Machine Learning Project

PGP – DSBA

May ‘22

Date: 25/11/2022

Swaathi Ramakrishnan

**Table of Contents**

**Contents**

|  |  |
| --- | --- |
| Problem 1 | 5 |
| 1.1 Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it | 5 |
| 1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers | 7 |
| 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) | 12 |
| 1.4 Apply Logistic Regression and LDA (linear discriminant analysis) | 13 |
| 1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results | 16 |
| 1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting | 20 |
| 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 | 25 |
| 1.8 Inference: Based on these predictions, what are the insights? (5 marks) | 32 |
|  |  |
| Problem 2 | 33 |
| 2.1 Find the number of characters, words, and sentences for the mentioned documents | 34 |
| 2.2 Remove all the stopwords from all three speeches | 35 |
| 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) | 36 |
| 2.4 Plot the word cloud of each of the speeches of the variable. (after removing the stopwords) | 37 |

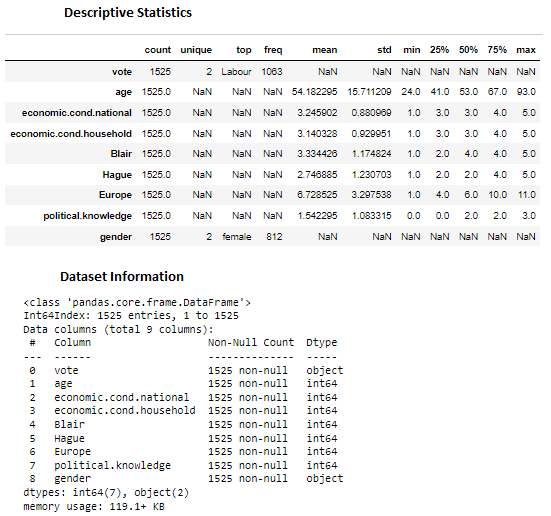
**List of Figures**

|  |  |
| --- | --- |
| Fig.1 Descriptive Statistics and Information of the Election Dataset | 5 |
| Fig 2. Duplicated rows, datatypes and null value check | 6 |
| Fig 3. Skewness of the data | 6 |
| Fig 4. Histograms and Box Plots for Election data | 8 |
| Fig 5. Catplot of the categorical variables | 8 |
| Fig 6. Pair Plot of election data | 9 |
| Fig 7. Correlation heatmap of election data | 10 |
| Fig 8. Before and After scaling | 12 |
| Fig 9. Data After scaling | 12 |
| Fig 10. Classification Report of Logistic Regression Train Data | 13 |
| Fig 11. Confusion Matrix of Logistic Regression Train Data | 13 |
| Fig 12. Classification Report of Logistic Regression Test Data | 13 |
| Fig 13. Confusion Matrix of Logistic Regression Test Data | 13 |
| Fig 14. Classification report and Confusion Matrix of LDA Train Data | 14 |
| Fig 15. Classification report and Confusion Matrix of LDA Test Data | 14 |
| Fig 16. Metrics of Logistic Regression and Linear Discriminant Analysis | 15 |
| Fig 17. Classification report and Confusion Matrix of KNN Train Data | 16 |
| Fig 18. Classification report and Confusion Matrix of KNN Test Data | 16 |
| Fig 19. Misclassification error to determine value of K | 17 |
| Fig 20. Classification report and Confusion Matrix of NB Train Data | 18 |
| Fig 21. Classification report and Confusion Matrix of NB Test Data | 18 |
| Fig 22. Metrics of KNN and Naive Bayes Algorithm | 18 |
| Fig 23. Logistic Regression Tuning using Grid Search | 20 |
| Fig 24. LDA Tuning using Grid Search | 20 |
| Fig 25. KNN Tuning using Grid Search | 21 |
| Fig 26. Comparing Accuracies before and after tuning | 21 |
| Fig 27. Bagging and Random Forest Classification | 22 |
| Fig 28. Bagging after model tuning | 22 |
| Fig 29. Number of estimators for Ada Boosting | 23 |
| Fig 30. Grid Search for Ada Boosting | 23 |
| Fig 31. Number of estimators for Gradient Boosting | 24 |
| Fig 32. Grid Search for Gradient Boosting | 24 |
| Fig 33. Bagging AUC-ROC Curve | 25 |
| Fig 34. Bagging AUC-ROC Curve after tuning | 25 |
| Fig 35. Ada Boosting AUC-ROC Curve | 25 |
| Fig 36. Ada Boosting AUC-ROC Curve after tuning | 25 |
| Fig 37. Gradient Boosting AUC-ROC Curve | 25 |
| Fig 38. Gradient Boosting AUC-ROC Curve after tuning | 25 |
| Fig 39. Classification report and Confusion Matrix of Bagging Train Data before tuning | 26 |
| Fig 40. Classification report and Confusion Matrix of Bagging Train Data after tuning | 26 |
| Fig 41. Classification report and Confusion Matrix of Bagging Test Data before tuning | 27 |
| Fig 42. Classification report and Confusion Matrix of Bagging Test Data after tuning | 27 |
| Fig 43. Classification report and Confusion Matrix of Ada Boosting Train Data before tuning | 28 |
| Fig 44. Classification report and Confusion Matrix of Ada Boosting Test Data after tuning | 28 |
| Fig 45. Classification report and Confusion Matrix of Ada Boosting Train Data before tuning | 28 |
| Fig 46. Classification report and Confusion Matrix of Ada Boosting Test Data after tuning | 29 |
| Fig 47. Classification report and Confusion Matrix of Gradient Boosting Train Data before tuning | 29 |
| Fig 48. Classification report and Confusion Matrix of Gradient Boosting Trian Data after tuning | 29 |
| Fig 49. Classification report and Confusion Matrix of Gradient Boosting Test Data before tuning | 30 |
| Fig 50. Classification report and Confusion Matrix of Gradient Boosting Test Data after tuning | 30 |
| Fig 51. Performance Metrics of all the models | 30 |
| Fig 52. WordCloud for Roosevelt’s speech | 37 |
| Fig 53. WordCloud for Kennedy’s speech | 38 |
| Fig 54. WordCloud for Nixon’s speech | 39 |

