Machine Learning

Pradeep Kumar Mishra
PGP-DSBA Online
Jun_B_21
Date: 09:Jan:2022

Content View

List of Figures :

List of Tables :

Problem Statement - 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.

Dataset for Problem: Election_Data.xlsx

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

Sample of the dataset :

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

Table-01

Types of variables and missing values in the dataset :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Unnamed: 0              1525 non-null   int64
 1   vote                    1525 non-null   object
 2   age                     1525 non-null   int64
 3   economic.cond.national  1525 non-null   int64
 4   economic.cond.household 1525 non-null   int64
 5   Blair                   1525 non-null   int64
 6   Hague                   1525 non-null   int64
 7   Europe                  1525 non-null   int64
 8   political.knowledge     1525 non-null   int64
 9   gender                  1525 non-null   object
dtypes: int64(8), object(2)
memory usage: 119.3+ KB
```

- From the above results we can see that there is no missing value present in the dataset.
- There are a total of 1525 rows and 9 columns in the dataset.
- Out of 9 variables there are 2 objects and the remaining are int64.

Check for duplicate data :

- There are 8 rows duplicated.
- I am going to remove them because they do not add any value.

Checking the Summary Statistic :

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
count	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000
mean	54.241266	3.245221	3.137772	3.335531	2.749506	6.740277	1.540541
std	15.701741	0.881792	0.931069	1.174772	1.232479	3.299043	1.084417
min	24.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
25%	41.000000	3.000000	3.000000	2.000000	2.000000	4.000000	0.000000
50%	53.000000	3.000000	3.000000	4.000000	2.000000	6.000000	2.000000
75%	67.000000	4.000000	4.000000	4.000000	4.000000	10.000000	2.000000
max	93.000000	5.000000	5.000000	5.000000	5.000000	11.000000	3.000000

Table - 02

	vote	gender
count	1517	1517
unique	2	2
top	Labour	female
freq	1057	808

Skewness :

```
age Skewed:  0.1398
economic.cond.national Skewed:  -0.2385
economic.cond.household Skewed:  -0.1441
Blair Skewed:  -0.5395
Hague Skewed:  0.1462
Europe Skewed:  -0.1419
political.knowledge Skewed:  -0.4229
```

- We see the age variable mean/media are nearly equal and positive skewed.
- Economic.cond.national, economic.cond.household, Blair, Europe and political.knowledge variables are negatively skewed .
- Hague is positively skewed.
- Gender variable is objects and most of the people are female.
- There are two parties one is Labour and other is Conservative, most of the people voted Labour party.

1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers. (7 Marks) Data Preparation: 4 marks.

Univariate Analysis :

Distplot :

By observing the below figure there is skewness in both sides (left side and right side) and data is not normally distributed.

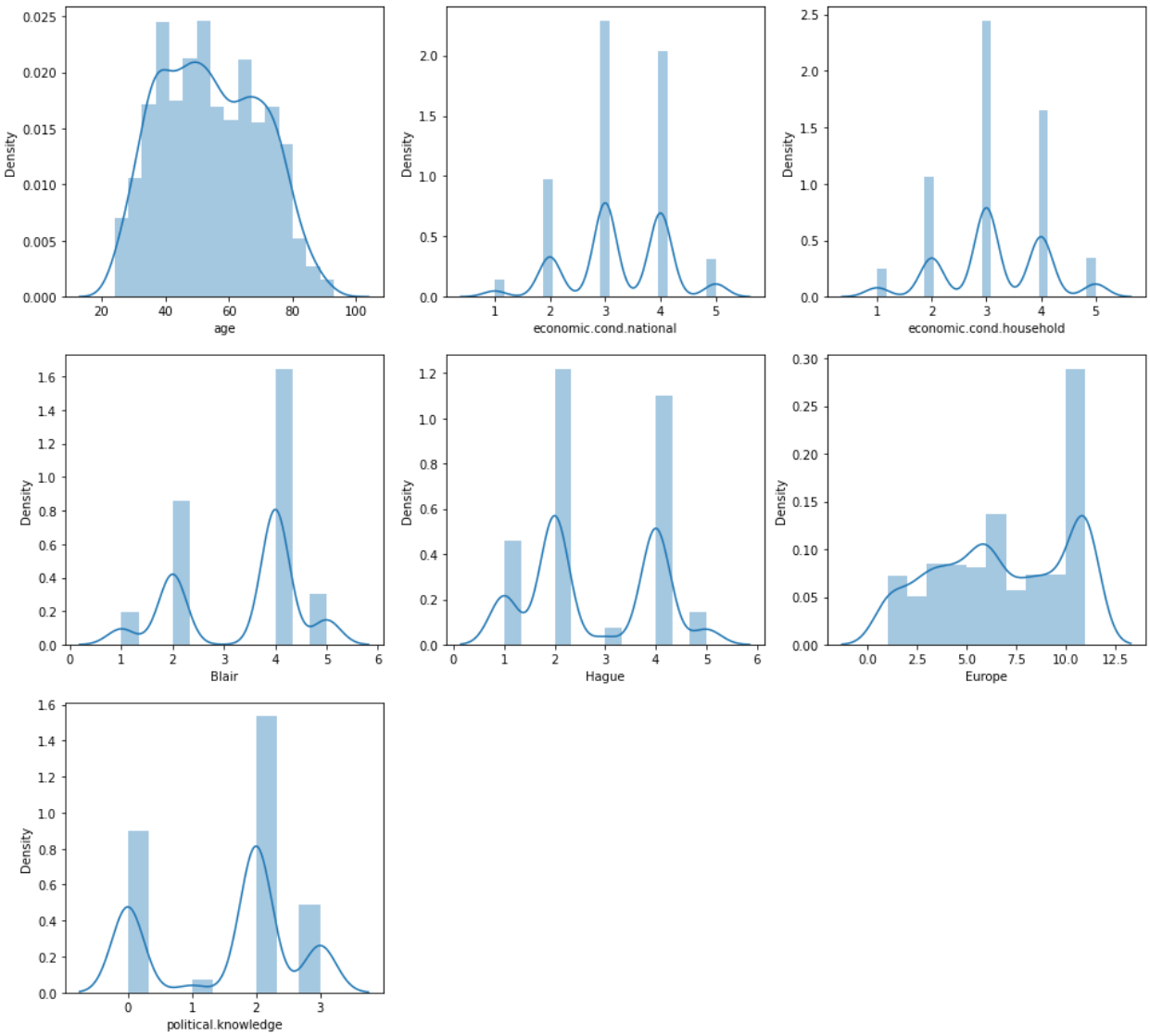

Figure-01

Boxplot :

By seeing the below boxplot there are some feature have outliers eg. economic.cond.national and economic.cond.household.

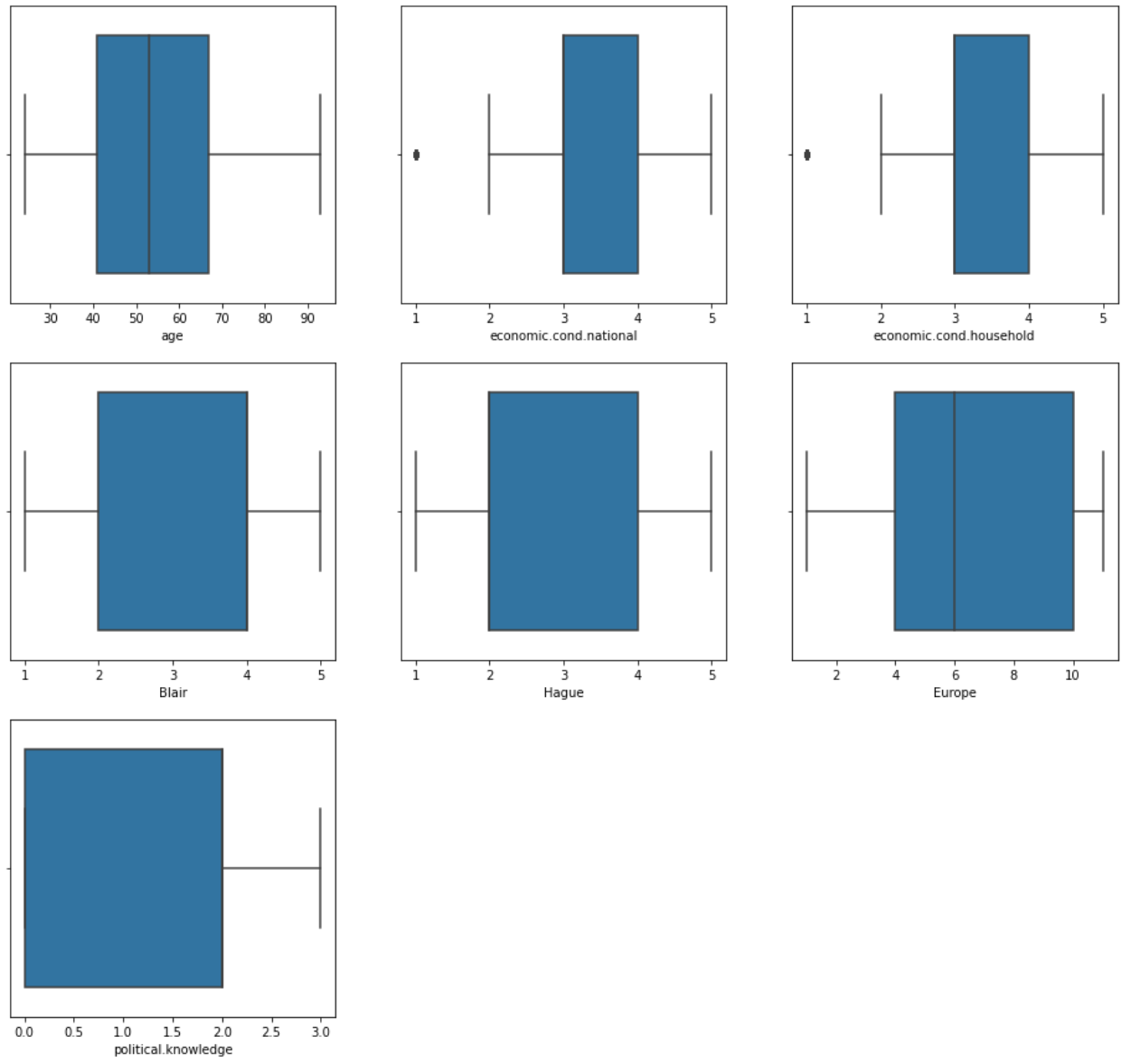
30 40 50 60 70 80 90
age
1 2 3 4 5
economic.cond.national
1 2 3 4 5
economic.cond.household
1 2 3 4 5
Blair
1 2 3 4 5
Hague
2 4 6 8 10
Europe
0.0 0.5 1.0 1.5 2.0 2.5 3.0
political.knowledge

Figure - 02

Countplot for categorical variables :

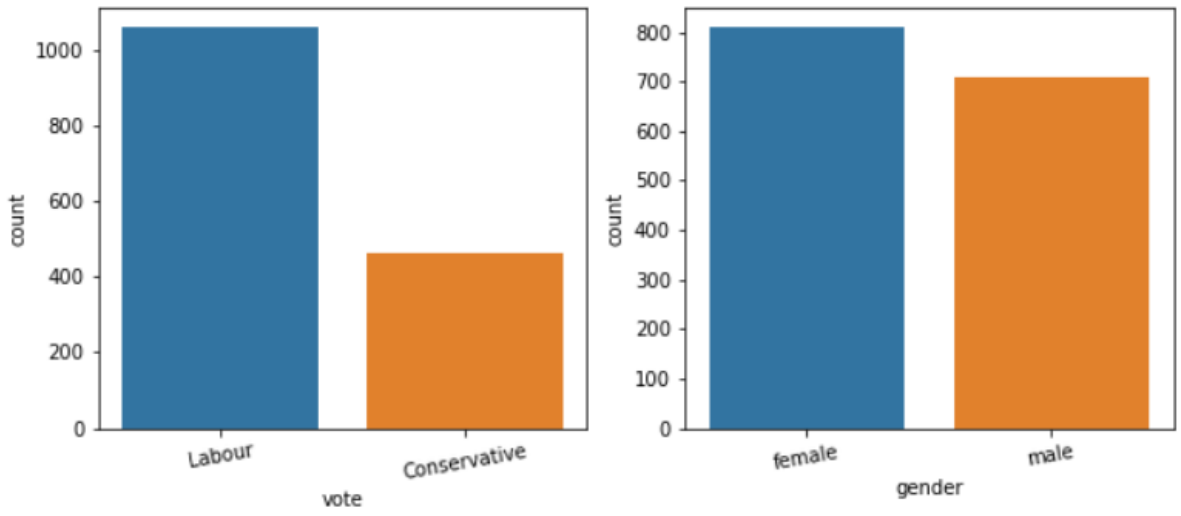

Figure - 03

