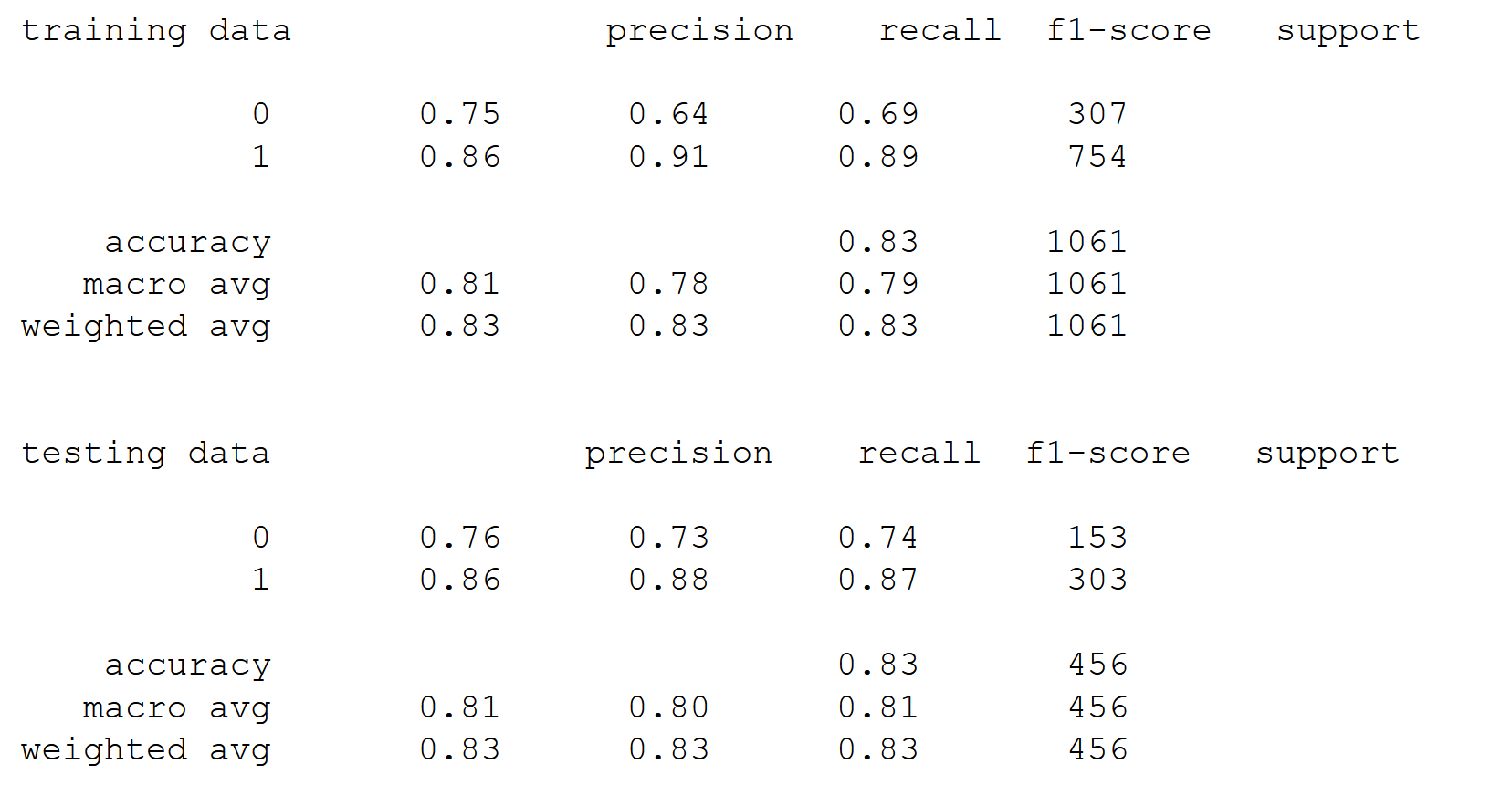
* As shown above, we treat the outliers in the model and then, as continuous variables have different weightages, we do scaling using standard scaler technique to get better results. Scaling is necessary here for this reason.
* Also,we encode the string values to get equivalent numerical results for use ahead.
* Overall data is split into train and test with test size as 0.3 and random state 1. vote\_Labour is taken as the dependent variable y.
* We see that in df1 the proportion of 0 and 1 in vote\_Labour is 30%, 70% respectively.

In y\_train also the split for 1 and 0 is 71, 29 respectively and in y\_test also the proportion is 67 and 33 respectively. Hence we don’t need to worry about any imbalanced dataset problem

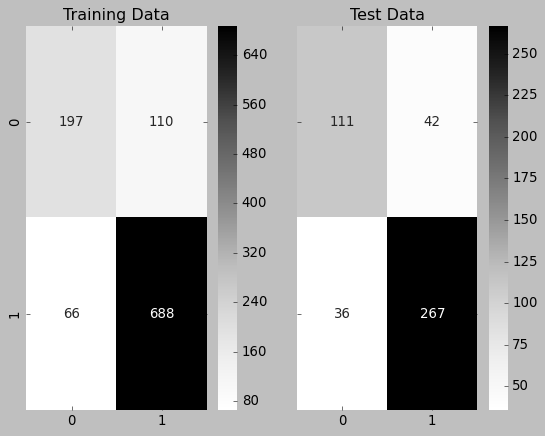
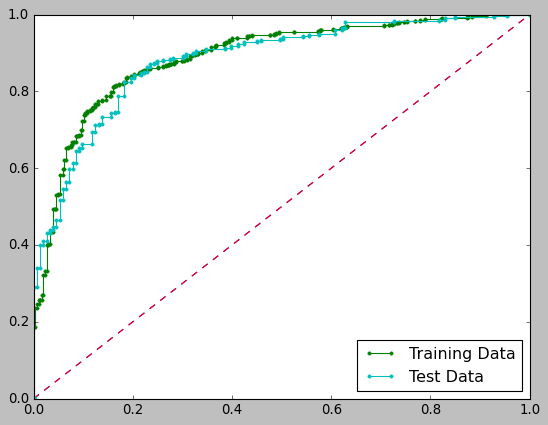
****

**1.4) Apply Logistic Regression and LDA (Linear Discriminant Analysis) (3 pts). Interpret the inferences of both models (2 pts)**

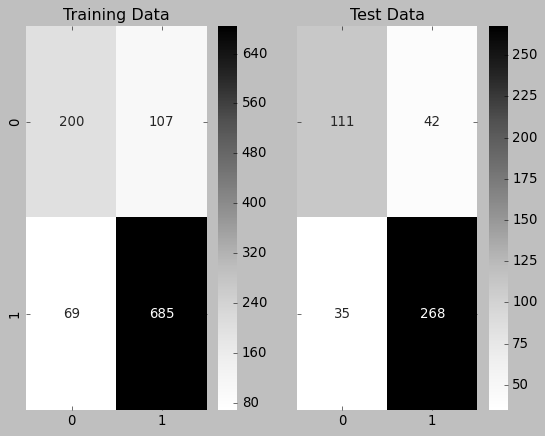
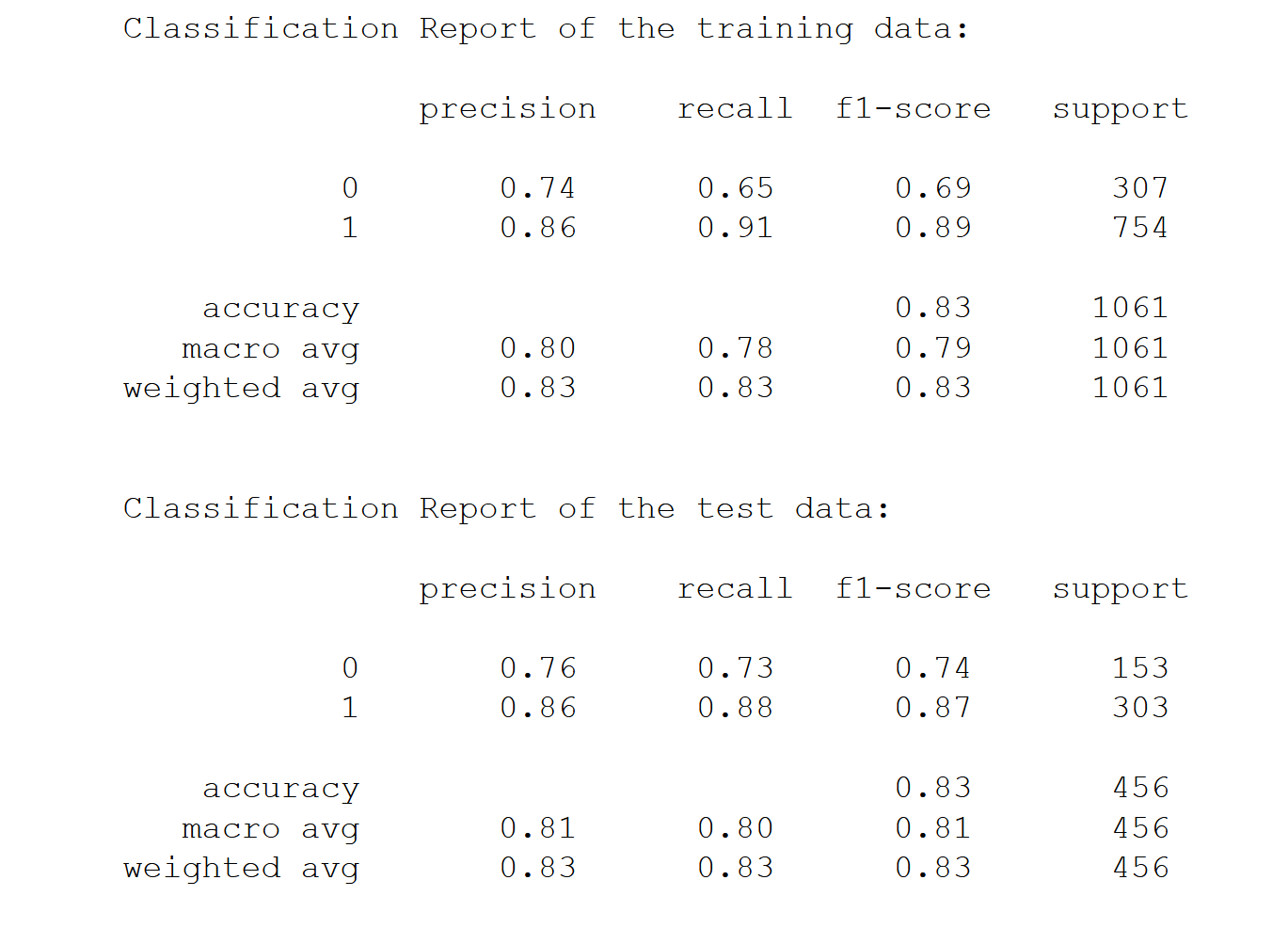
* Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable.
* Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative distribution function of logistic distribution.
* Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1".
* 1 is for labour party and 0 is for conservative party.
* Interest Class is 1 i.e. vote\_Labour = 1.
* We get the following classification matrix