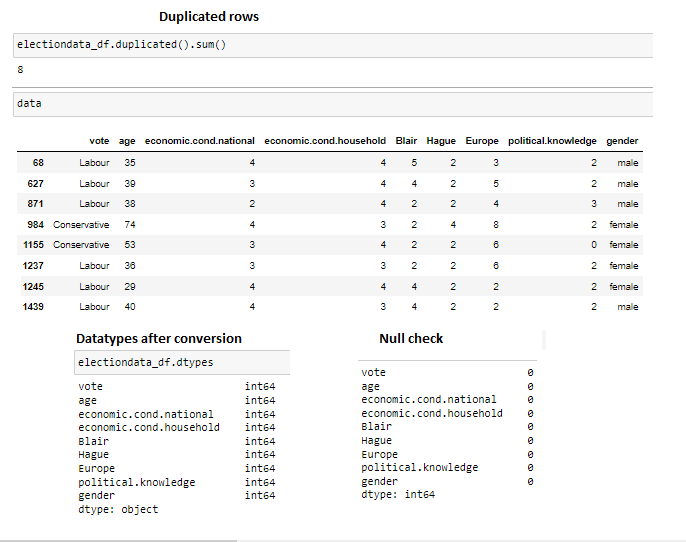
**Problem 1:**

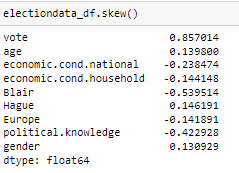
You are hired by one of the leading news channels CNBE who wants to analyze recent elections. This survey was conducted on 1525 voters with 9 variables. You have to build a model, to predict which party a voter will vote for on the basis of the given information, to create an exit poll that will help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

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



*Fig.1 Descriptive Statistics and Information of the Election Dataset*

*Fig 2. Duplicated rows, datatypes and null value check*



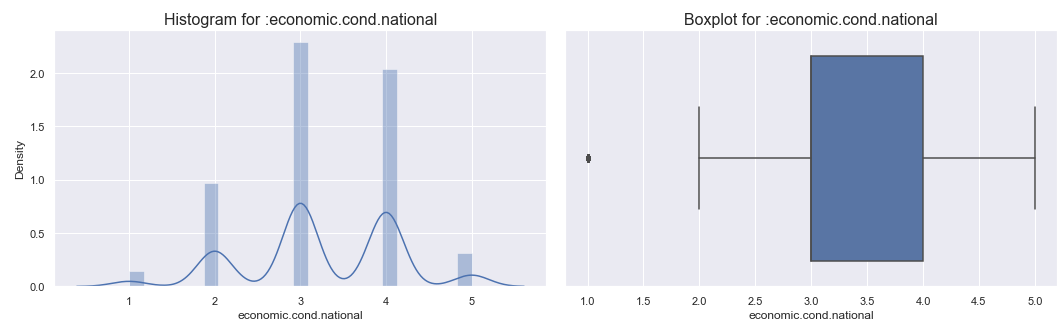
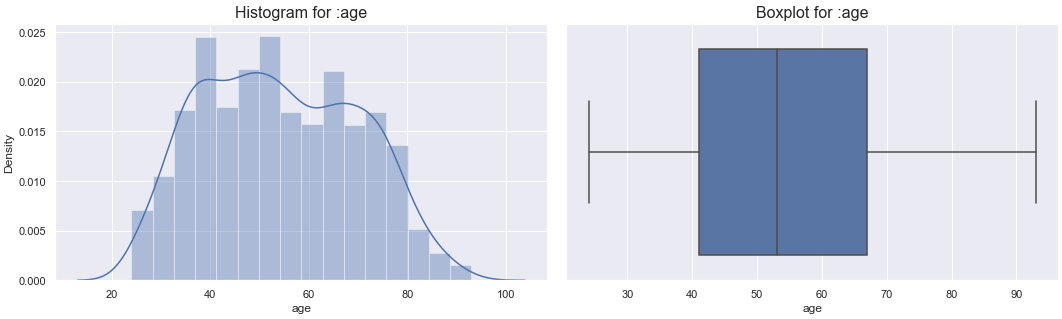
*Fig 3. Skewness of the data*

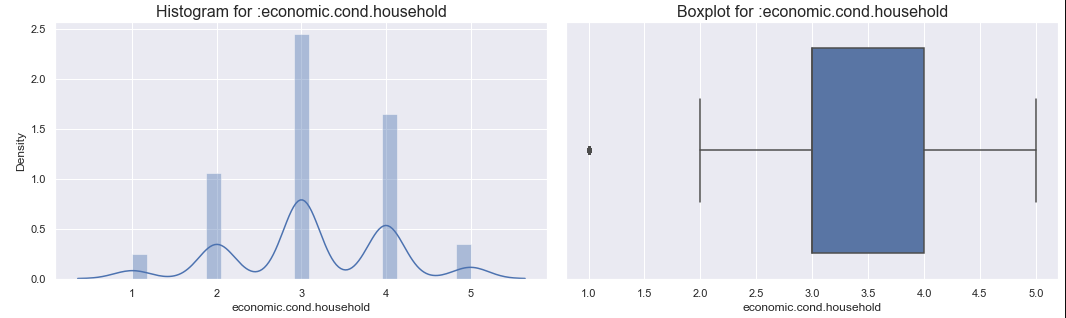
**Inference:**

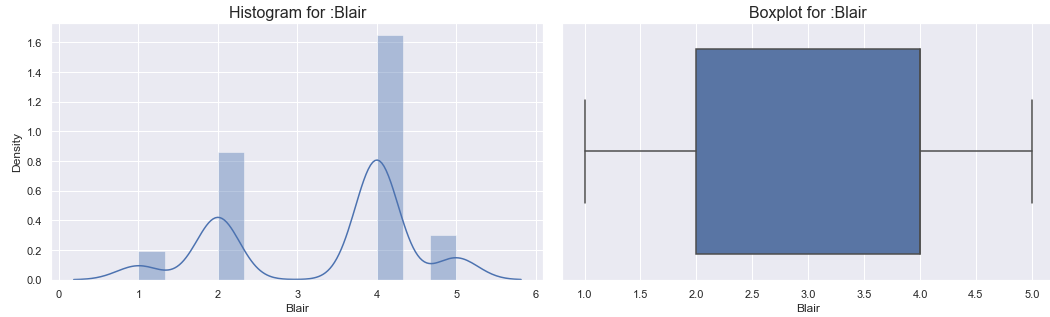
* There are 1525 rows of data, out of which 8 were duplicated rows, hence dropped them
* There are no null values in the dataset
* Gender is a categorical column since it has only 2 distinct values ‘Female’ and ‘Male’, hence replaced them to have 0 and 1 respectively. There are more females than males in the dataset
* Vote is a categorical column since it has only 2 distinct values ‘Conservative’ and ‘Labour’, hence replaced them to have 1 and 0 respectively.
* The figure Skewness of the data indicates that the data is moderately skewed