```
VOTE :  2
Labour          1057
Conservative     460
Name: vote, dtype: int64

*****************************

GENDER :  2
female    808
male      709
Name: gender, dtype: int64

*****************************
```

Bi-variate Analysis:

Swarmplot :

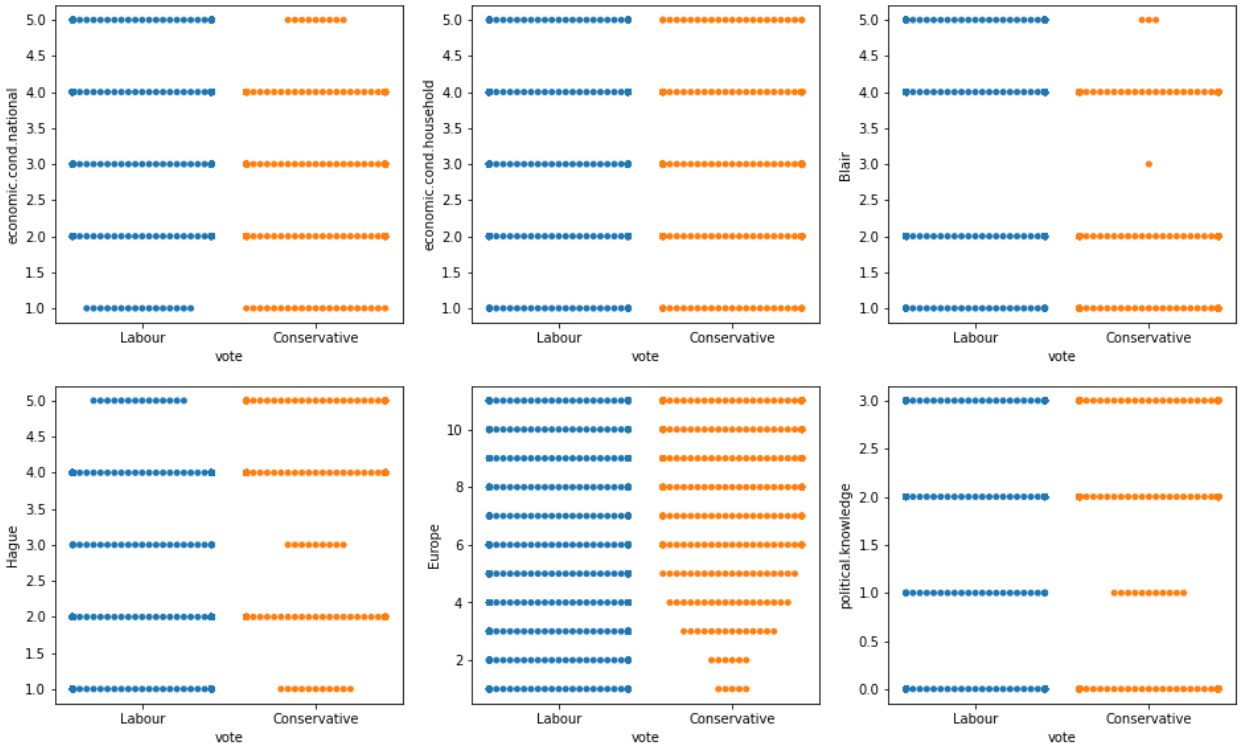

Figure - 04

- The Labour party has 5 ratings more than Conservative party in economic.cond.national.
- The Labour and Conservative party both get nearly equal rating(1 to 5) in economic.cond.household.
- Who have more political knowledge are voting for the Labour party.

Boxplot :

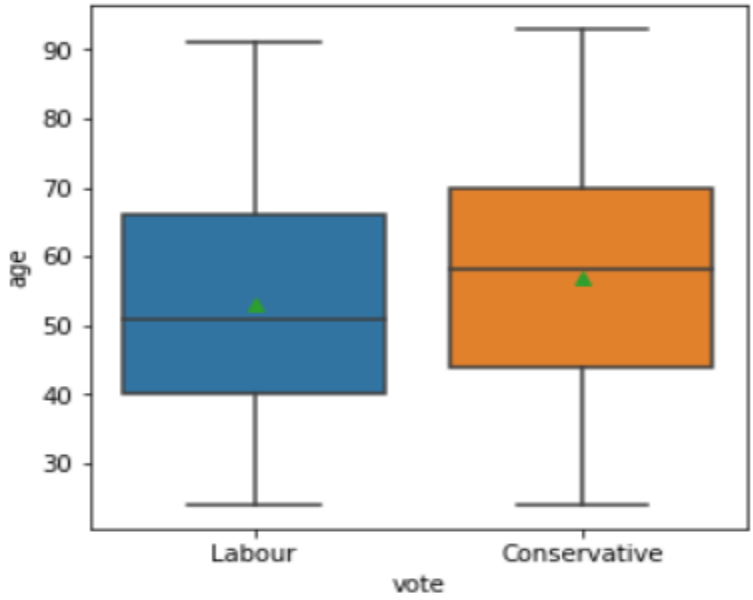

Figure - 05