* Accuracy of Logistic regression classifier is test set 0.83 and recall for test set is 0.88. So, testing data of logistic regression is able to predict the true cases of voting results by 86% accuracy.
* We get the following confusion matrix and AUC ROC curve of Logistic regression classifier

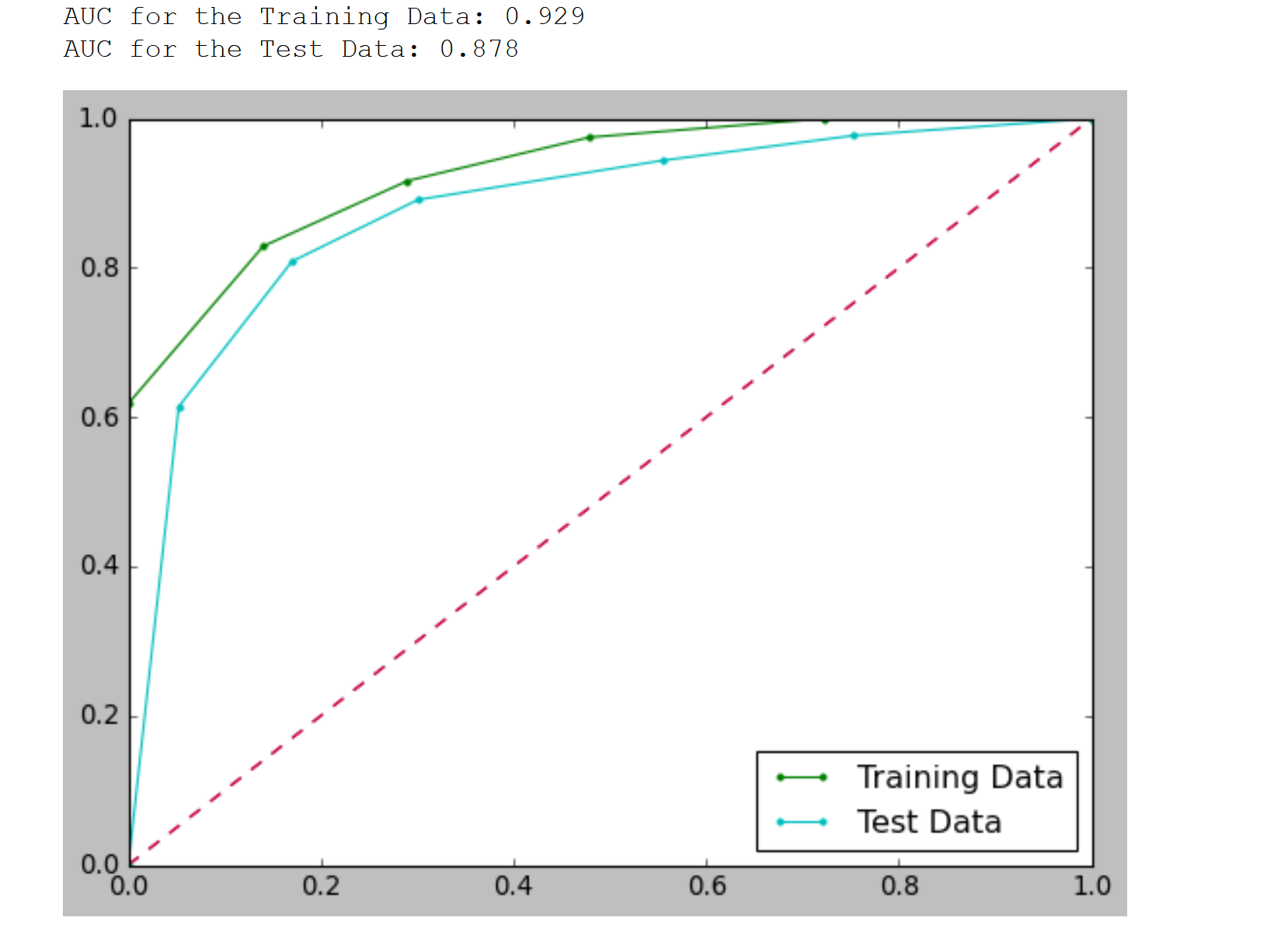
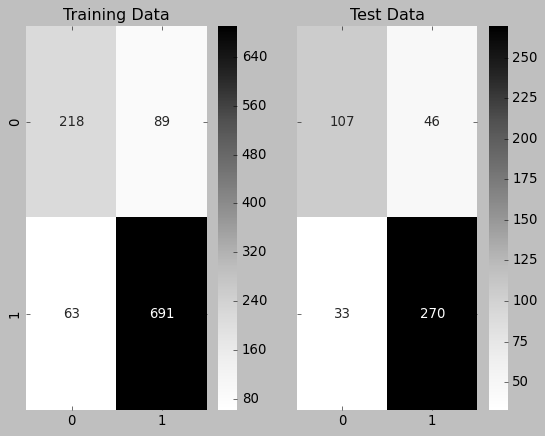
* Recall refers to the percentage of total relevant results correctly classified by the algorithm and hence we will compare Recall of class "1" for all models.
* Also, we apply LDA
* Linear Discriminant Analysis or LDA is a dimensionality reduction technique. The goal of LDA is to project the features in higher dimensional space onto a lower-dimensional space in order to avoid the curse of dimensionality and also reduce resources and dimensional costs.
* 1 is for labour party and 0 is for conservative party.
* We get the following classification matrix and confusion matrix of LDA

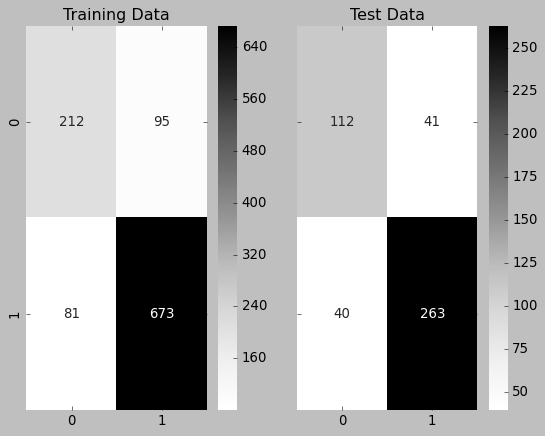
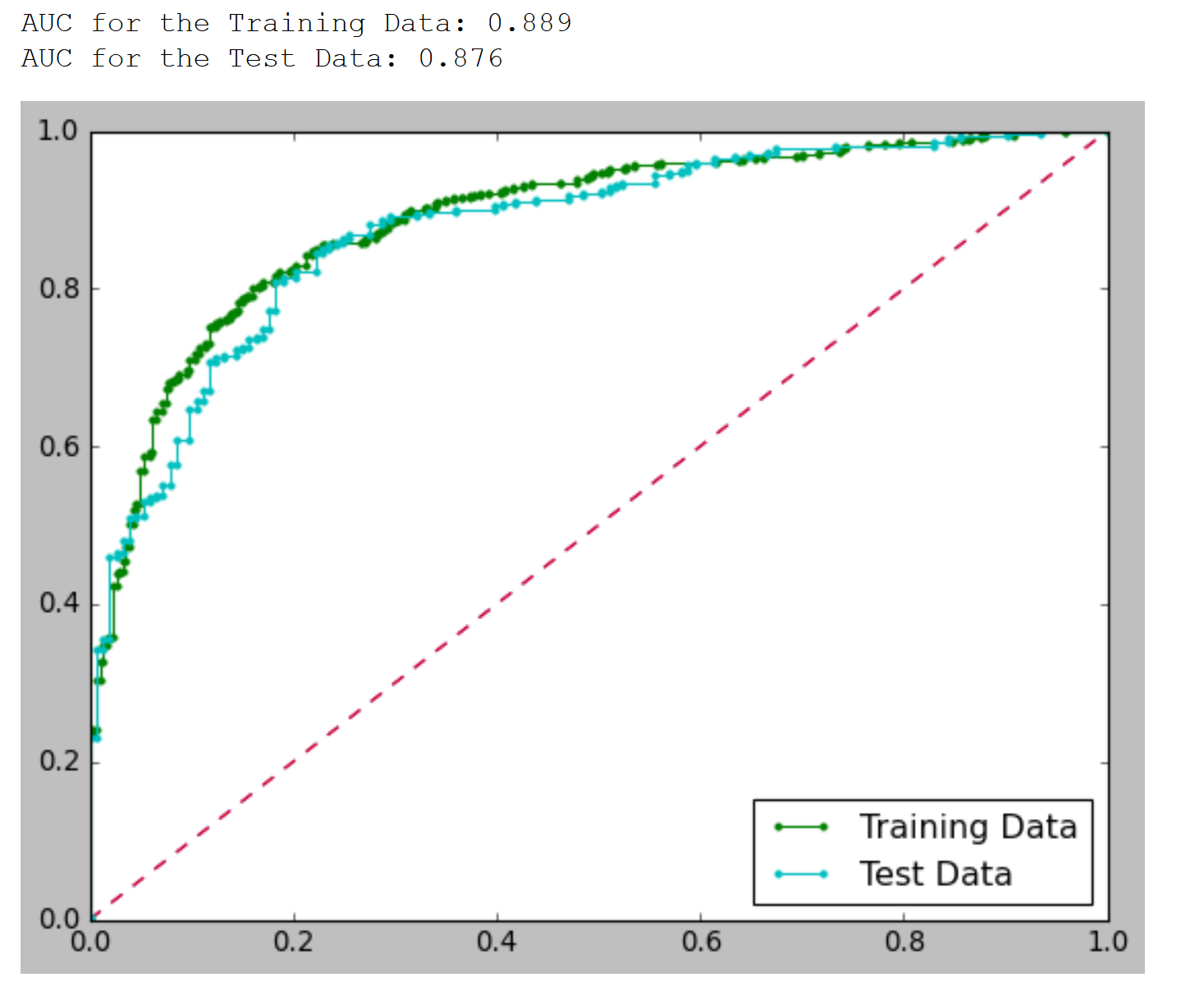
* Accuracy of LDA is test set 0.86 and recall for test set is 0.88. So, testing data of logistic regression is able to predict the true cases of voting results by 86% accuracy.
* Recall for LDA model is higher than that of logistic regression model. So LDA is better able to predict the true cases.