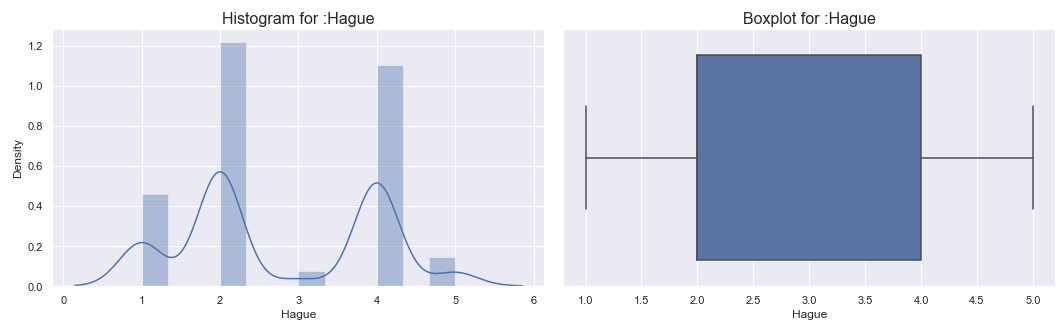
**1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers**

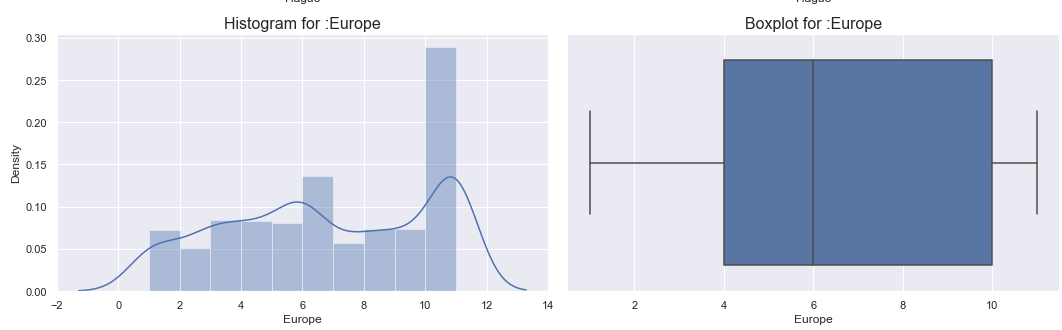
**Univariate Analysis:** Analysing one variable at a time

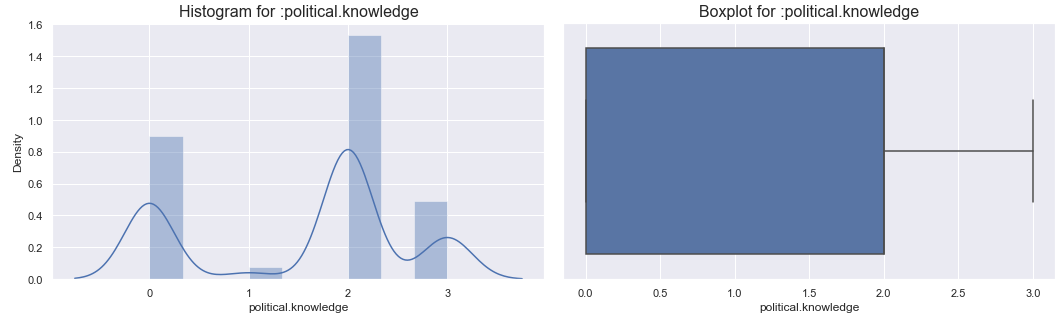


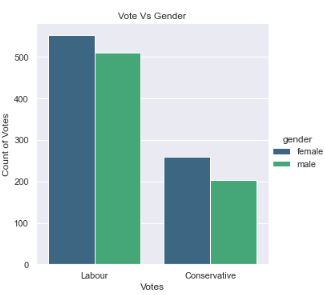






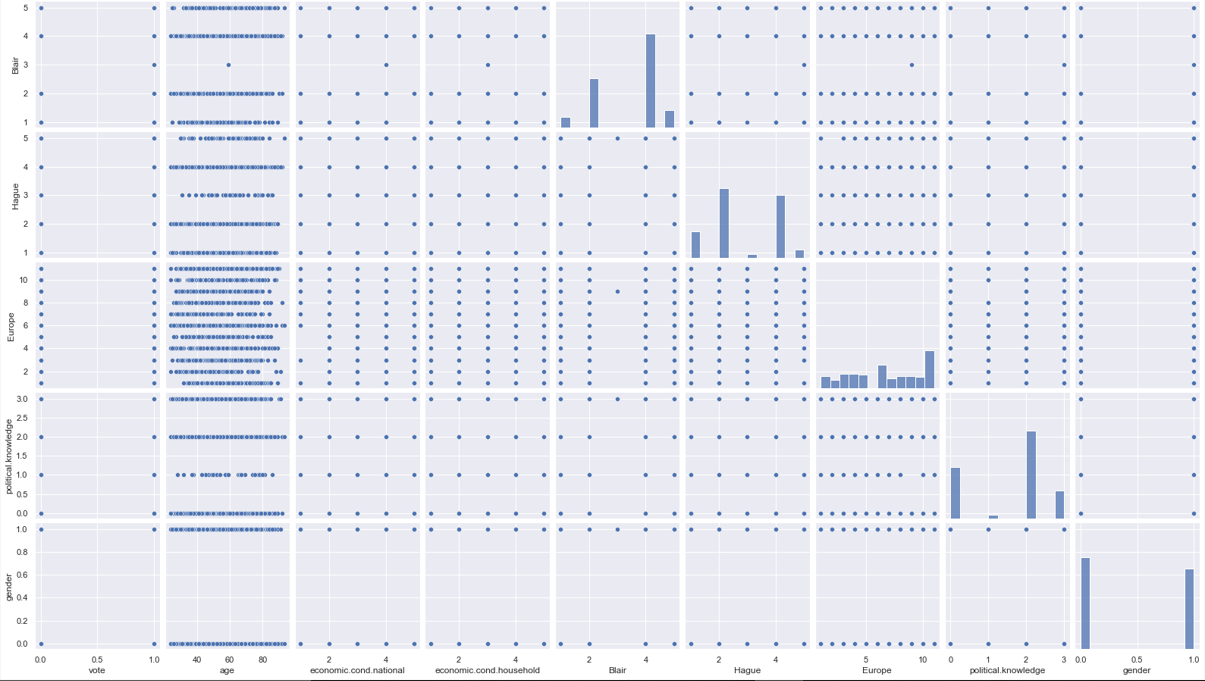


 *Fig 4. Histograms and Box Plots for Election data*



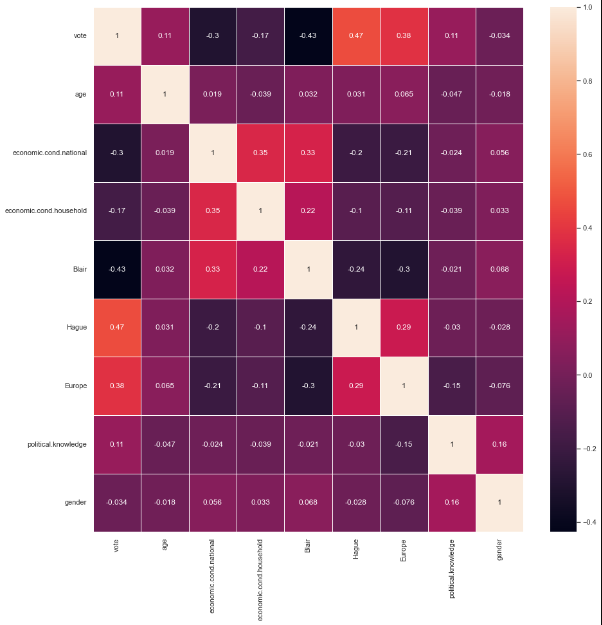
*Fig 5. Catplot of the categorical variables*