- Seeing the above boxplot we can say that young people choose the Labour party because the mean age and range (40-67) of the Labour party is less than Conservative range (45-70).

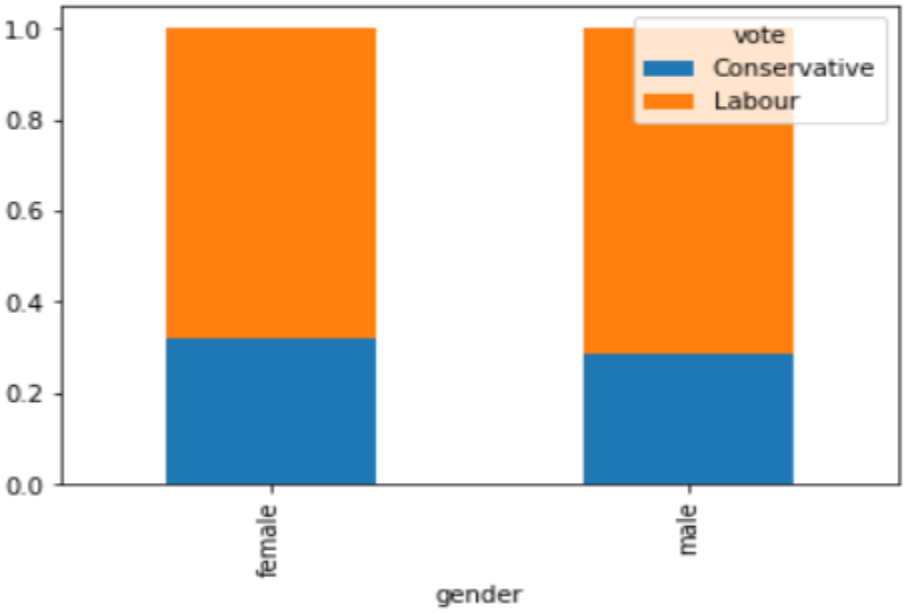

Figure - 06

- Female choosing Conservative party compared to male.

Heatmap :

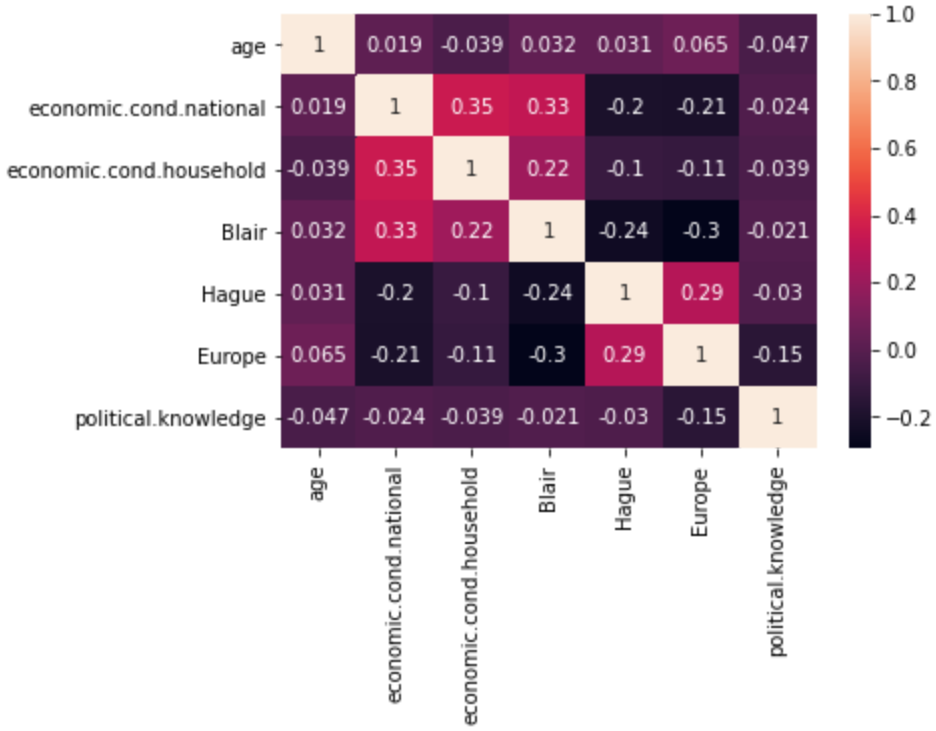

Figure - 07

Pairplot :

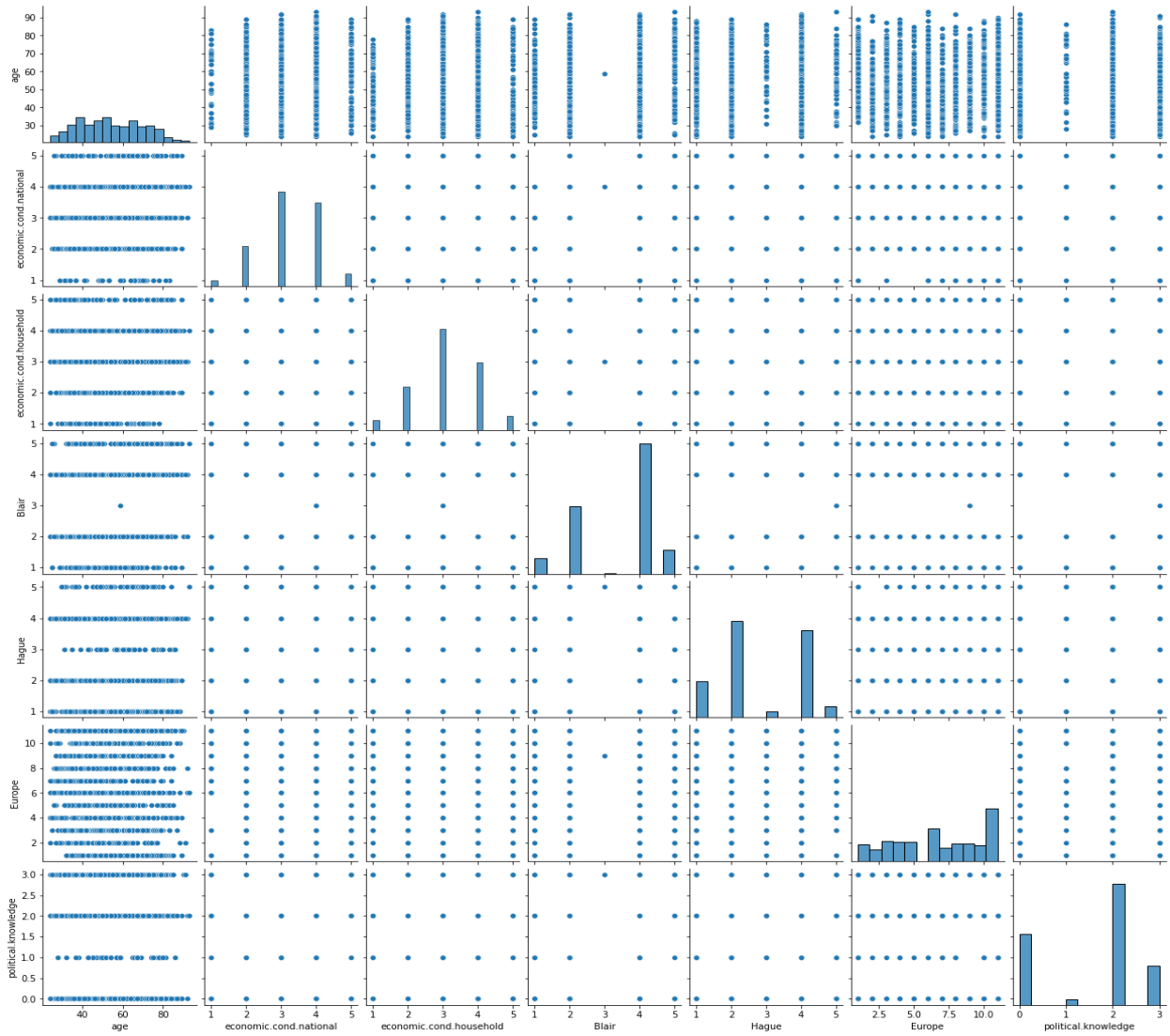

Figure - 08

- After observing heatmap and pairplot figures there is not much multicollinearity.

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).

To build the models we have to change the object data types to numeric values.