**1.5) Apply KNN Model and Naïve Bayes Model (5 pts). Interpret the inferences of each model (2 pts)**

* Below is the confusion matrix and AUC-ROC curve for KNN model:

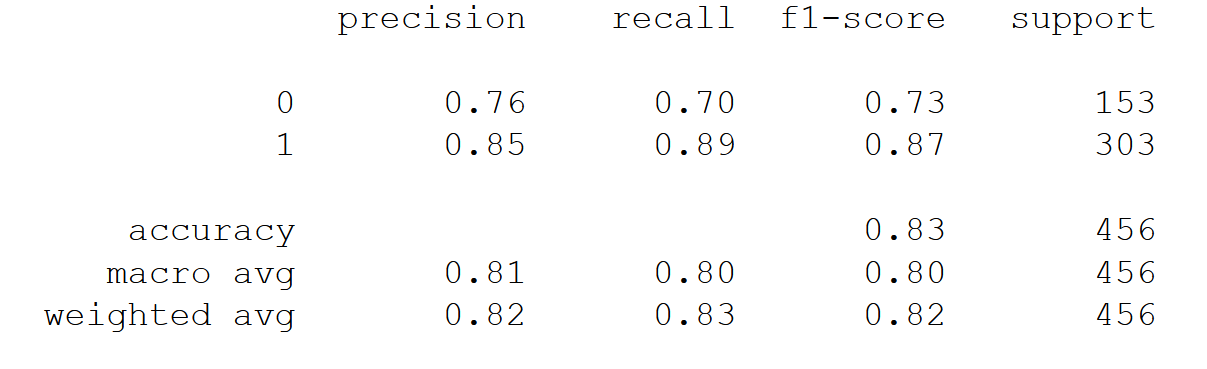


* Below is the confusion matrix and AUC-ROC curve for Naïve Bayes model:

* Recall refers to the percentage of total relevant results correctly classified by the algorithm and hence we will compare Recall of class "1" for all models.
* Following is the classification matrix for train and test sets of KNN model



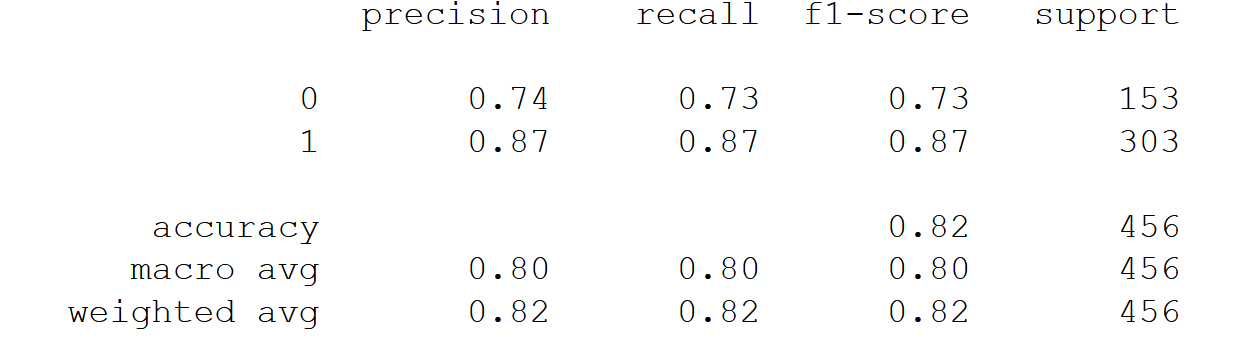


* Applying the KNN model we get a train accuracy of 0.86 and test accuracy 0.83. Hence, we can say that KNN model is not overfitted as the values are close to each other.

Recall for class “1” in train set is 0.92 and Recall for class “1” in test set is 0.92

* Following is the classification matrix for train and test sets of the Naïve Bayes model





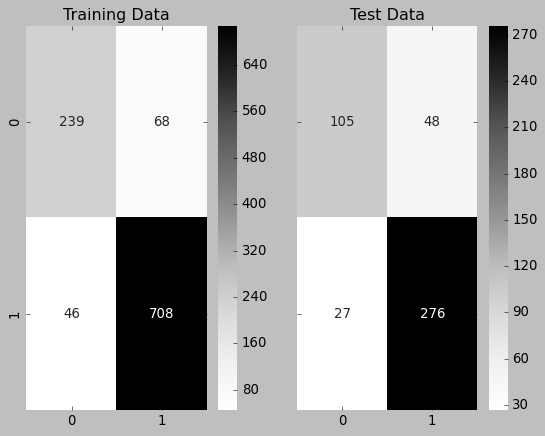
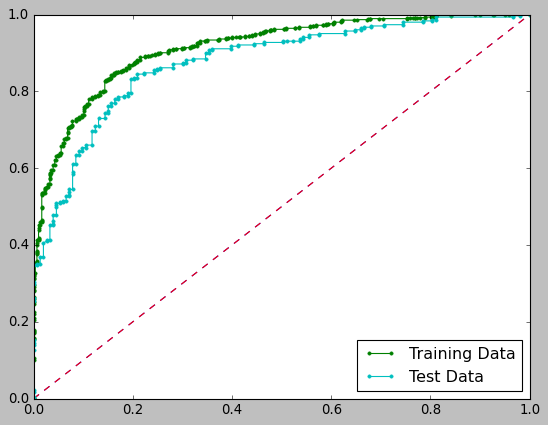
* Applying the Naïve Bayes model, we get a train accuracy of 0.83 and test accuracy 0.82.

Recall for class “1” in train set is 0.89 and Recall for class “1” in test set is 0.87

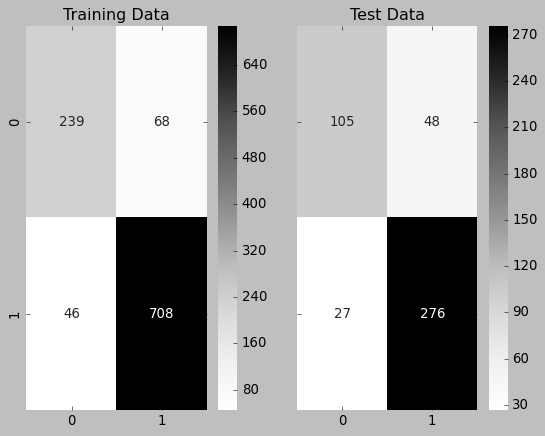
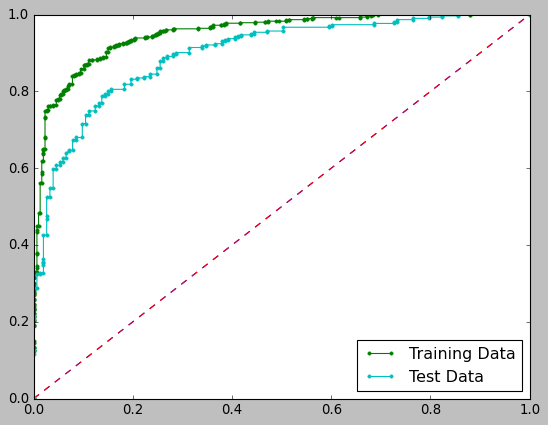
* Hence we can infer that KNN model performed better at correctly classifying relevant results.

**1.6) Model Tuning (2 pts), Bagging (2.5 pts) and Boosting (2.5 pts).**

* Model tuning basically refers to increasing the accuracy by changing the hyperparameters.
* The random forest algorithm relies on a parallel ensemble method called "bagging" to generate its weak classifiers. Bagging is a colloquial term for bootstrap aggregation. Bootstrap aggregation is a powerful method that allows us to decrease the variance of an estimate by averaging multiple estimates that are measured from random subsamples of a population. Bagging and random forests are algorithms that aim to reduce the complexity of models that overfit the training data.
* Model which have not performed well on the train data set, also have not performed well on the test data set However Random Forest which had a 100% score on the train data set has shown a poor result on the test data set .. Hence a clear case of overfitting.
* The term ‘Boosting’ refers to a family of algorithms which converts weak learner to strong learners. Boosting is an approach to increase the complexity of models that suffer from high bias,
* We use Adaboost and Gradient Boost in this question.
* AdaBoost is an ensemble learning method (also known as “meta-learning”) which was initially created to increase the efficiency of binary classifiers. AdaBoost uses an iterative approach to learn from the mistakes of weak classifiers, and turn them into strong ones.
* Following are the confusion matrix and AUC ROC for Adaboost

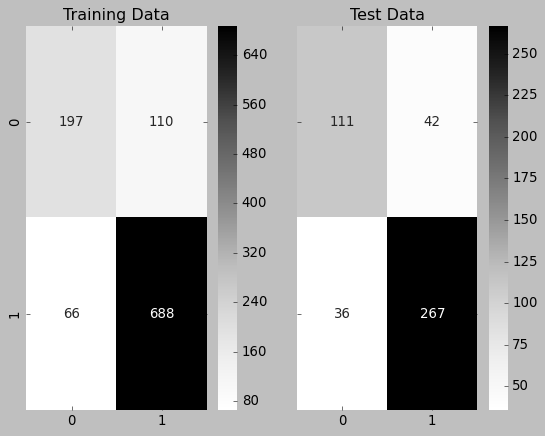
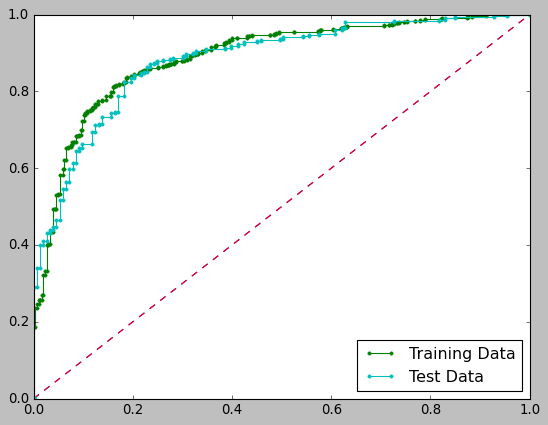
 

* Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.
* Following are the confusion matrix and AUC ROC for Gradient boosting