**Bivariate analysis:** Analysing 2 variables at a time



*Fig 6. Pair Plot of election data*

**Multivariate analysis:** Analysing more than two variables/all variables at a time



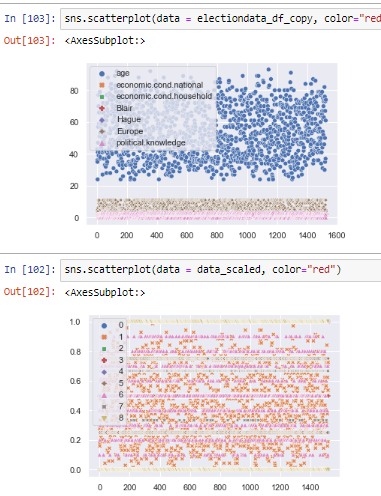
*Fig 7. Correlation heatmap of election data*

**Inference:**

* Age data is normally distributed
* There are no outliers. Minimum age is 23 and maximum age is 93
* Most of them are aged between 40 and 70
* Economic cond national and Economic cond household have one outlier each. These data are not normally distributed. The top values are 3 and 4
* Blair and Hague have no outliers. These data are not normally distributed. The top values are 2 and 4. Minimum is 1 and maximum is 5
* Europe has no outlier. It is not normally distributed. Minimum is 2 and maximum is 11
* Political Knowledge has no outlier and is not normally distributed. The mimimum is 0 and maximum is 2.
* The scatterplots from the pairplot show that there is no or very less correlation between the variables.
* Blair, Europe and political knowledge seem to be left skewed
* From fig 5 it is evident that female votes are higher than male votes in both labour and conservative parties. Labour party has more votes than the conservative party overall.
* From the correlation heatmap it is evident that there is very less correlation between the variables.
* Some are moderately positively correlated like:
  + Economic condition national with economic condition household and Blair
  + Vote with Hague and Europe
  + Europe with Hague
* Some are moderately negatively correlated like:
  + Vote with Economic Condition National and Blair
  + Economic condition national with Hague and Europe
  + Blair with Hague and Europe
  + Europe with political knowledge

**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)**

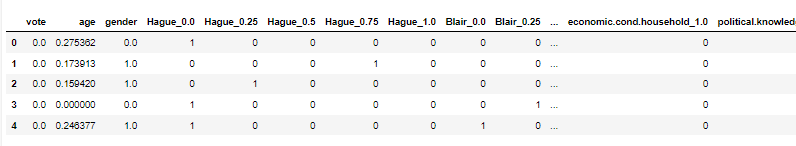
Feature Scaling is done to standardize or normalize all the features in a dataset into a fixed scale. The features in this dataset have varying magnitudes, hence to bring them to the same level of magnitude we are going to perform min-max-scaling technique, wherein the minimum value is 0 and the maximum value is 1.



Before scaling

After scaling

*Fig 8. Before and After scaling*

*Fig 9. Data After scaling*

Splitting the data into train and test 70:30

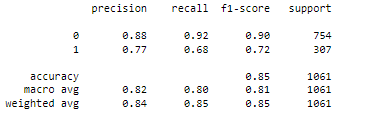
x\_train,x\_test , y\_train, y\_test = train\_test\_split(x,y,test\_size = 0.30 ,random\_state = 1)

x\_train (1061, 37) x\_test (456, 37)

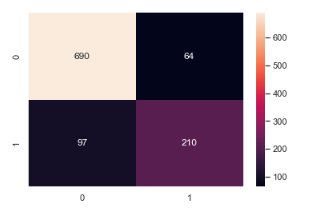
y\_train (1061,) y\_test (456,)

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

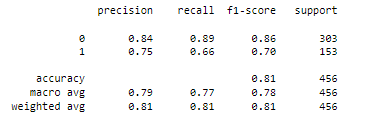
**Logistic regression:** It is used to predict the categorical dependant variable using the other independent variables. Vote is our target variable.



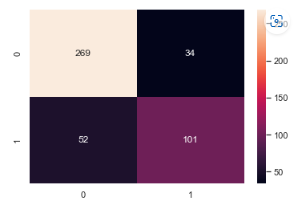
*Fig 10. Classification report of Logistic Regression Train Data*

**

*Fig 11. Confusion Matrix of Logistic Regression Train Data*

**

*Fig 12. Classification report of Logistic Regression Test Data*



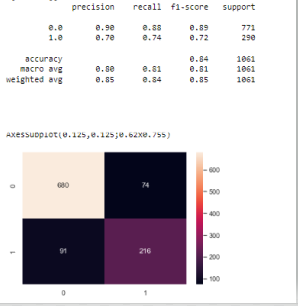
*Fig 13. Confusion Matrix of Logistic Regression Test Data*

**Inference:**

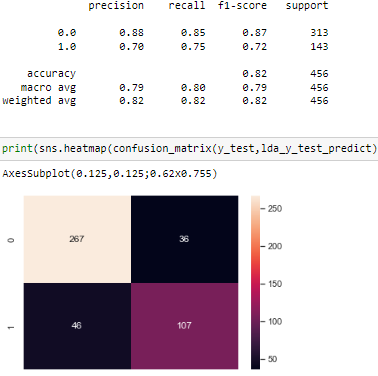
We can see a slightly higher error in the test data than train data, in both the models. But we can understand that there is no over or under fitting of the model as the error in both train and test data is very low. Hence the model is valid.

**Linear Discriminant Analysis**

It is a predictive modelling algorithm used for classification.