```
feature: vote
['Labour', 'Conservative']
Categories (2, object): ['Conservative', 'Labour']
[1 0]


feature: gender
['female', 'male']
Categories (2, object): ['female', 'male']
[0 1]
```

Now we can see the below data types are all numerical values.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1517 entries, 0 to 1524
Data columns (total 8 columns):
 #   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   gender                  1517 non-null   int8
dtypes: int64(7), int8(1)
memory usage: 128.6 KB
```

Is Scaling necessary here or not :

- Scaling is a necessity when using Distance-based models such as KNN etc. Scaling can be done on continuous and ordinal variables.

Data Split:

- Drop the target columns (vote) for the training and testing set.
- For training and testing purposes we are splitting the dataset into train and test data in the ratio 70:30 .
- After splitting the dimensions of the training and test data.

```
X_train (1061, 8)
X_test (456, 8)
train_labels (1061,)
test_labels (456,)
```

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

Logistic Regression :

Grid Search Method :

Grid search method is used for logistic regression to find the optimal solving and the parameters.

```
grid={'penalty':['l2','none','l1'],
      'solver':['sag','lbfgs','newton-cg','saga'],
      'tol':[0.0001,0.00001]}
```

```
LogisticRegression(max_iter=10000, n_jobs=-1, penalty='l1', solver='saga',
                   tol=1e-05)
```

The Grid search method give, solver= saga, penalty= l1

Predicting the training data :

	precision	recall	f1-score	support
0	0.75	0.64	0.69	307
1	0.86	0.91	0.89	754
accuracy			0.83	1061
macro avg	0.80	0.78	0.79	1061
weighted avg	0.83	0.83	0.83	1061

Accuracy of training data = 83.0 %

Predicting the test data :

	precision	recall	f1-score	support
0	0.76	0.73	0.74	153
1	0.86	0.88	0.87	303
accuracy			0.83	456
macro avg	0.81	0.80	0.81	456
weighted avg	0.83	0.83	0.83	456

Accuracy of testing data = 83.0 %

Conclusion : accuracy of training and testing difference is less than 10 % so we can say the model is valid.

LDA (linear discriminant analysis) :

Classification report on training data :

	precision	recall	f1-score	support
0	0.74	0.65	0.69	307
1	0.86	0.91	0.89	754
accuracy			0.83	1061
macro avg	0.80	0.78	0.79	1061
weighted avg	0.83	0.83	0.83	1061

Classification report on test data :

	precision	recall	f1-score	support
0	0.77	0.73	0.74	153
1	0.86	0.89	0.88	303
accuracy			0.83	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.83	0.83	456

Conclusion : accuracy of training and testing difference is less than 10 % so we can say the model is valid.

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

Naïve Bayes Model :

Classification report on training data :

	precision	recall	f1-score	support
0	0.73	0.69	0.71	307
1	0.88	0.90	0.89	754
accuracy			0.84	1061
macro avg	0.80	0.79	0.80	1061
weighted avg	0.83	0.84	0.83	1061

Classification report on test data :

	precision	recall	f1-score	support
0	0.74	0.73	0.73	153
1	0.87	0.87	0.87	303
accuracy			0.82	456
macro avg	0.80	0.80	0.80	456
weighted avg	0.82	0.82	0.82	456

Conclusion : accuracy of training and testing difference is less than 10 % so we can say the model is valid.

KNN Model :

Generally, good KNN performance usually requires preprocessing of data to make all variables similarly scaled and centered.

apply z score on continuous columns and see the performance for KNN.

Build model with everything default :

Classification report on training data :

	precision	recall	f1-score	support
0	0.77	0.71	0.74	307
1	0.88	0.91	0.90	754
accuracy			0.85	1061
macro avg	0.83	0.81	0.82	1061
weighted avg	0.85	0.85	0.85	1061

Classification report on test data :

	precision	recall	f1-score	support
0	0.76	0.71	0.73	153
1	0.86	0.88	0.87	303
accuracy			0.82	456
macro avg	0.81	0.80	0.80	456
weighted avg	0.82	0.82	0.82	456

Run the KNN with no of neighbours to be 1,3,5..19 and *Find the optimal number of neighbours from K=1,3,5,7....19 using the Mis classification error

Note : Misclassification error (MCE) = 1 - Test accuracy score. Calculate MCE for each model with neighbours = 1,3,5...19 and find the model with lowest MCE.

```
array([-0.21710526, -0.0607235 , -0.02935   , -0.03152438, -0.0243109 ,
       -0.00926799, -0.01521239, -0.00924938, -0.00047745, -0.01051846])
```

Plot misclassification error vs k (with k value on X-axis) using matplotlib :

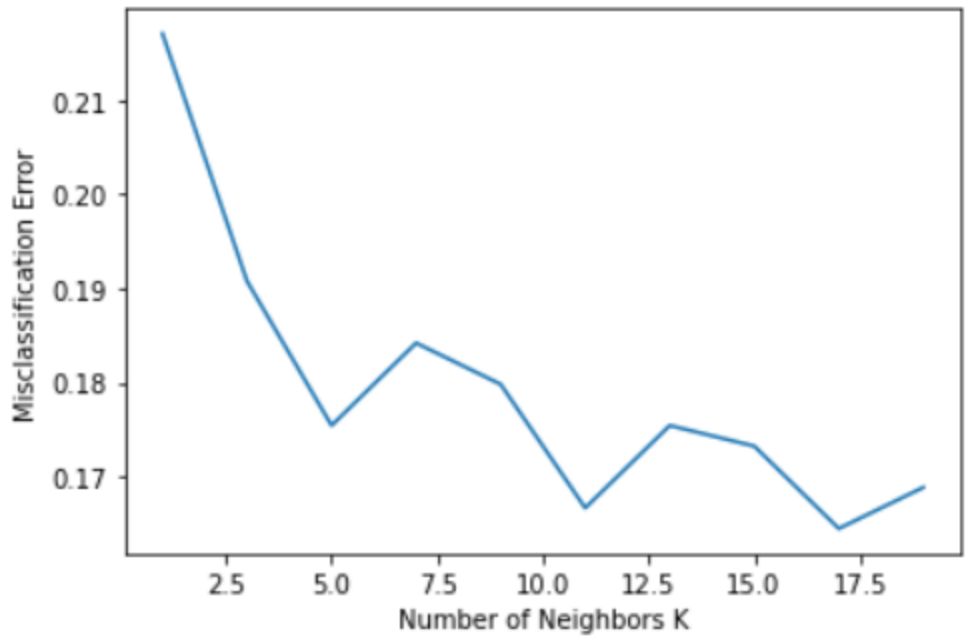

Figure - 09

For K = 11 it is giving the best test accuracy.

Classification report on train data :

	precision	recall	f1-score	support
0	0.75	0.68	0.71	307
1	0.87	0.91	0.89	754
accuracy			0.84	1061
macro avg	0.81	0.79	0.80	1061
weighted avg	0.84	0.84	0.84	1061

Classification report on test data :

	precision	recall	f1-score	support
0	0.79	0.69	0.74	153
1	0.85	0.90	0.88	303
accuracy			0.83	456
macro avg	0.82	0.80	0.81	456
weighted avg	0.83	0.83	0.83	456

Conclusion : accuracy of training and testing difference is less than 10 % so we can say the model is valid.

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

Building a Random Forest Classifier :

grid search for finding out optimal values for the hyper parameters to build a Random Forest Classifier

```
param_grid = {
    'max_depth': [10,12,15],
    'max_features': [3,4,5],
    'min_samples_split': [40,45,60],
    'n_estimators': [101,51]
}

rfcl = RandomForestClassifier(random_state=0)

grid_search = GridSearchCV(estimator = rfcl, param_grid = param_grid, cv = 10)
```

Getting the optimal values for the training dataset :

```
{'max_depth': 12,
 'max_features': 3,
 'min_samples_split': 40,
 'n_estimators': 51}
```

We can check what is the feature importance of the given above optimal values of training dataset.

	Imp
Hague	0.271886
Europe	0.233233
Blair	0.223563
economic.cond.national	0.089782
political.knowledge	0.080115
age	0.061031
economic.cond.household	0.035473
gender	0.004917

Classification report on train data :

	precision	recall	f1-score	support
0	0.82	0.69	0.75	307
1	0.88	0.94	0.91	754
accuracy			0.87	1061
macro avg	0.85	0.82	0.83	1061
weighted avg	0.86	0.87	0.86	1061

Classification report on test data :

	precision	recall	f1-score	support
0	0.82	0.65	0.73	153
1	0.84	0.93	0.88	303
accuracy			0.84	456
macro avg	0.83	0.79	0.80	456
weighted avg	0.83	0.84	0.83	456

Conclusion : accuracy of training and testing difference is less than 10 % so we can say the model is valid.

Bagging Classifier :

Classification report on train data :

	precision	recall	f1-score	support
0	0.97	0.94	0.95	307
1	0.98	0.99	0.98	754
accuracy			0.97	1061
macro avg	0.97	0.96	0.97	1061
weighted avg	0.97	0.97	0.97	1061

Classification report on test data :

	precision	recall	f1-score	support
0	0.79	0.67	0.72	153
1	0.84	0.91	0.88	303
accuracy			0.83	456
macro avg	0.82	0.79	0.80	456
weighted avg	0.83	0.83	0.82	456

Conclusion : accuracy of training and testing difference is not less than 10 % so we can say the model is not valid .