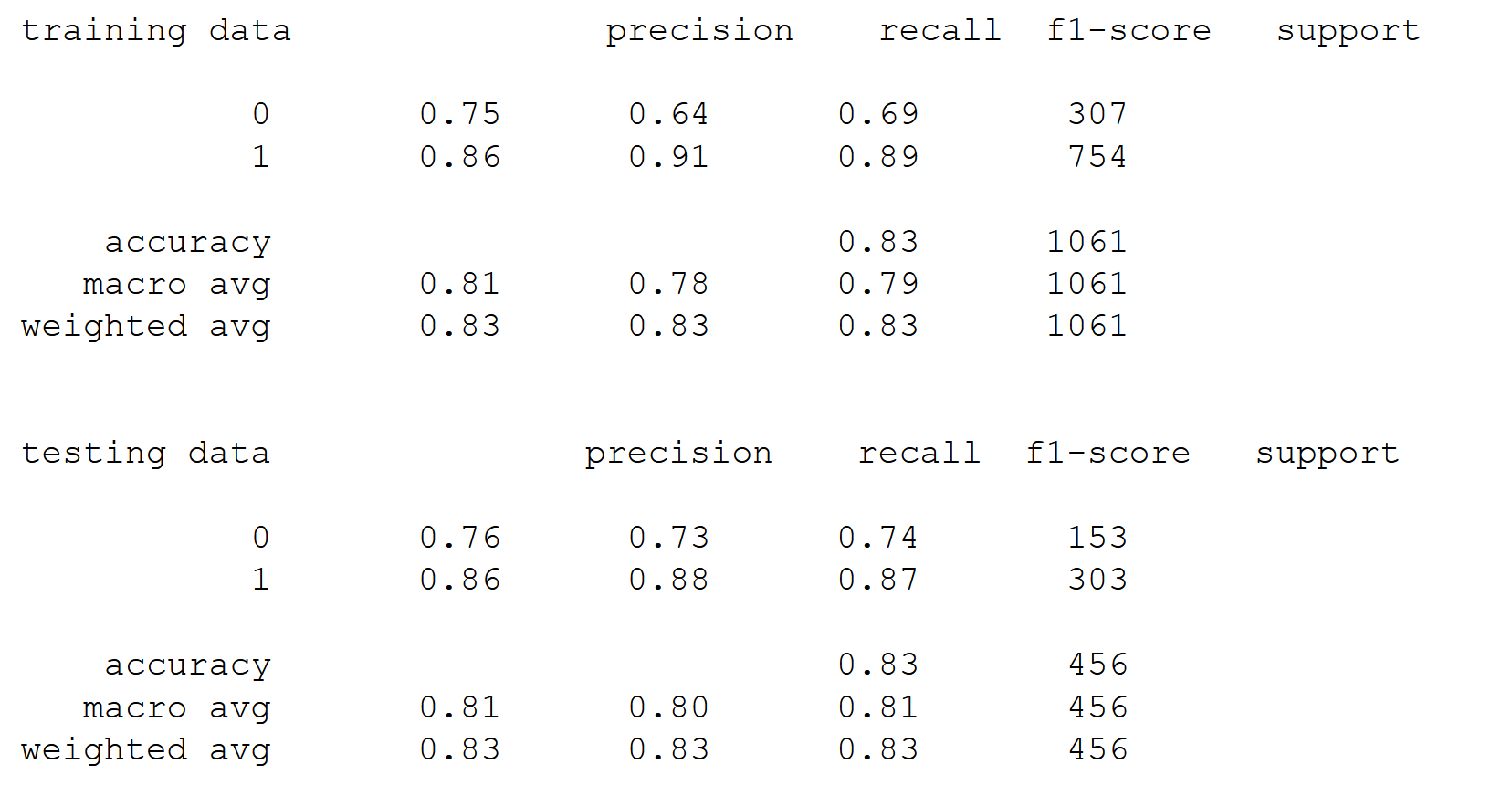
 

**1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model (4 pts) Final Model - Compare all models on the basis of the performance metrics in a structured tabular manner. Describe on which model is best/optimized (3 pts)**

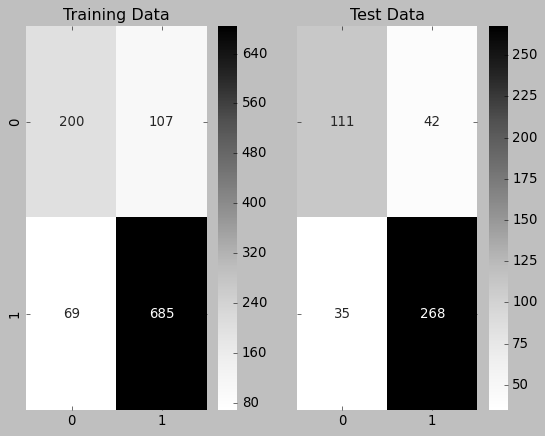
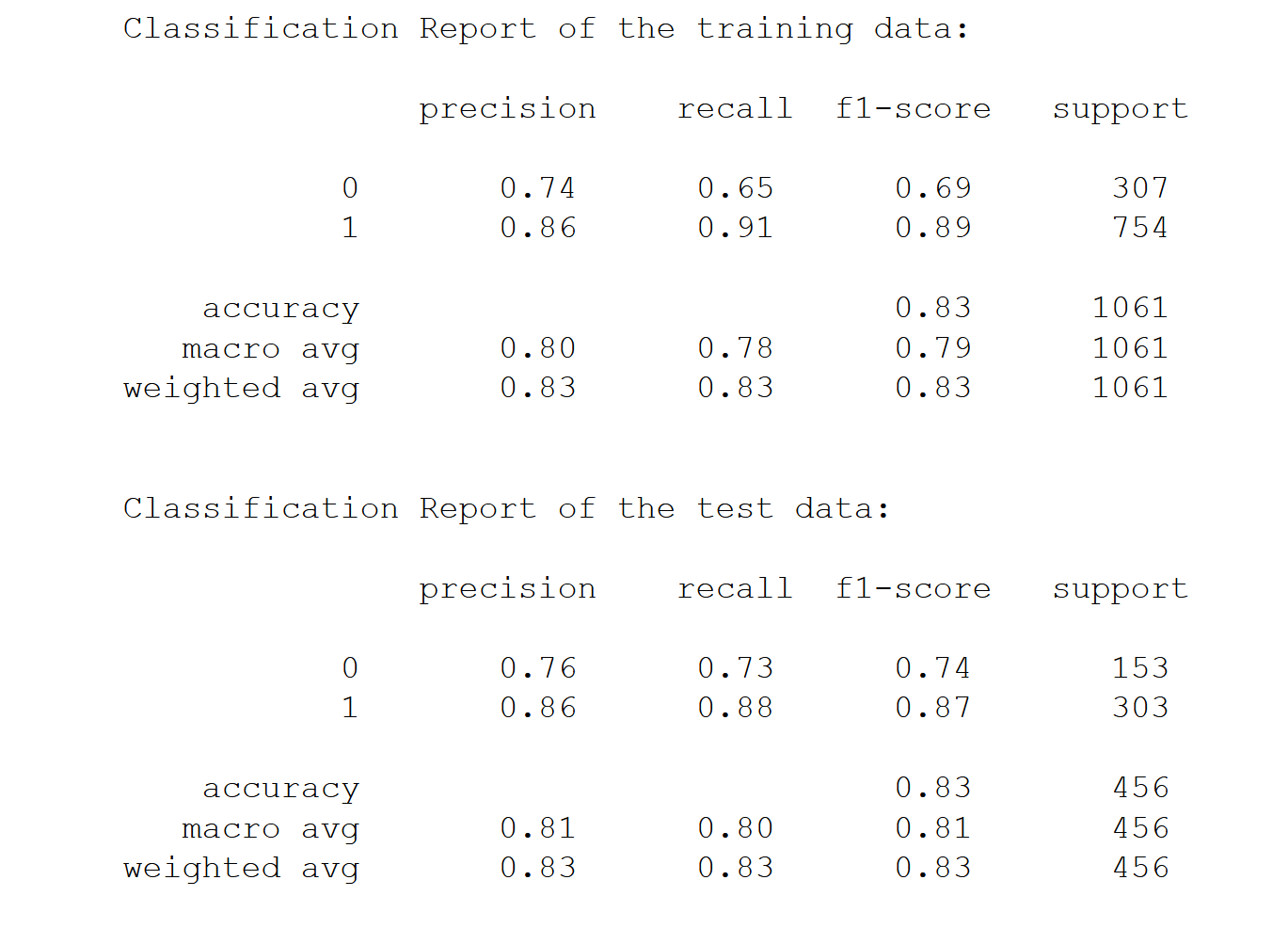
* We get the following confusion matrix and AUC ROC curve of Logistic regression classifier

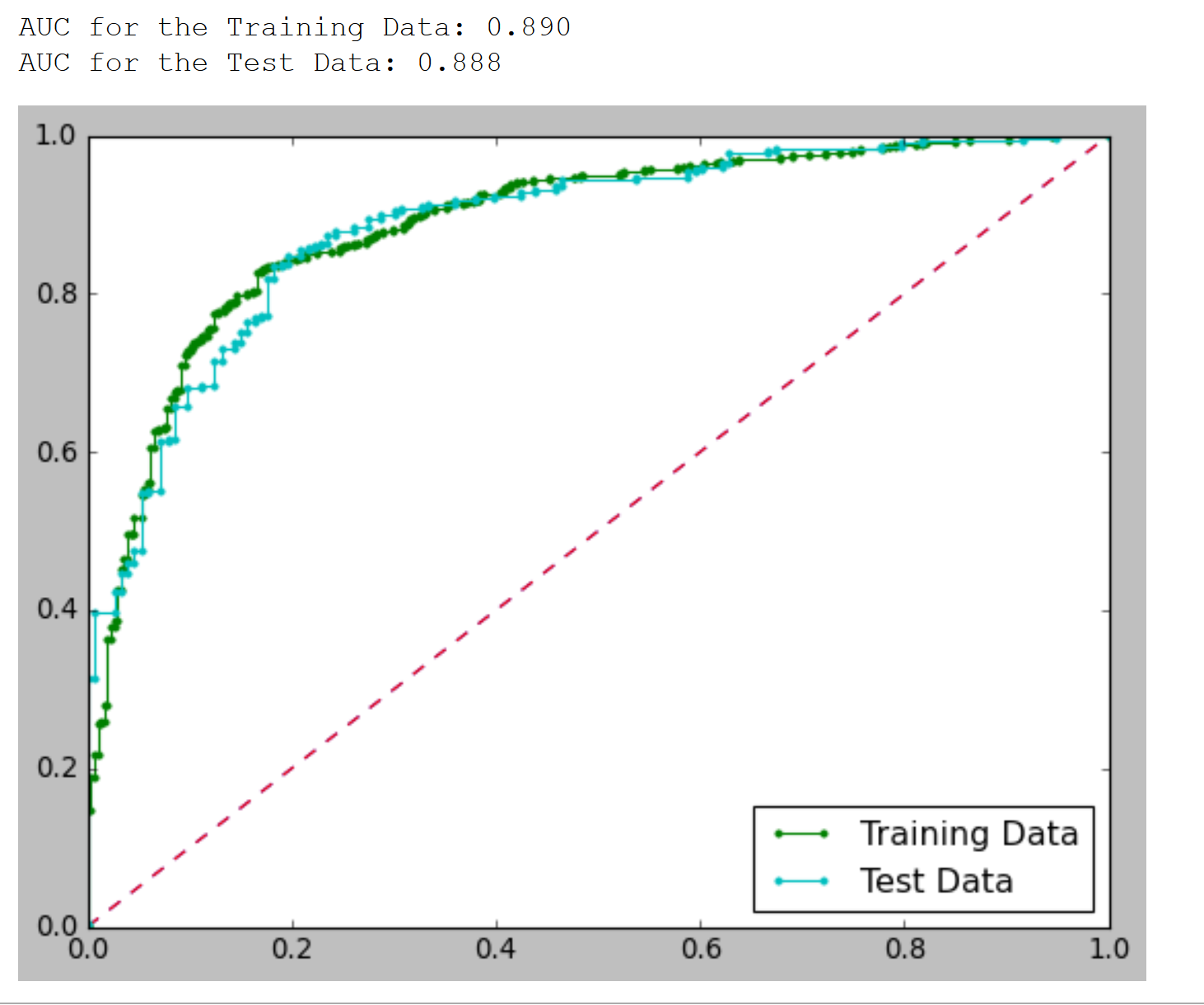
* the following classification matrix of Logistic regression classifier