****

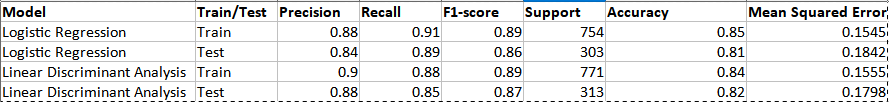
*Fig 14. Classification report and Confusion Matrix of LDA Train Data*

**

*Fig 15. Classification report and Confusion Matrix of LDA Test Data*

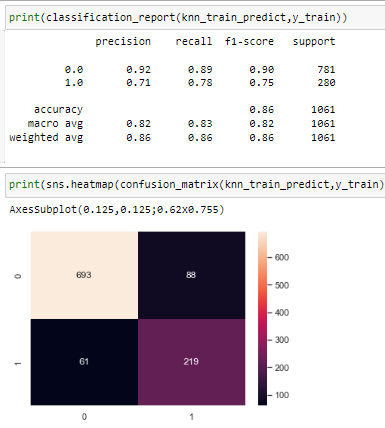
**Inference:**

We can see a slightly higher error in the test data than train data, in both the models. But we can understand that there is no over or under fitting of the model as the error in both train and test data is very low. Hence the model is valid. Recall scores are good in both the models.

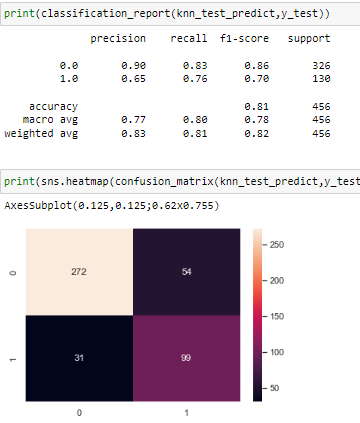
*Fig 16. Metrics of Logistic Regression and Linear Discriminant Analysis*

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

**K-Nearest Neighbour:** It is a simple Machine learning algorithm based on supervised learning, wherein it classifies a new data point based on the similarity. The k-value can be determined using the misclassification error graph.

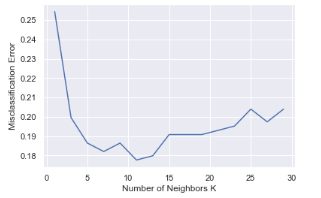


*Fig 17. Classification report and Confusion Matrix of KNN Train Data*



*Fig 18. Classification report and Confusion Matrix of KNN Test Data*

Misclassification error to determine value of K: K = 11 and 12 have minimum values for the misclassification error from the graph below. Hence choosing the n-neighbours to be 11.

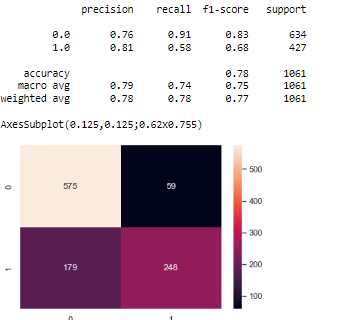


*Fig 19. Misclassification error to determine value of K*

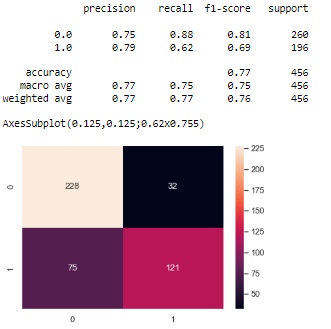
**Inference:**

We can conclude that there is no over or under fitting of the model as the difference in accuracy between both train and test data is less than 10%. Hence the model is valid.

**Naive Bayes:** It is a classification technique based on Bayes theorem. It works fast and performs better with less training data.



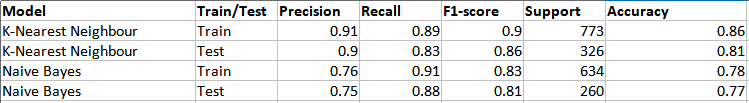
*Fig 20. Classification report and Confusion Matrix of NB Train Data*



*Fig 21. Classification report and Confusion Matrix of NB Test Data*

**Inference:**

We can conclude that there is no over or under fitting of the model as the difference in accuracy between train and test data is less than 10%. Hence the model is valid.

*Fig 22. Metrics of KNN and Naive Bayes Algorithm*

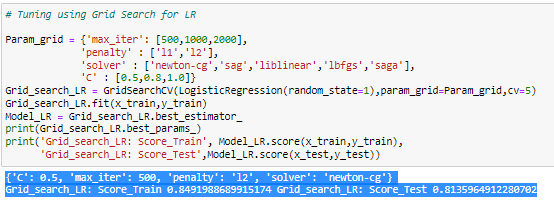
**Interpretation of above models – LR, LDA, KNN and NB:**

* None of the models above are over-under fitted, since the difference between the accuracy scores of both train and test set is not more than 10%.
* Both Recall and Precision are deciding factors in deciding which model needs to be considered.
* Both KNN and LDA have almost equal recall and precision scores, hence these can be considered for further analysis.
* Accuracy of all models except Naive Bayes is good.
* By tuning the hyper parameters of the above models we can select the final better model after comparing the metrics. For this we will further be including ensemble techniques like Boosting and Bagging.

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

Naive Bayes model does not have any hyperparameters, hence using the model metrics as it is.

**Logistic Regression Tuning:**



*Fig 23. Logistic Regression Tuning using Grid Search*

**Inference:**

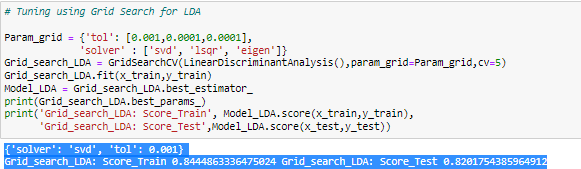
* The model here has used 500 iterations for tuning.
* newton-cg is chosen as solver.
* This solver supports only l2 as penalty.
* 0.8 value for c represents strong regularization
* Accuracy has not changed after tuning.

Best Params:

{'C': 0.5, 'max\_iter': 500, 'penalty': 'l2', 'solver': 'newton-cg'}

Grid\_search\_LR: Score\_Train 0.8491988689915174 Grid\_search\_LR: Score\_Test 0.8135964912280702

**LDA Tuning:**



*Fig 24. LDA Tuning using Grid Search*

**Inference:**

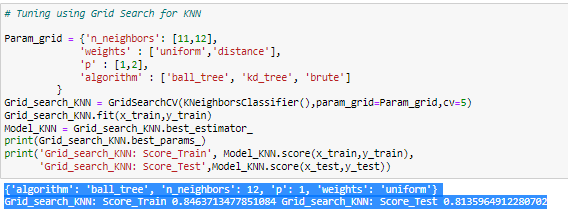
* Solver chosen is ‘sdv’ by the model.
* Accuracy has not changed much after tuning.