This is an overfitting model.

Gradient Boosting :

Classification report on training data :

	precision	recall	f1-score	support
0	0.84	0.78	0.81	307
1	0.91	0.94	0.93	754
accuracy			0.89	1061
macro avg	0.88	0.86	0.87	1061
weighted avg	0.89	0.89	0.89	1061

Classification report on test data :

	precision	recall	f1-score	support
0	0.80	0.69	0.74	153
1	0.85	0.91	0.88	303
accuracy			0.84	456
macro avg	0.82	0.80	0.81	456
weighted avg	0.83	0.84	0.83	456

Conclusion : accuracy of training and testing difference is less than 10 % so we can say the model is valid.

Ada Boost :

Classification report on testing data :

	precision	recall	f1-score	support
0	0.76	0.70	0.73	307
1	0.88	0.91	0.90	754
accuracy			0.85	1061
macro avg	0.82	0.80	0.81	1061
weighted avg	0.85	0.85	0.85	1061

Classification report on test data :

	precision	recall	f1-score	support
0	0.75	0.67	0.71	153
1	0.84	0.88	0.86	303
accuracy			0.81	456
macro avg	0.79	0.78	0.79	456
weighted avg	0.81	0.81	0.81	456

Conclusion : accuracy of training and testing difference is less than 10 % so we can say the model is valid.

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.

Comparison of the performance metrics of all models:

	Accuracy	AUC score
Logistic regression train	0.83	0.890
Logistic regression test	0.83	0.882
LDA train	0.83	0.889
LDA test	0.83	0.888
KNN Model train	0.84	0.909
KNN Model test	0.83	0.890
Naïve Bayes train	0.84	0.888
Naïve Bayes test	0.82	0.876
Random Forest train	0.87	0.927
Random Forest test	0.84	0.895
Bagging Classifier train	0.97	0.997
Bagging Classifier test	0.83	0.896
Gradient Boosting train	0.89	0.951
Gradient Boosting test	0.84	0.899
Ada Boost train	0.85	0.915

Ada Boost test	0.81	0.877

Table - 03

In the confusion matrix we get 2X2 table :

True Negative (TN) :

- The predicted value matches the actual value
- The actual value was negative and the model predicted a negative value

False Positive (FP) – Type 1 error :

- The predicted value was falsely predicted
- The actual value was negative but the model predicted a positive value
- Also known as the Type 1 error

False Negative (FN) – Type 2 error :

- The predicted value was falsely predicted
- The actual value was positive but the model predicted a negative value
- Also known as the Type 2 error

True Positive (TP) :

- The predicted value matches the actual value
- The actual value was positive and the model predicted a positive value

	TN[0][0]	**FP[0][1]**	**FN[1][0]**	**TP[1][1]**
Logistic regression train	196	111	66	688
Logistic regression test	111	42	36	267
LDA train	200	107	69	685
LDA test	111	42	34	269
KNN Model train	209	98	69	685
KNN Model test	106	47	29	274
Naïve Bayes train	211	96	79	675
Naïve Bayes test	112	41	40	263
Random Forest train	213	94	48	706

Random Forest test	99	54	21	282
Bagging Classifier train	289	18	10	744
Bagging Classifier test	102	51	27	276
Gradient Boosting train	239	68	46	708
Gradient Boosting test	105	48	27	276
Ada Boost train	241	93	66	688
Ada Boost test	103	50	35	268

Table - 04

Comparison between precision, recall and f1-score :

- recall(1):- it means how many of the actual true data points are identified as true data points by the model.
- precision(1):- it means among the points by the model, how many are really positive.
- F1 score - F1 Score is the weighted average of Precision and Recall.
- F1 Score = 2*(Recall * Precision) / (Recall + Precision).

	precision-0	precision-1	recall-0	recall-1	f1-score-0	f1-score -1
Logistic regression train	0.75	0.86	0.64	0.91	0.69	0.89
Logistic regression test	0.76	0.86	0.73	0.88	0.74	0.87
LDA train	0.74	0.86	0.65	0.91	0.69	0.89
LDA test	0.77	0.86	0.73	0.89	0.74	0.88
KNN Model train	0.75	0.87	0.68	0.91	0.71	0.89
KNN Model test	0.79	0.85	0.69	0.90	0.74	0.88
Naïve Bayes train	0.73	0.88	0.69	0.90	0.71	0.89
Naïve Bayes test	0.74	0.87	0.73	0.87	0.73	0.87
Random Forest train	0.82	0.88	0.69	0.94	0.75	0.91
Random Forest test	0.82	0.84	0.65	0.93	0.73	0.88
Bagging Classifier	0.97	0.98	0.94	0.99	0.95	0.98

train						
Bagging Classifier test	0.79	0.84	0.67	0.91	0.72	0.88
Gradient Boosting train	0.84	0.91	0.78	0.94	0.81	0.93
Gradient Boosting test	0.80	0.85	0.69	0.91	0.74	0.88
Ada Boost train	0.76	0.88	0.70	0.91	0.73	0.90
Ada Boost test	0.75	0.84	0.67	0.88	0.71	0.86

Table - 05

Plot ROC curve and get ROC_AUC score for each model :

- AUC stands for area under the curve.
- ROC graph is a trade off between True positive rate and false positive rate.

AUC ROC curve for Logistic Regression Train :

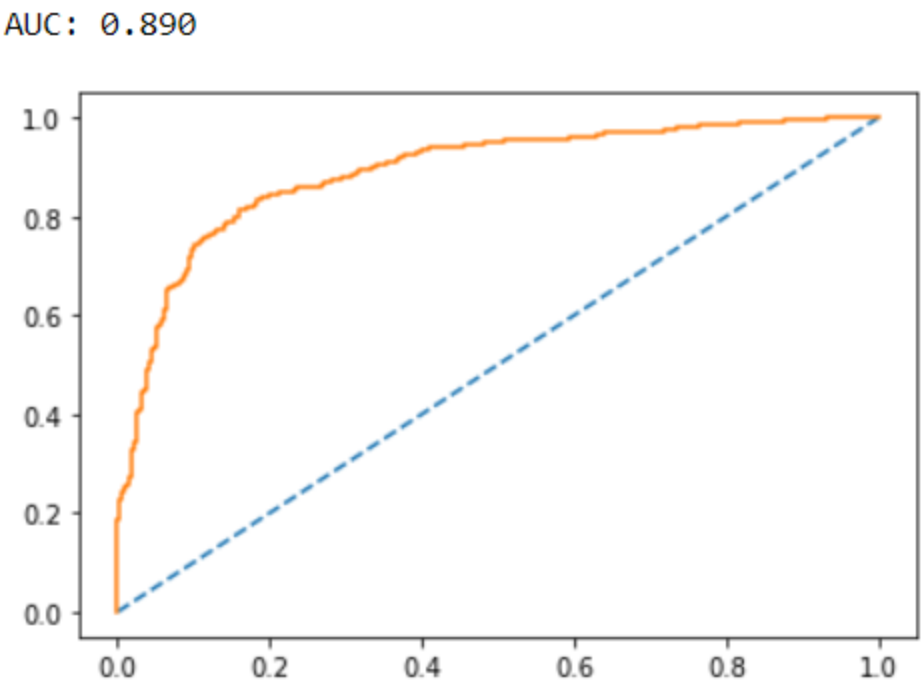

Figure - 10

AUC ROC curve for Logistic Regression Test :

AUC: 0.882

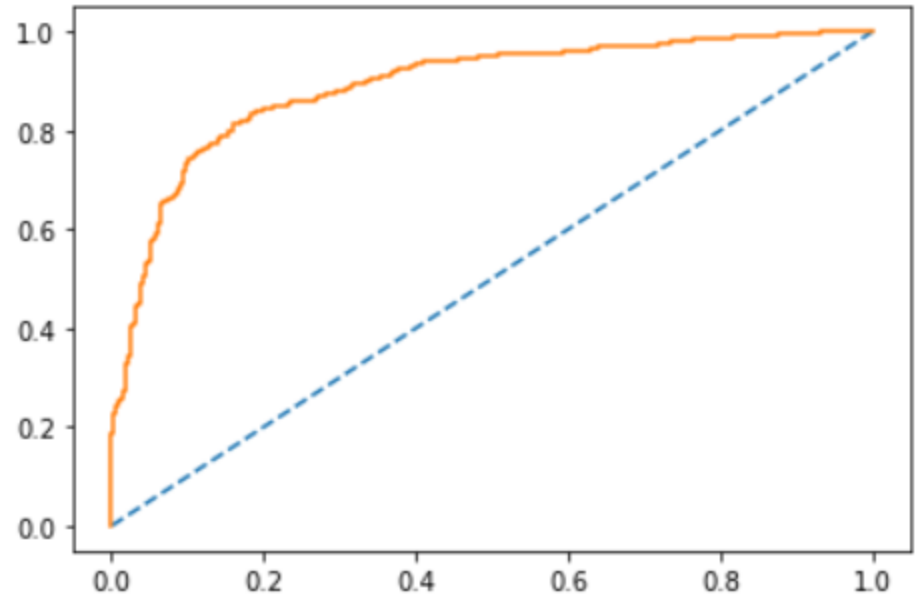

Figure -11

AUC ROC curve for LDA Train :