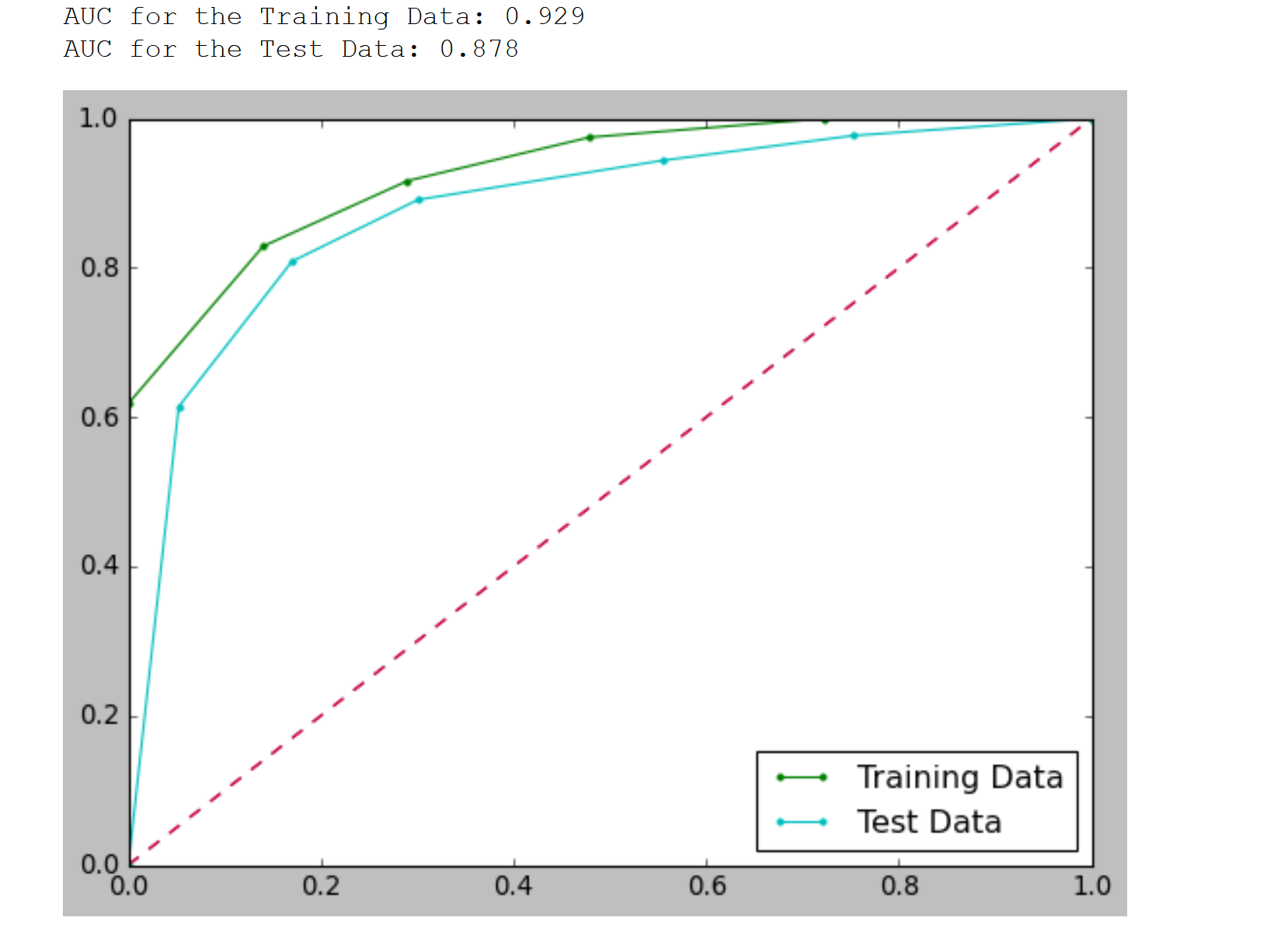
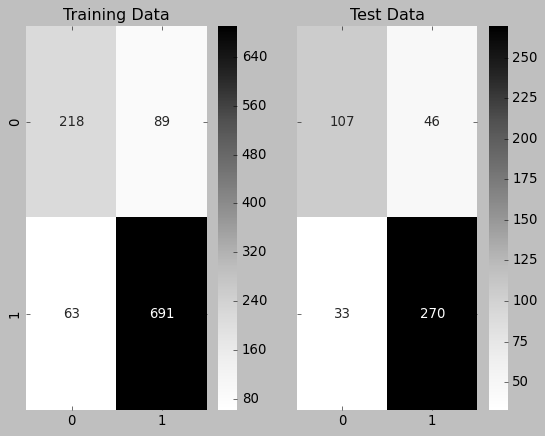
* We get the following classification matrix and confusion matrix of LDA

* We get the following AUC ROC curve of LDA

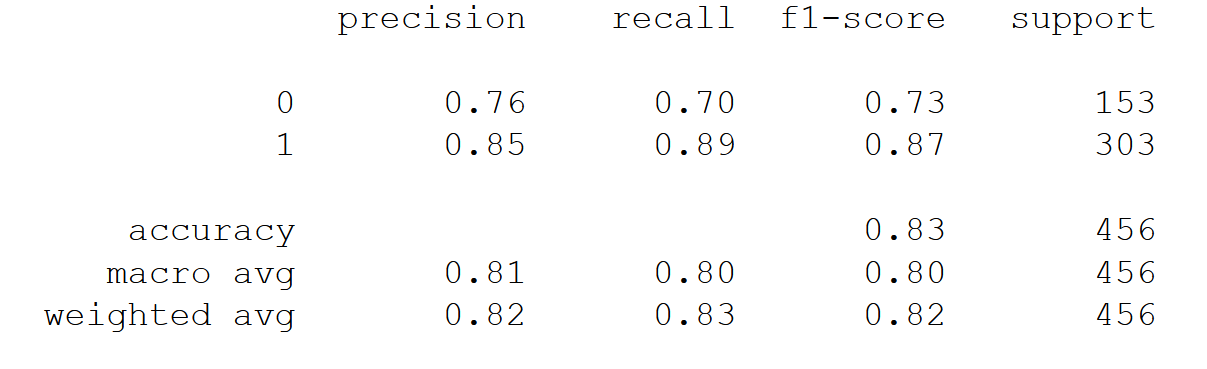


* Below is the confusion matrix and AUC-ROC curve for KNN model:

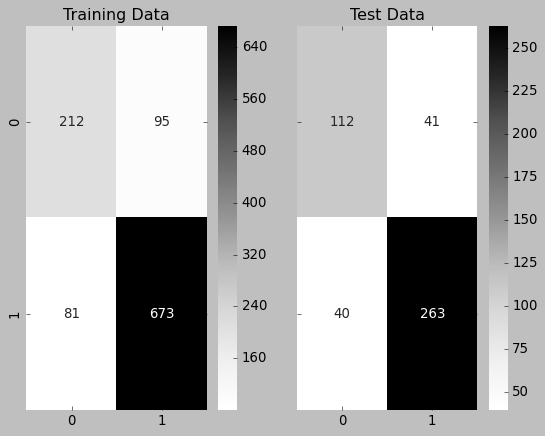
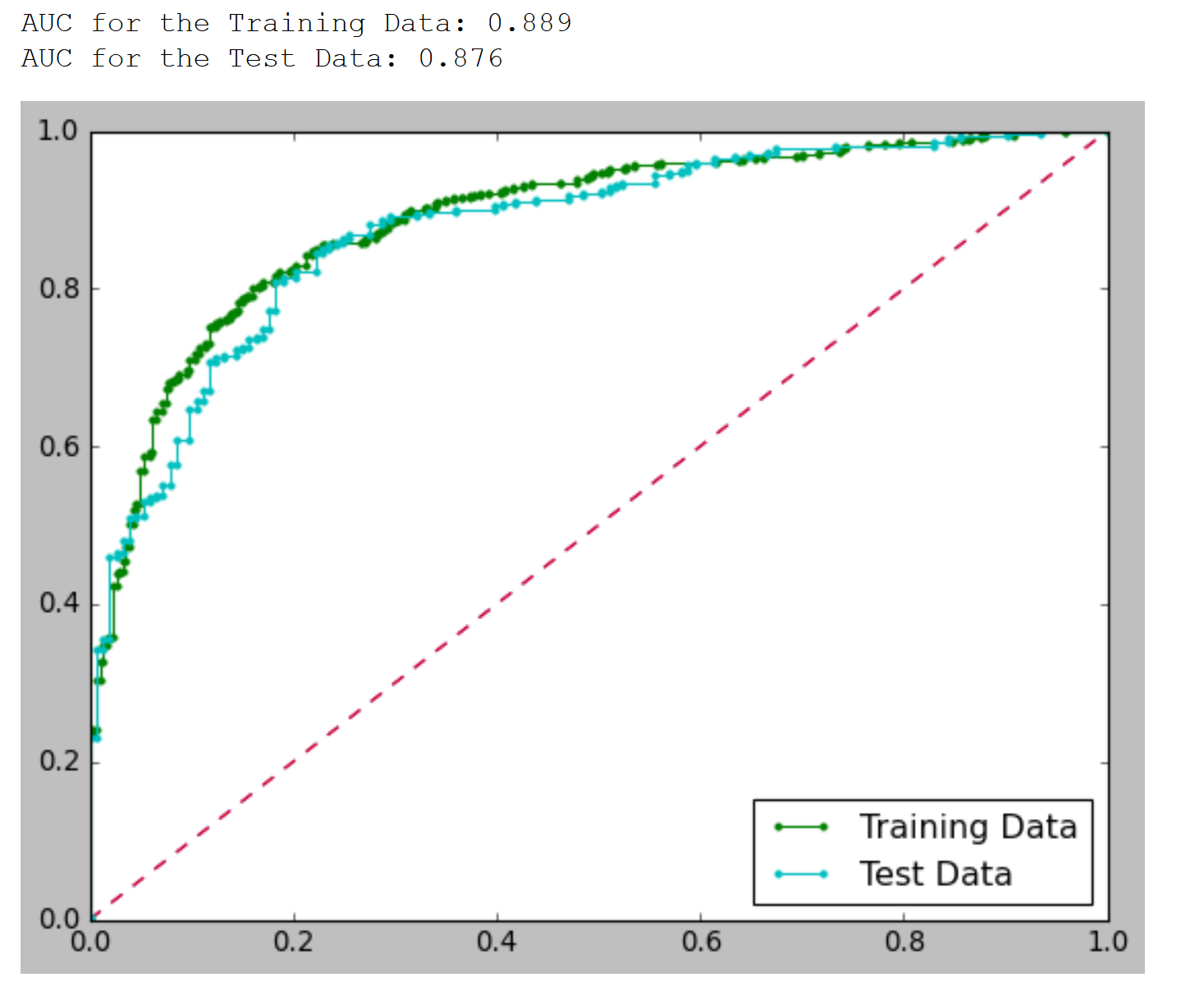