Best Params:

{'solver': 'svd', 'tol': 0.001}

Grid\_search\_LDA: Score\_Train 0.8444863336475024 Grid\_search\_LDA: Score\_Test 0.8201754385964912

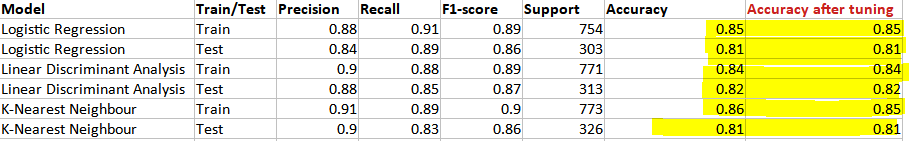
**KNN Tuning:**



*Fig 25. KNN Tuning using Grid Search*

**Inference:**

* N-Neighbours used as 11 and 12 using the graph for Misclassification error
* The model has chosen 12 as the best parameter
* Weights is chosen as ‘uniform’ by the model which means all the data points in the neighbourhood have equal weights
* Algorithm is chosen as ‘ball-tree’

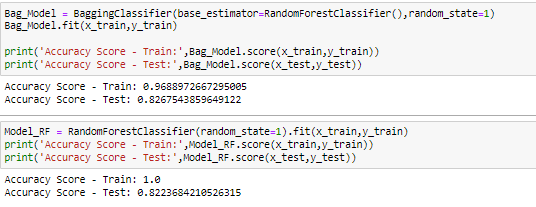


*Fig 26. Comparing Accuracies before and after tuning*

The above figure indicates that the accuracies have not changed much after tuning for all the 3 models.

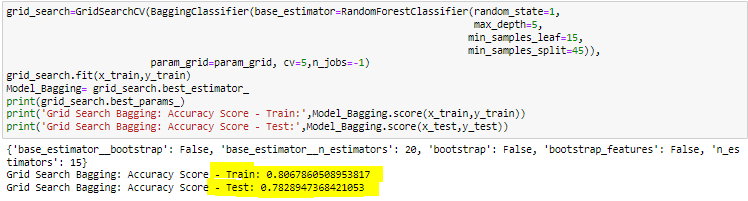
**Ensemble methods**

**Bagging**

****

*Fig 27. Bagging and Random Forest Classification*

Both models for Bagging and Random Forest show high accuracy. The difference between test and train accuracies are higher that 10%. Hence these are not valid models and are over-fitted. Hence we need to tune these models hyperparametrs using Grid Search.

*Fig 28. Bagging after model tuning*

Best Params:

{'base\_estimator\_\_bootstrap': False, 'base\_estimator\_\_n\_estimators': 20, 'bootstrap': False, 'bootstrap\_features': False, 'n\_estimators': 15}

Grid Search Bagging: Accuracy Score - Train: 0.8067860508953817

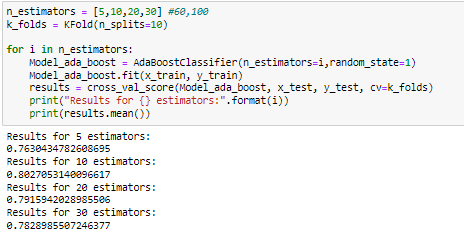
Grid Search Bagging: Accuracy Score - Test: 0.7828947368421053

**Inference:**

* Base estimator is RF
* From the best params it is evident that the difference in the accuracy scores between Test and Train is minimal.
* Hence Grid search has helped to tune the model hyperparameters and over-fitting of the model is corrected.

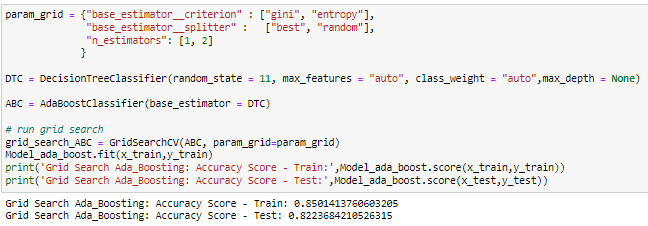
**Boosting**

**Ada Boosting:**

****

*Fig 29. Number of estimators for Ada Boosting*

From the above figure it is evident that the accuracy is good for 10 estimators.

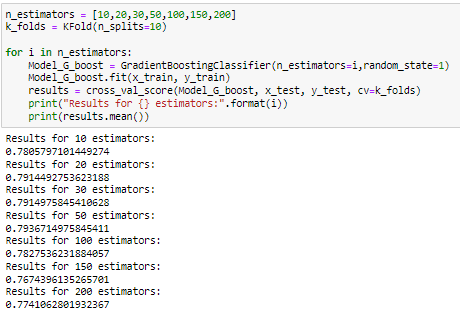


*Fig 30. Grid Search for Ada Boosting*

**Inference:**

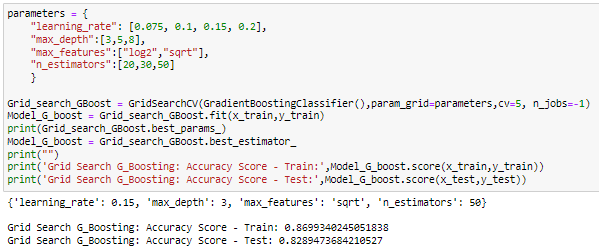
* Decision Tree classifier was used to fit the model
* The accuracy score from the above is 85% and the difference between train and test scores is minimal. Hence not over-fitted.

**Gradient Boosting:**



*Fig 31. Number of estimators for Gradient Boosting*

From the above figure it is evident that the accuracy is good for 50 estimators.

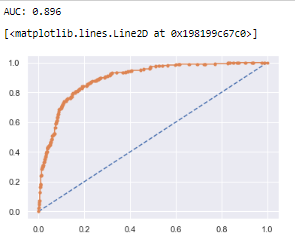
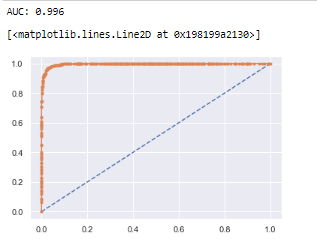


*Fig 32. Grid Search for Gradient Boosting*

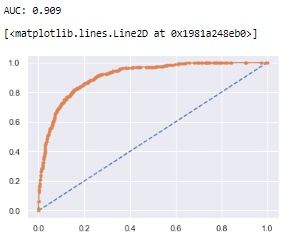
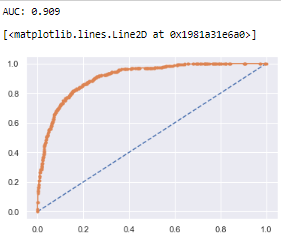
**Inference:**

* 3 is chosen as max\_depth, hence the depth of the tree is 3
* n\_estimators indicate the number of forest trees
* From the best params it is evident that the difference in the accuracy scores between Test and Train is minimal. Hence no over-fitting.