AUC: 0.889

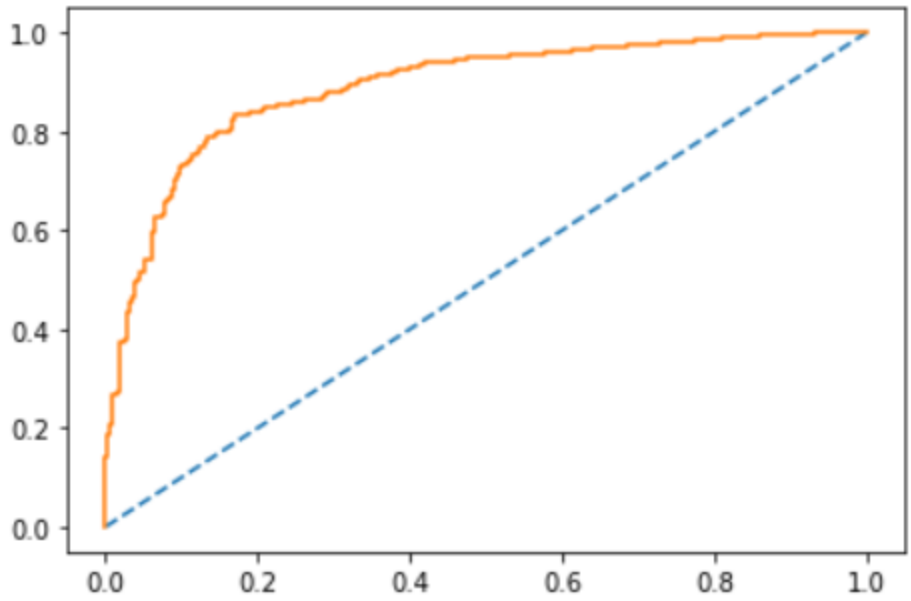

Figure - 12

AUC ROC curve for LDA Test :

```
AUC: 0.888
```

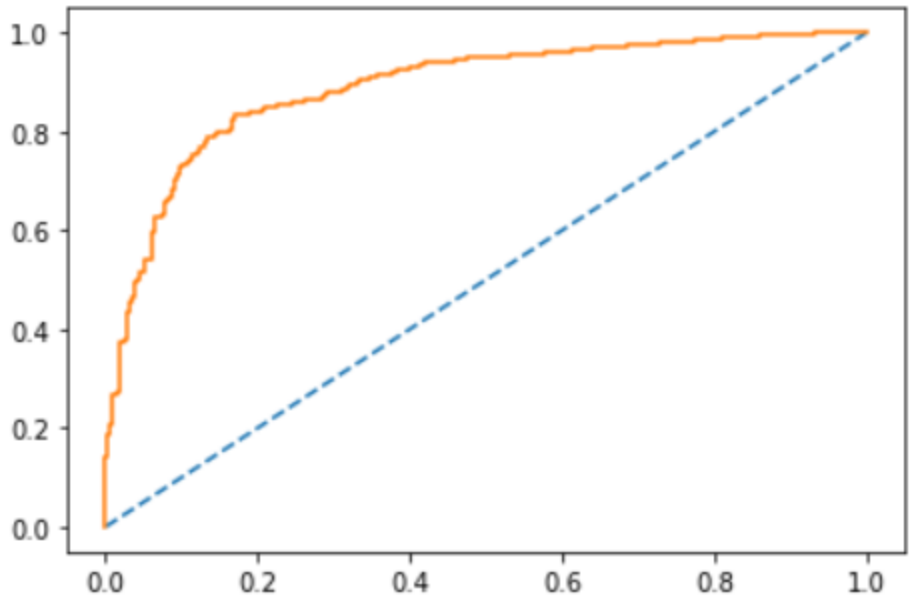

Figure - 13

AUC ROC curve for Naïve Bayes Model Train :

```
AUC: 0.888
```

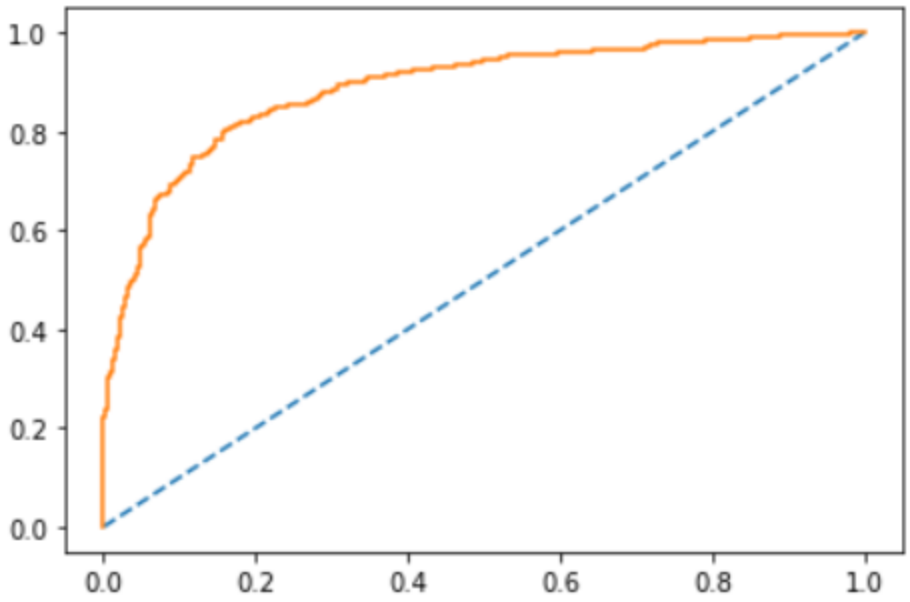

Figure - 14

AUC ROC curve for Naïve Bayes Model test :

AUC: 0.876

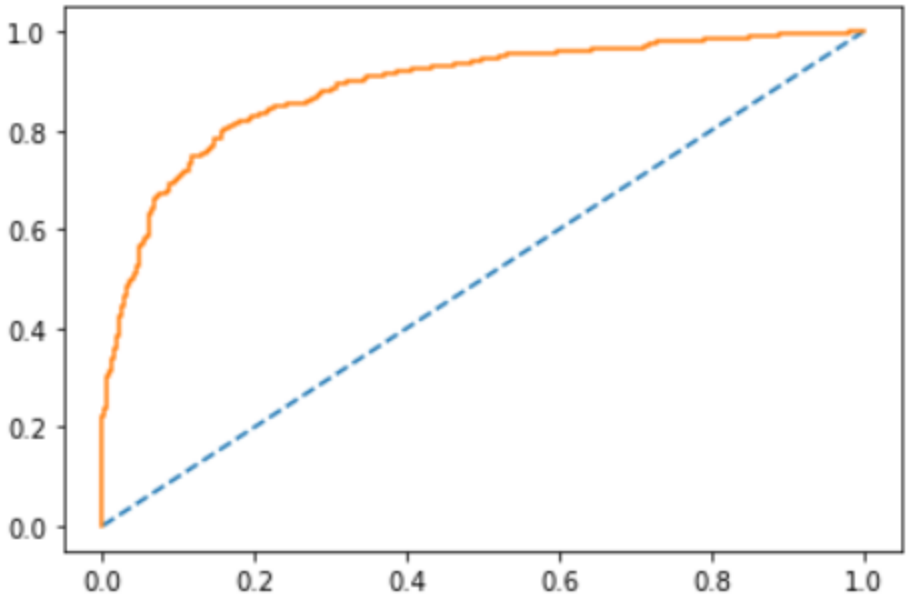

Figure - 15

AUC ROC curve for KNN Model train :