* Following is the classification matrix for train and test sets of KNN model



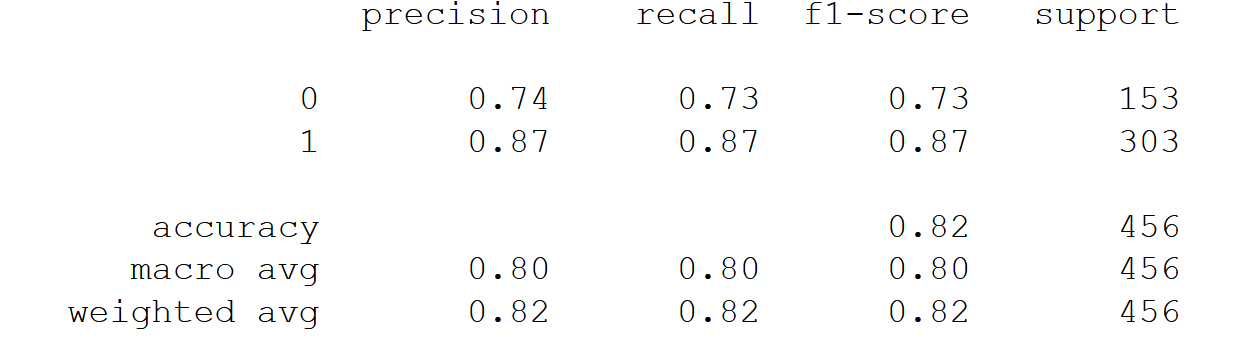


* Below is the confusion matrix and AUC-ROC curve for Naïve Bayes model:

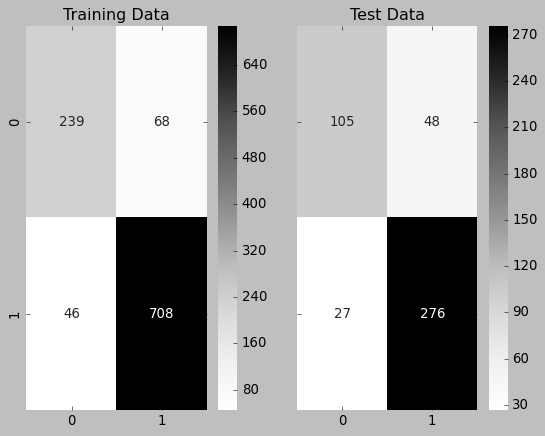
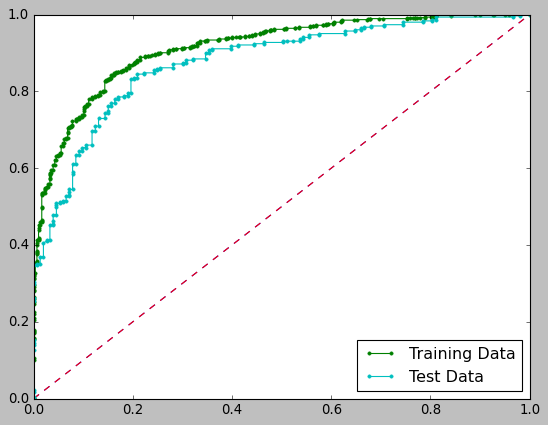
 

* Following is the classification matrix for train and test sets of the Naïve Bayes model

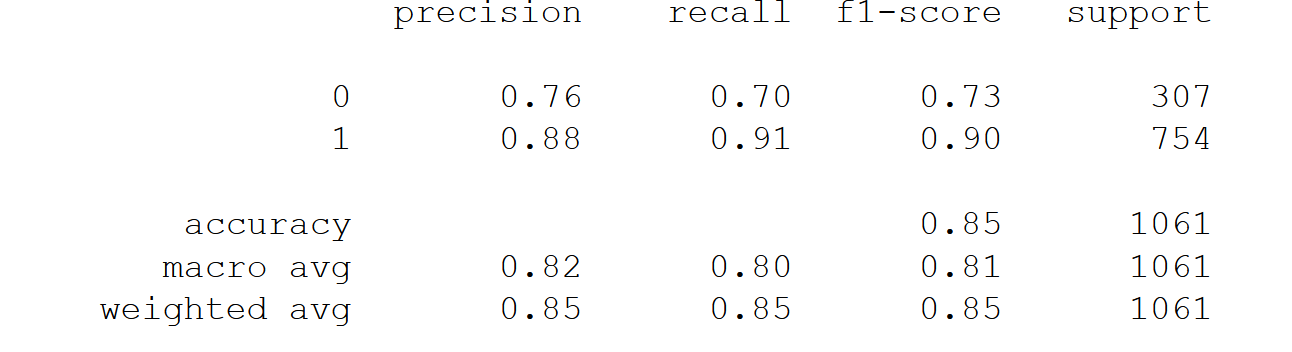


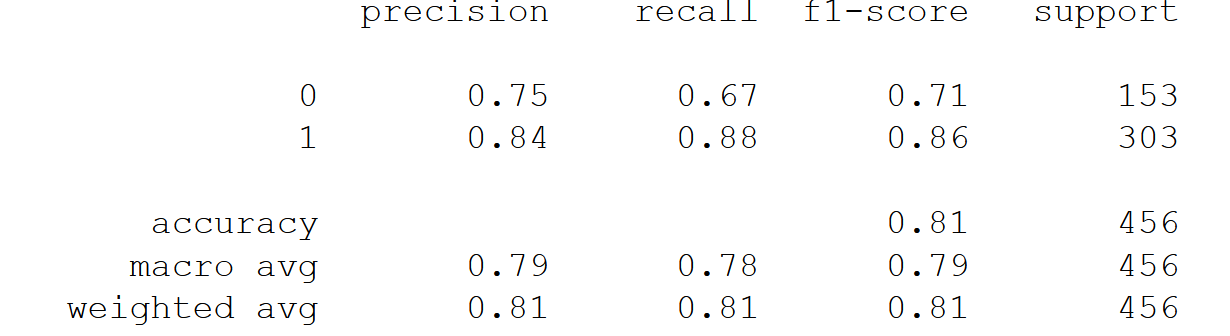


* Following are the confusion matrix and AUC ROC for Adaboost

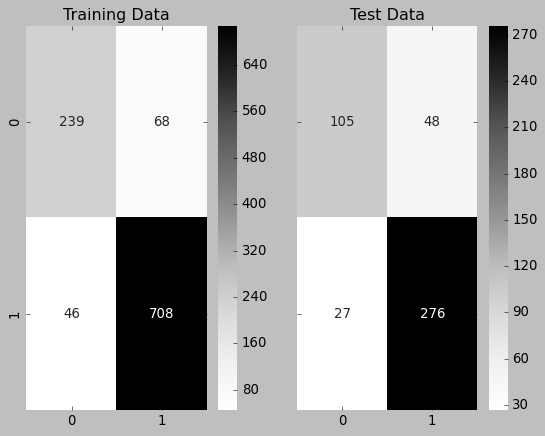
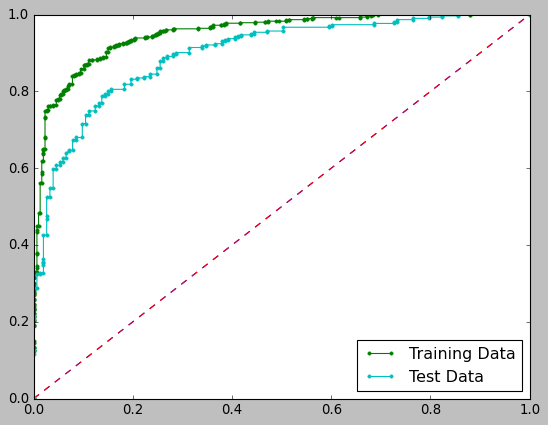
 