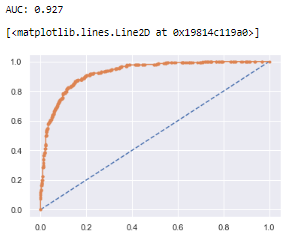
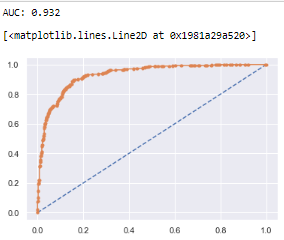
**1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized**



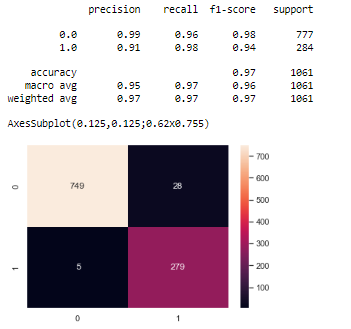
*Fig 33. Bagging AUC-ROC Curve Fig 34. Bagging AUC-ROC Curve after tuning*

** 

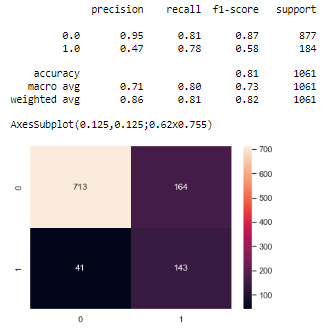
*Fig 35. Ada Boosting AUC-ROC Curve Fig 36. Ada Boosting AUC-ROC Curve after tuning*

**

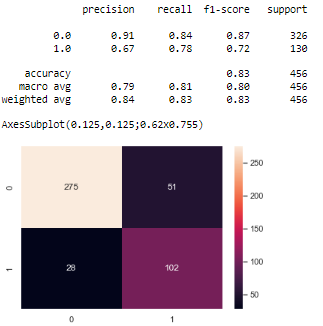
*Fig 37. Gradient Boosting AUC-ROC Curve Fig 38. Gradient Boosting AUC-ROC Curve after tuning*

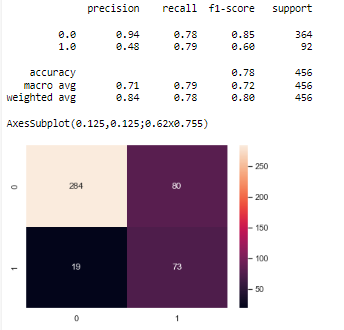
**

*Fig 39. Classification report and Confusion Matrix of Bagging Train Data before tuning*

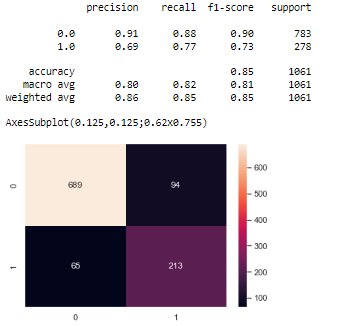
**

*Fig 40. Classification report and Confusion Matrix of Bagging Train Data after tuning*

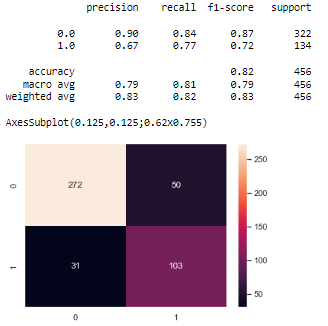
**

*Fig 41. Classification report and Confusion Matrix of Bagging Test Data before tuning*

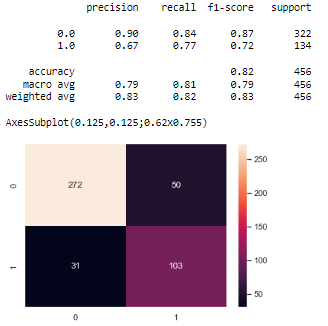
*Fig 42. Classification report and Confusion Matrix of Bagging Test Data after tuning*



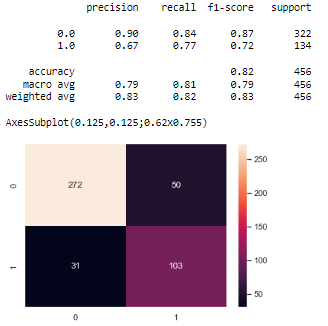
*Fig 43. Classification report and Confusion Matrix of Ada Boosting Train Data before tuning*

**

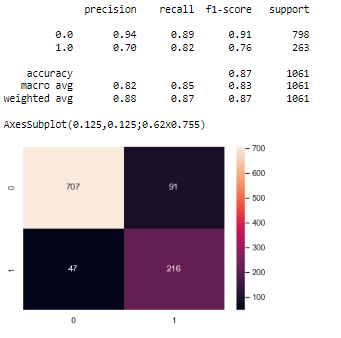
*Fig 44. Classification report and Confusion Matrix of Ada Boosting Train Data after tuning*



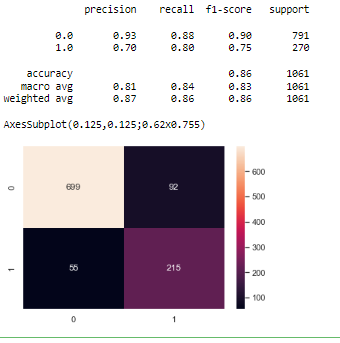
*Fig 45. Classification report and Confusion Matrix of Ada Boosting Test Data before tuning*

**

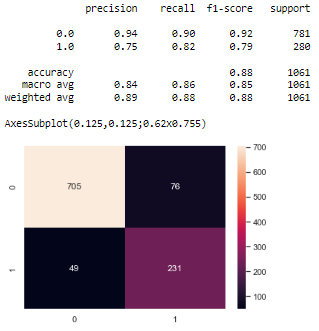
*Fig 46. Classification report and Confusion Matrix of Ada Boosting Test Data after tuning*

**

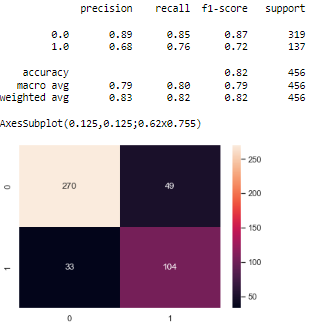
*Fig 47. Classification report and Confusion Matrix of Gradient Boosting Train Data before tuning*