AUC: 0.909

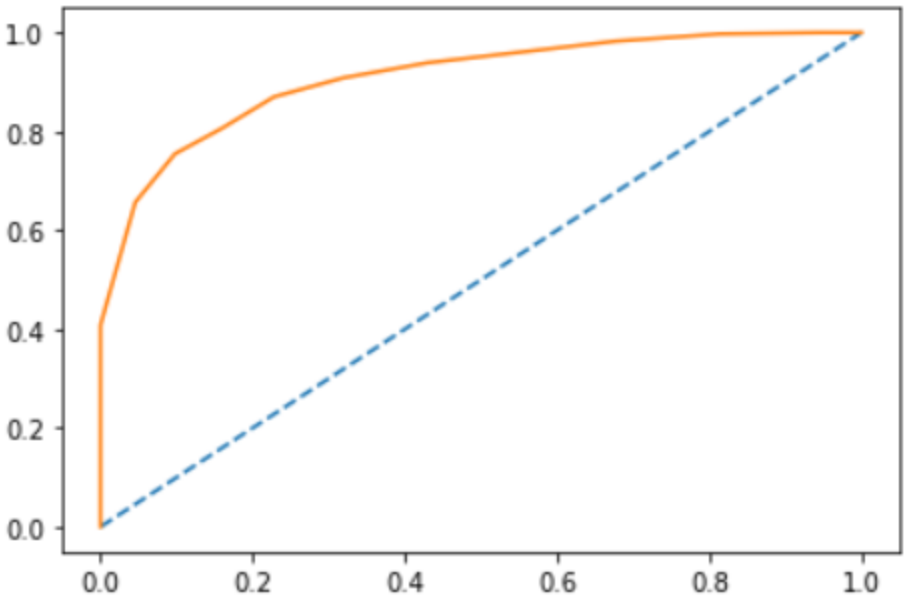

Figure - 16

AUC ROC curve for KNN Model test :

```
AUC: 0.890
```

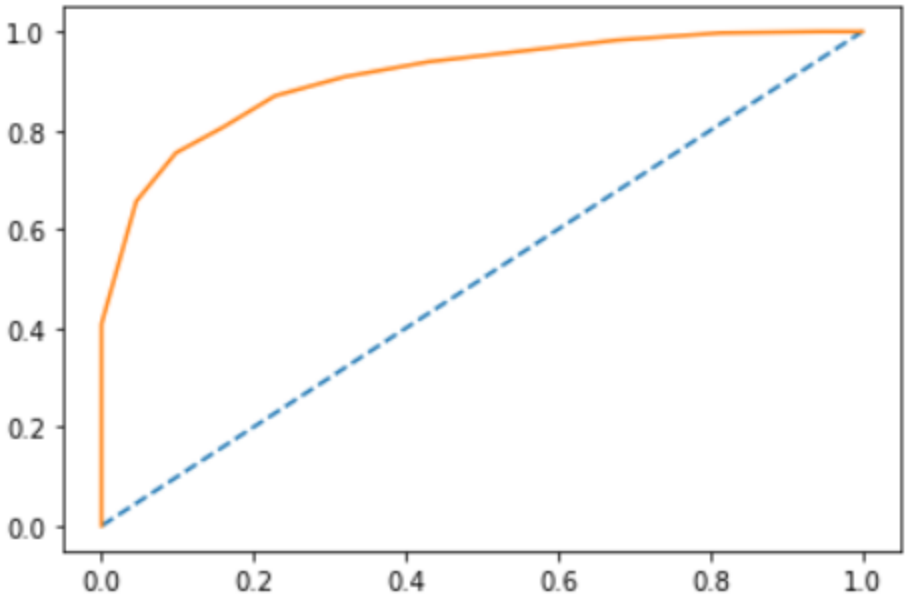

Figure - 17

AUC ROC curve for Random forest Model Train :

```
AUC: 0.927
```

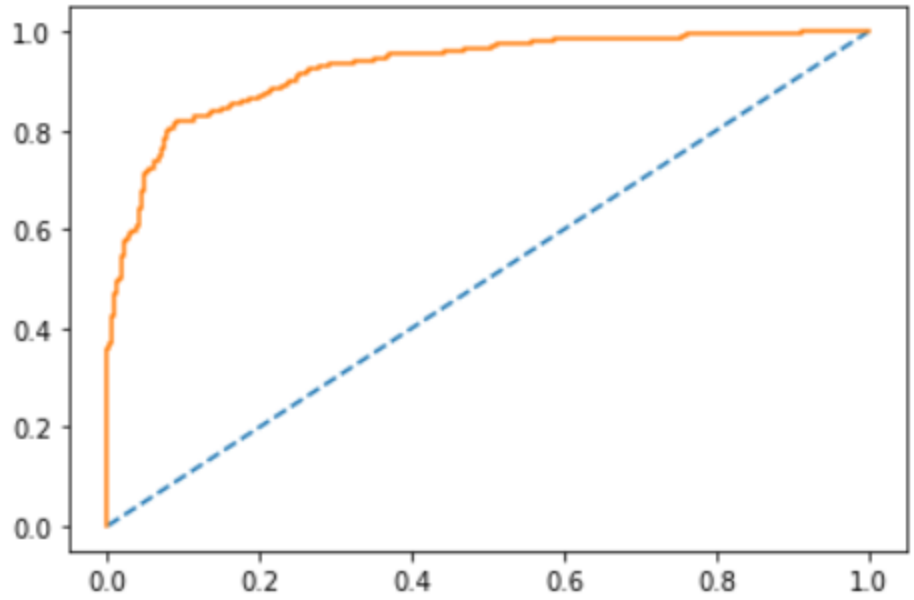

Figure - 18

AUC ROC curve for Random forest Model test :

```
AUC: 0.895
```

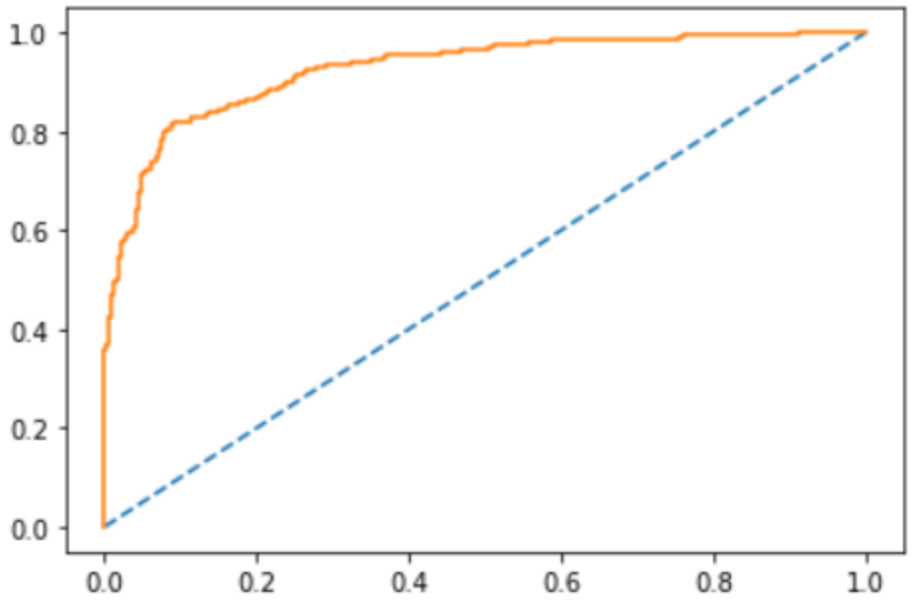

Figure - 19

AUC ROC curve for BaggingClassifier Model Train :

```
AUC: 0.997
```

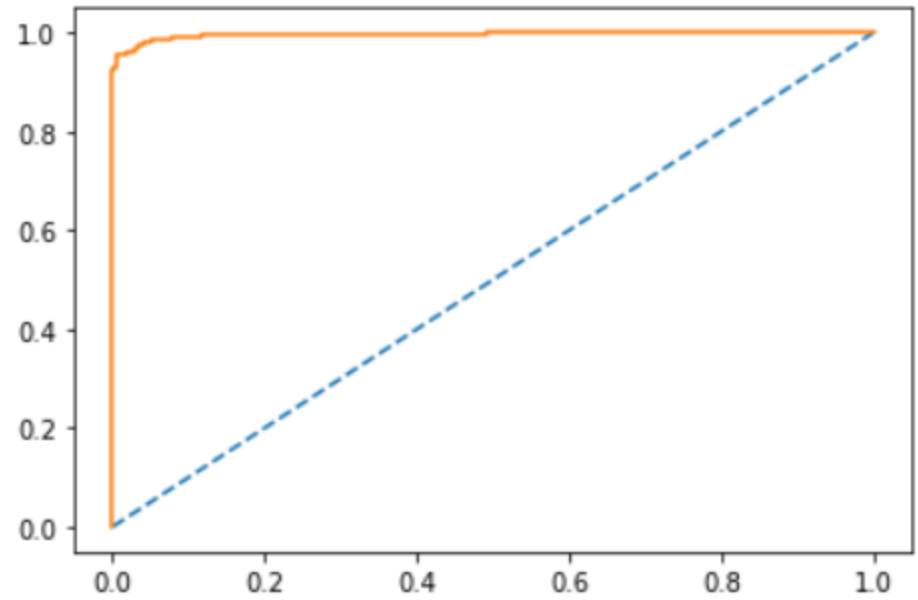

Figure - 20

AUC ROC curve for BaggingClassifier Model test :