* Following is the train and test sets classification matrix for adaboost



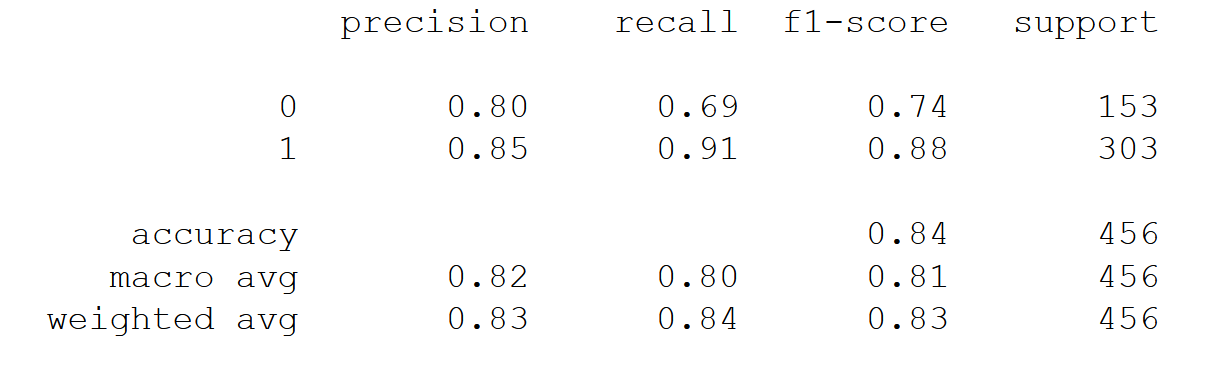


* Following are the confusion matrix and AUC ROC for Gradient boosting

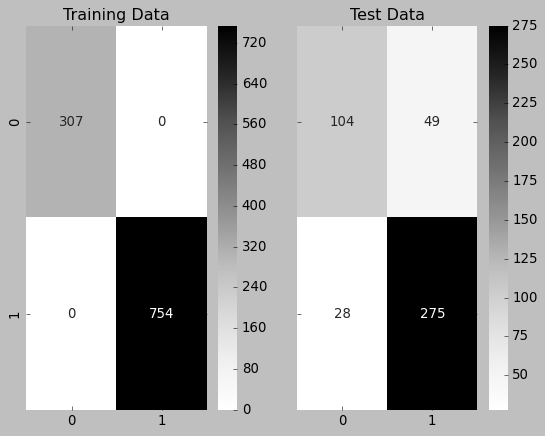
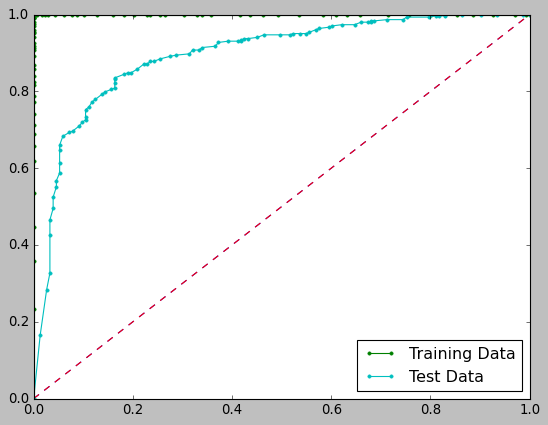
 

* Following is the train and test sets classification matrix for Gradient boosting

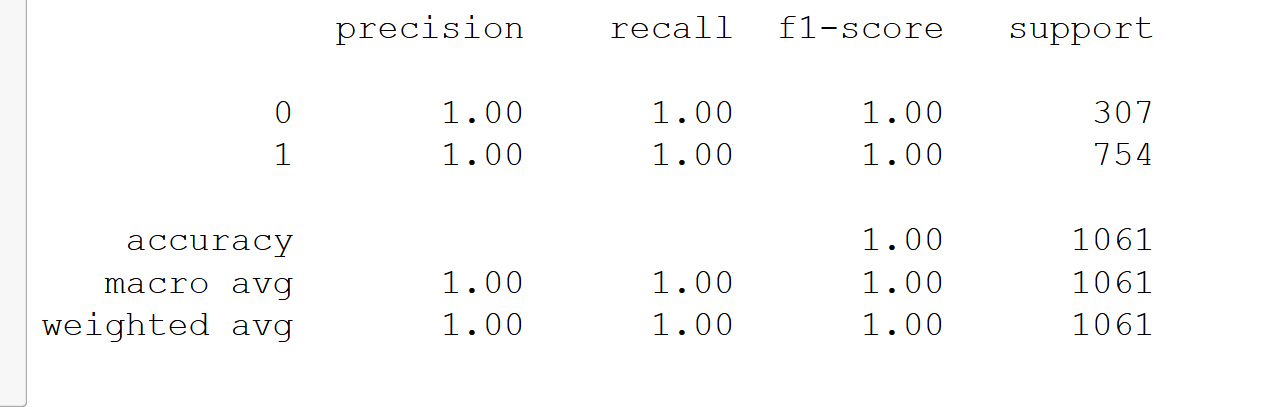


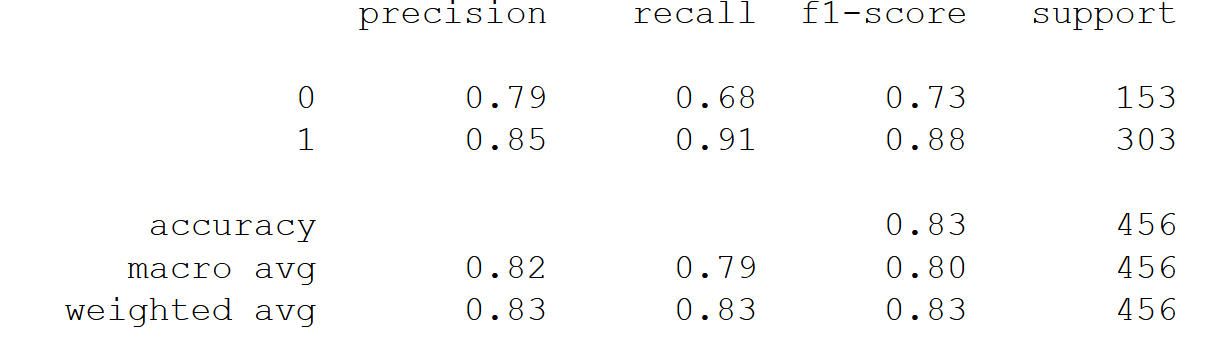


* Following are the confusion matrix and AUC ROC for Random forest

* Following is the train and test sets classification matrix for Random forest





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Model score \_ test** | **Accuracy\_test** | **Recall\_test(1)** | **AUC \_ test** |
| **Logistic regression** | **0.828** | **0.83** | **0.88** | **0.883** |
| **LDA** | **0.831** | **0.83** | **0.88** | **0.888** |
| **KNN** | **0.826** | **0.83** | **0.89** | **0.878** |
| **Naïve Bayes** | **0.822** | **0.82** | **0.87** | **0.876** |
| **Random forest** | **0.831** | **0.83** | **0.91** | **0.896** |
| **Adaboost** | **0.813** | **0.81** | **0.88** | **0.877** |
| **Gradient boosting** | **0.835** | **0.84** | **0.91** | **0.899** |

**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.**

* Based on the analysis we can infer that Gradient boosting model is best optimized.
* Based on the high recall score of vote\_labor, we can recommend the news channel to use gradient boosting model for prediction.
* The accuracy of true positives and negatives is also high for gradient boosting model
* However, for all other models than gradient boosting, the 16% false cases may hold a possibility of being true as the accuracy score is low.
* Gradient boosting is best optimized for election data to predict which party a voter has voted for on the basis of the given information and labor party is going to win in this scenario based on the true positive and negative numbers.

**Problem Statement 2*:***

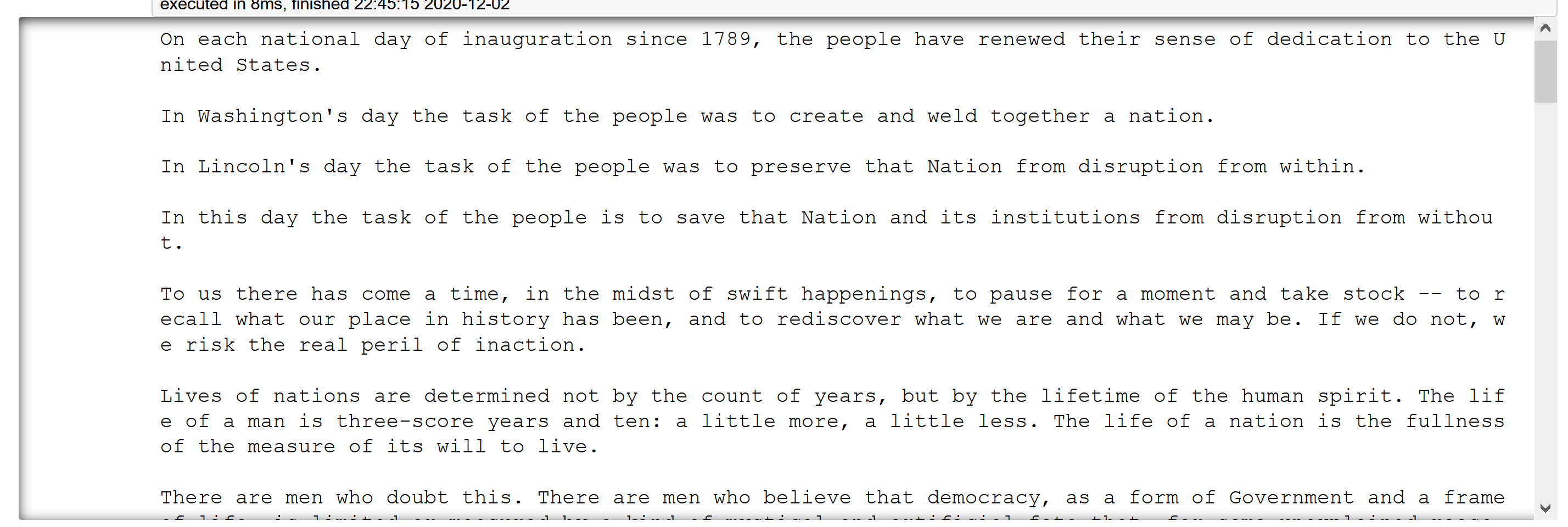
**In this particular project, we are going to work on the inaugural corpora from the nltk in Python. We will be looking at the following speeches of the Presidents of the United States of America:**