**

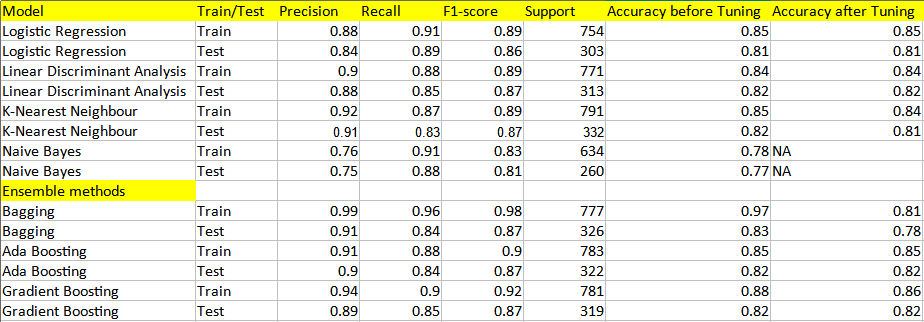
*Fig 48. Classification report and Confusion Matrix of Gradient Boosting Train Data after tuning*



*Fig 49. Classification report and Confusion Matrix of Gradient Boosting Test Data before tuning*

**

*Fig 50. Classification report and Confusion Matrix of Gradient Boosting Test Data after tuning*



*Fig 51. Performance Metrics of all the models*

**Inference:**

* None of the models are over-fitted
* After tuning also both train and test models showed a good accuracy > 80%
* Overall, the tuned models have performed better than the normal models

**Final model selection:**

* Gradient Boosting model has performed the best after tuning having the highest accuracy score of 86%
* It also has the best AUC score of 0.932 for train and 0.927 for test data.
* It has the highest Precision (0.93) and Recall (0.88) values when compared with other models
* This can be concluded as the most consistent model and best optimized.

**1.8 Inference: Based on these predictions, what are the insights? (5 marks)**

* Tuned Gradient Boost model is the best selected final model
* There was no over fitting in any of the models except Bagging
* Gender and economic.cond.household may have no or minimal impact on modelling. So they are considered as weak predictors
* About 30% of the total population have no idea about politics
* Blair has more votes than Hague and the scoring is also much better
* All models have performed well on training and test data, and the tuned models have performed better than normal models.
* Labour party has more votes than conservative party
* Median age of voters in conservative party is higher than than of labor party, hence old age people prefer conservative party
* Models with low accuracy scores and over fitted models have been tuned using Grid Search
* There are more number of female voters compared to male voters

**Business recommendations:**

* The dataset is considered a small dataset. More data could give us better modelling results.
* Hyperparams tuning is very important while building a model. This may take more process and time. But with more number of params, we may arrive at better results.

**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:

President Franklin D. Roosevelt in 1941

President John F. Kennedy in 1961

President Richard Nixon in 1973

(Hint: use .words(), .raw(), .sent() for extracting counts)

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

Number of characters in Roosevelt : 7571

Number of characters in Kennedy : 7618

Number of characters in Nixon : 9991

Number of words in Roosevelt : 1360

Number of words in Kennedy : 1390

Number of words in Nixon : 1819

Number of sentences in Roosevelt : 67

Number of sentences in Kennedy : 52

Number of sentences in Nixon : 68

**2.2 Remove all the stopwords from all three speeches**

Number of words in Roosevelt before removing stop-words: 1360

Number of words in Kennedy before removing stop-words: 1390

Number of words in Nixon before removing stop-words: 1819

Number of words in Roosevelt after removing stop-words: 871

Number of words in Kennedy after removing stop-words: 904

Number of words in Nixon after removing stop-words: 1094

Sample Sentence after removing stop-words from Roosevelt's speech:

people create weld together nation lincoln day task people preserve nation disruption within day

Sample Sentence after removing stop-words from Kennedy's speech:

truman reverend clergy fellow citizens observe today victory party celebration freedom symbolizing

Sample Sentence after removing stop-words from Nixon's speech:

country share together met four years ago america bleak spirit depressed prospect seemingly endless

**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)**

Most common and top 3 words in Roosevelt’s speech:

Nation- 12

Spirit- 9

Life- 9

Most common and top 3 words in Kennedy’s speech:

World- 8

Sides- 8

Pledge- 7

Most common and top 3 words in Nixon’s speech:

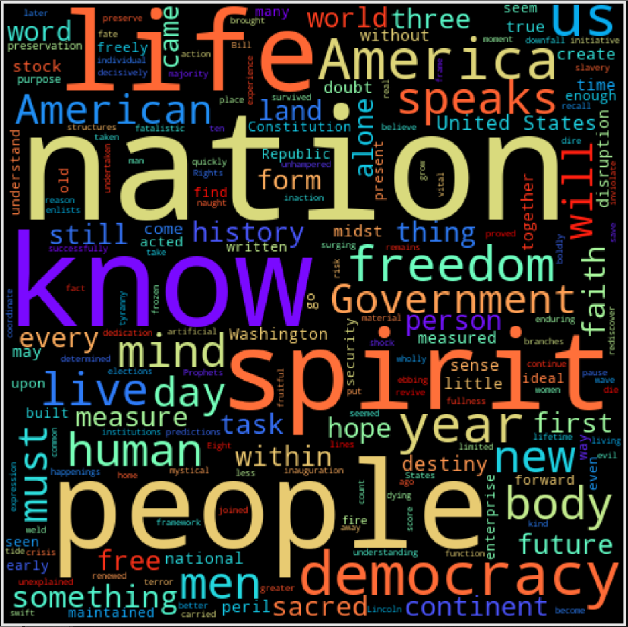
America- 21

Peace- 19

World- 18

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

**Speech 1 – Roosevelt**



*Fig 52. WordCloud for Roosevelt’s speech*

**Speech 2 – Kennedy**

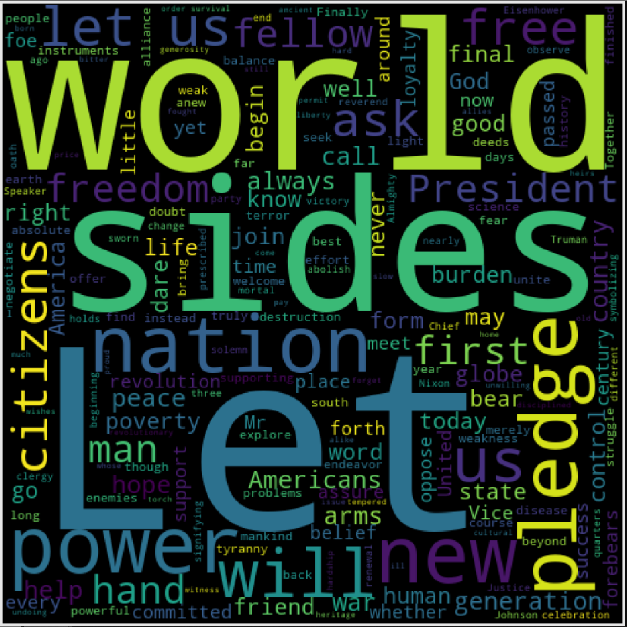


Fig 53. WordCloud for Kennedy’s speech

**Speech 3 – Nixon**



Fig 54. WordCloud for Nixon’s speech