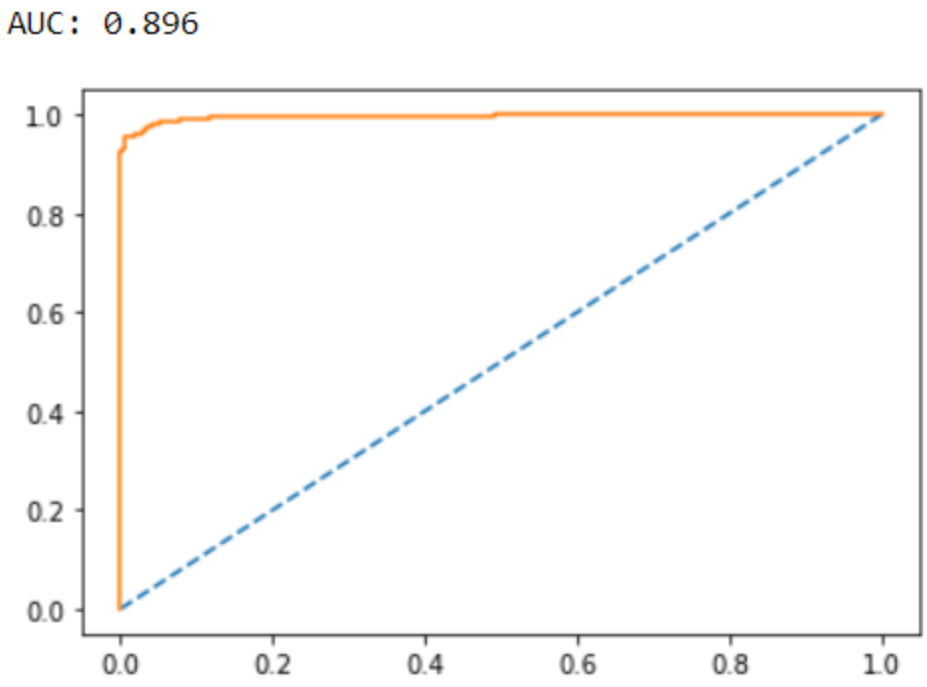

Figure - 21

AUC ROC curve for Gradient Boosting Model Train :

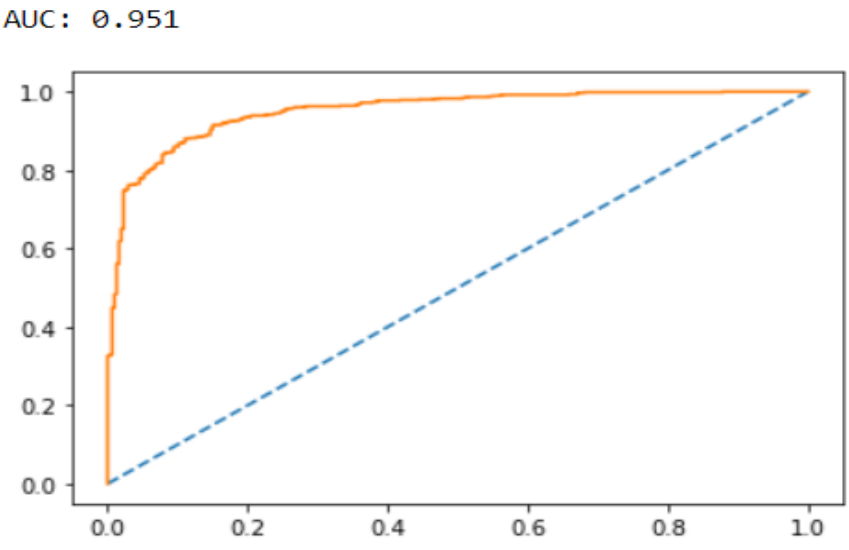

Figure - 22

AUC ROC curve for Gradient Boosting Model test :

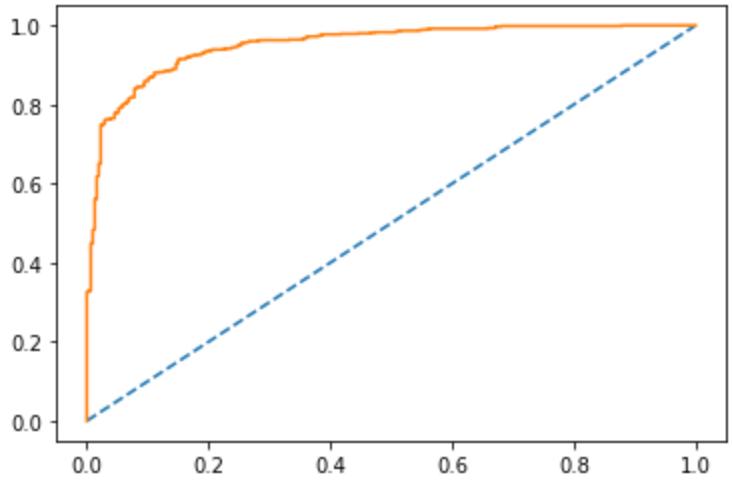

Figure - 23

AUC ROC curve for Ada Boost Model Train :

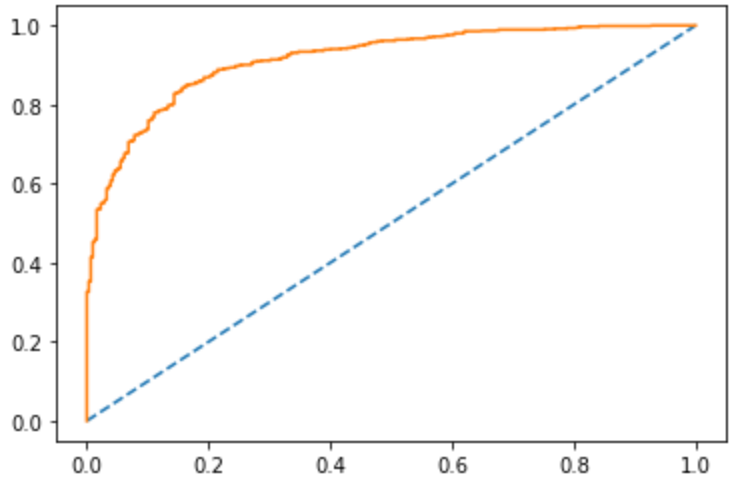

Figure - 24

AUC ROC curve for Ada Boost Model test :

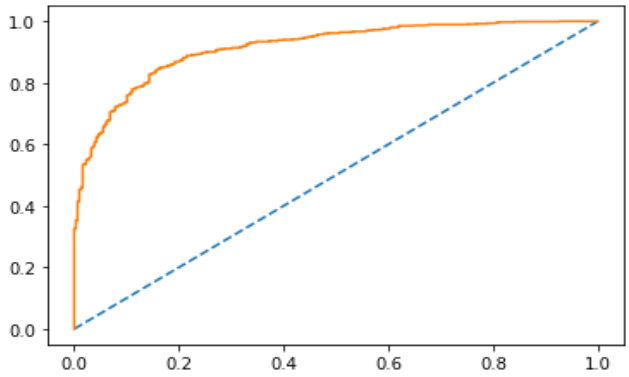

Figure - 25

Conclusion :

- After observing each and every model I go with the Gradient Boosting model because it has the highest accuracy on train and test. Model is also valid.
- In this problem both Classes are equally important.

1.8 Based on these predictions, what are the insights?

- In the datasets most of the data is ordinal variables.
- The Labour party gets more votes than Conservative.
- Who have more political knowledge are voting for the Labour party.
- Mid-age range people voting the Labour party.
- Female voting Conservative compared to male.

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 1973

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

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

data	**characters**	**words**	**sentences**
1941-Roosevelt.txt	7571	1536	68
inaugural.raw('1961-Kennedy.txt')	7618	1546	52
inaugural.raw('1973-Nixon.txt')	9991	2028	69

Table-06

2.2 Remove all the stopwords from all three speeches.

For the removing stopwords download the below point

- nltk.download("stopwords")
- nltk.download("punkt")

I also add some stopwords based on data.

Eg. list_stop=['--',"let",'know',"us","day","since"]

After that download nltk.download('wordnet') and lemetize the data.

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)

1941-Roosevelt.txt :

words	No. of occurrence
life	6
nation	5
human	5

Table - 07

inaugural.raw('1961-Kennedy.txt') :

words	No. of occurrence
pledge	7
side	7
ask	5

Table - 08

inaugural.raw('1973-Nixon.txt') :

words	No. of occurrence
peace	11
great	9
policies	7

Table - 09

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

1941-Roosevelt.txt :

Figure-26

inaugural.raw('1961-Kennedy.txt') :

Figure - 27

inaugural.raw('1973-Nixon.txt') :

Figure - 28