**President Franklin D. Roosevelt in 1941**

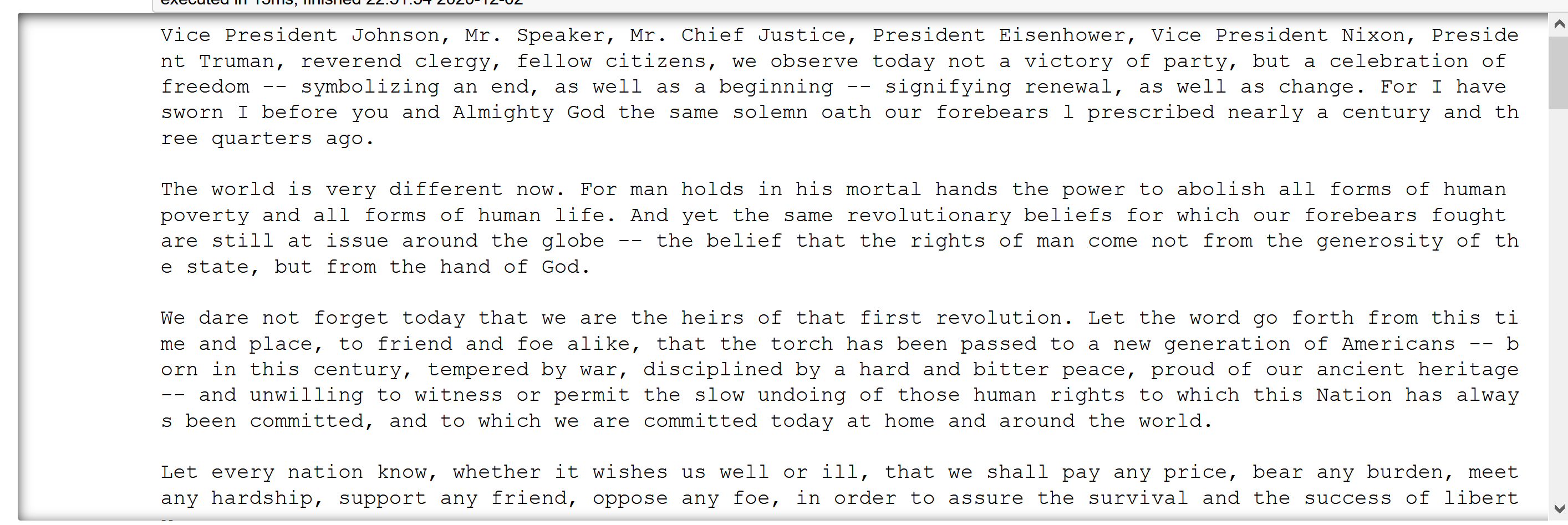
**President John F. Kennedy in 1961**

**President Richard Nixon in 19732.**

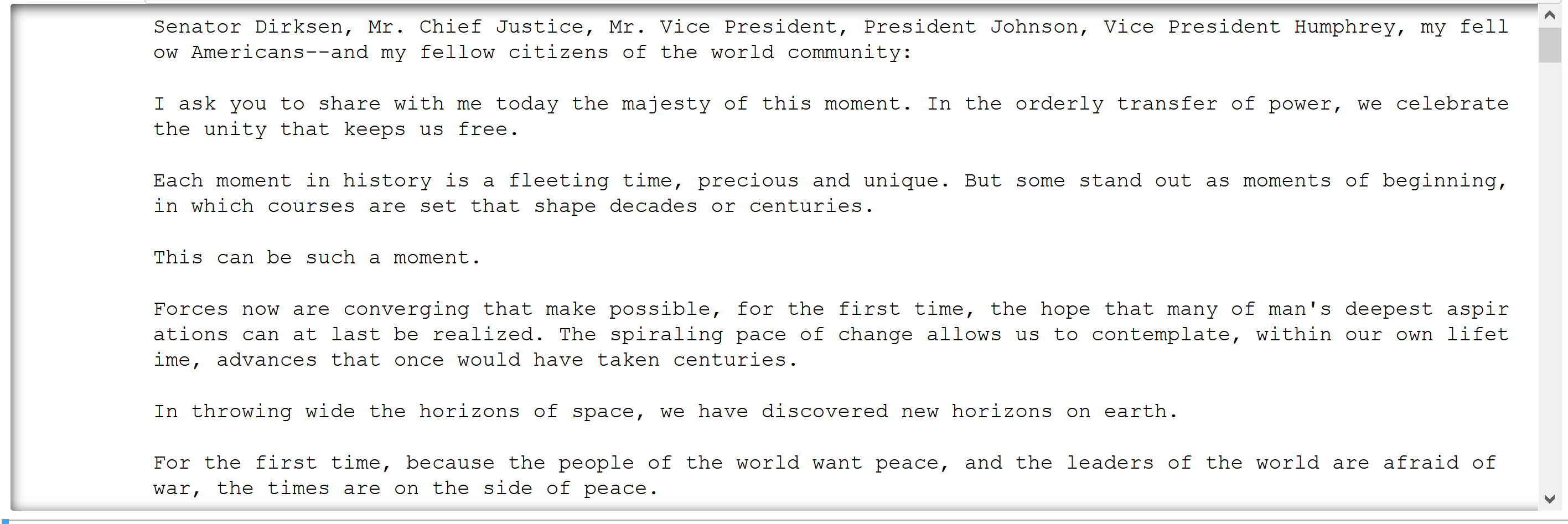
* **Find the number of characters, words and sentences for the mentioned documents.**
* First of all, ‘1941-Roosevelt.txt’ speech is read in python.



* ‘1941-Roosevelt.txt’ speech has **7571 characters, 1526 words, and 68 sentences**
* ‘1961-Kennedy.txt’ speech is read in python.

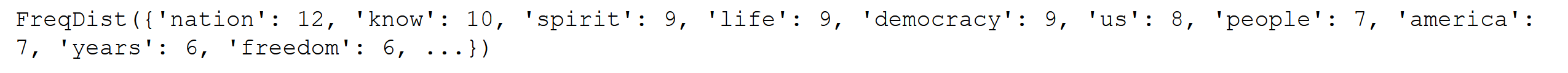


* ‘1961-Kennedy.txt’ speech has **7618 characters, 1546 words, and 52 sentences**
* ‘1969-Nixon.txt’ speech is read in python.



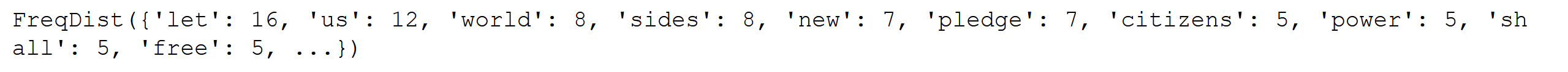
* ‘1969-Nixon.txt’ speech has **11624 characters, 2425 words, and 106 sentences**
* **Remove all the stopwords from all the three speeches..**
* After removing stop words from ‘1941-Roosevelt.txt’ speech, we get 630 clean words
* After removing stop words from ‘1961-Kennedy.txt’ speech, we get 697 clean words
* After removing stop words from ‘1969-Nixon.txt’ speech, we get 1013 clean words
* **Which word occurs the most number of times in his inaugural address for each president? Mention the top three words. (after removing the stopwords)**
* Following is the frequency distribution of the words in ‘1941-Roosevelt.txt’ speech,

Most common 3 words are ‘***nation’, ‘know’ and ‘spirit’***



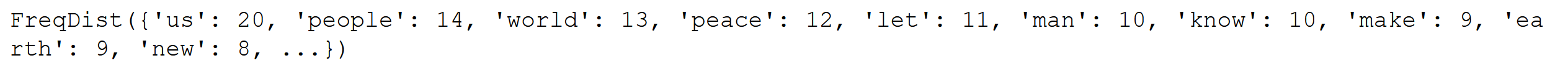
* Following is the frequency distribution of words in ‘1961-Kennedy.txt’ speech,

Most common 3 words are ‘***let’, ‘us’ and ‘world’***

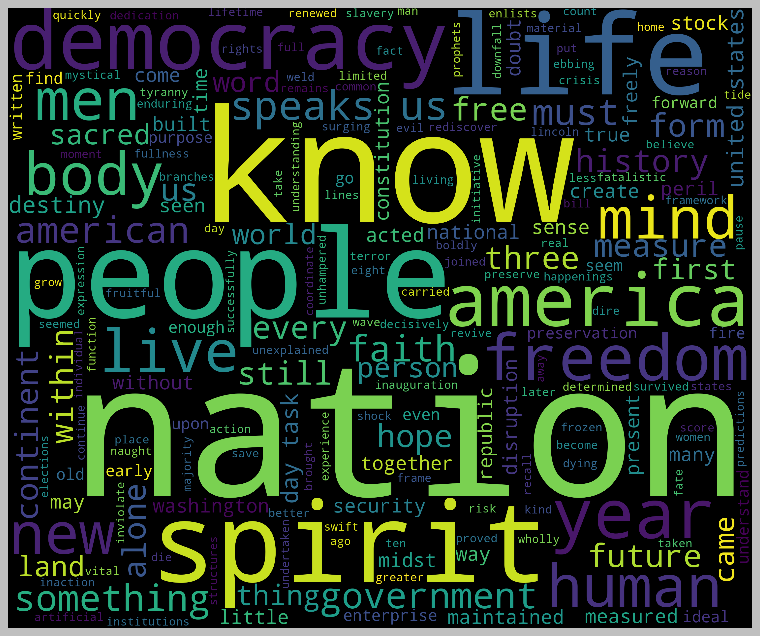


* Following is the frequency distribution of words in ‘1969-Nixon.txt’ speech,

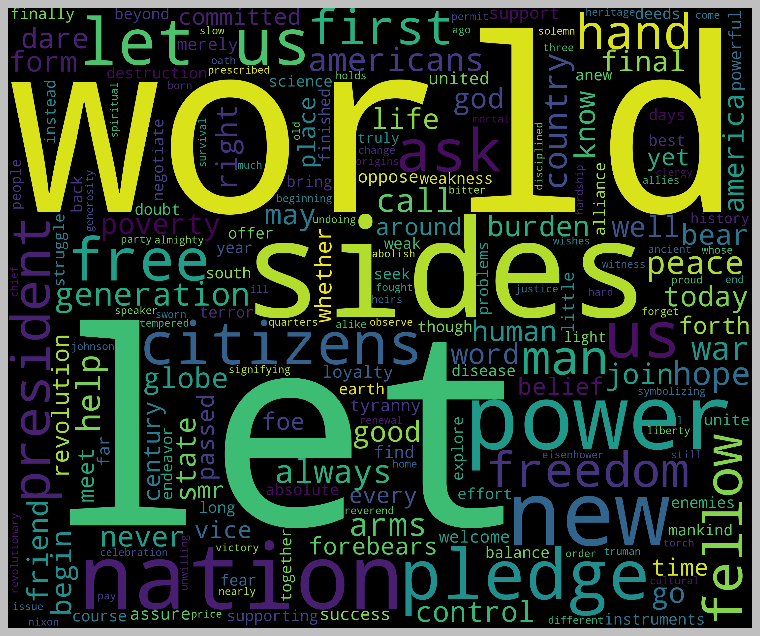
Most common 3 words are ‘***people, ‘us’ and ‘world’***



* **Plot the word cloud of each of the speeches of the variable. (after removing the stopwords)**
* Wordcloud of ‘1941-Roosevelt.txt’ speech



* Wordcloud of ‘1961-Kennedy.txt’ speech



* Wordcloud of ‘1941-Roosevelt.txt’ speech

