8/12/2021

Machine Learning Report PGP -DSBA

Saloni Juwatkar
PGP – DATA SCIENCE AND BUSINESS ANALYTICS

Table of Contents

1 Problem Statement: 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. 8

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). 13

1.4 Apply Logistic Regression and LDA (linear discriminant analysis). 14

1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results. 14

1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting. 14

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. ..14

1.8 Based on these predictions, what are the insights? 28

2 Problem Statement: 2 29

2.1 Find the number of characters, words, and sentences for the mentioned documents 29

2.2 Remove all the stopwords from all three speeches. 30

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) 31

2.4 Plot the word cloud of each of the speeches of the variable. 32

List of Figures

Figure 1: PS:1 Sample Dataset 5
Figure 2: PS:1: Sample Tail Dataset 5
Figure 3: PS:1: Shape of Dataset 5
Figure 4: PS:1: Data Description 6
Figure 5: PS:1: Data Info 6
Figure 6: PS:1: Unique Data 6
Figure 7: PS:1 Data before and after duplicate Data imputation 6
Figure 8: PS:1: Missing Data 7
Figure 9: PS:1: Univariate Analysis 8
Figure 10: PS:1: Boxplot Before Outlier Treatment 9
Figure 11: PS:1: Boxplot after Outlier Treatment 10
Figure 12: PS1-Data after Scaling 10
Figure 13: PS:1: Pair plot 11
Figure 14: PS:1: Heatmap 12
Figure 15: PS:1: Encoding 13
Figure 16: PS:1: Data Encoded Dataframe 13
Figure 17: PS:1: Data Info 13
Figure 18: PS:1: X- Frame 14
Figure 19: PS:1: Train and Test Dataset 14
Figure 20: LR-Classification Report 14
Figure 21: LR-Test Data Confusion Matrix 15
Figure 22: LR-Train Data Classification Report 15
Figure 23: LR-Train Data Confusion Matrix 15
Figure 24: LDA-Test Data Classification Report 16
Figure 25: LDA - Test Data Confusion Matrix 16
Figure 26: LDA - Train Data Classification Report 17
Figure 27: LDA - Train Data Confusion Matrix 17
Figure 28: KNN 17
Figure 29: KNN- Test Data Classification Report 18
Figure 30: KNN - Test Data Confusion Matrix 18
Figure 31: KNN- Test Data AUC Curve 18
Figure 32: KNN - Train Data Classification Report 19
Figure 33: KNN - Train Data Confusion Report 19
Figure 34: KNN Train Data AUC Curve 19
Figure 35: NB Test Data Classification Reports 20
Figure 36: NB Test Data Confusion Matrix 20
Figure 37: NB Test Data AUC Curve 20
Figure 38: NB Train Data Classification Report 21
Figure 39: NB Train Data Confusion Matrix 21
Figure 40: NB Train Data AUC Curve 21
Figure 41: Bagging Test Data Classification Report 22

Figure 42: Bagging Test Data Confusion Matrix....22
Figure 43: Bagging Test Data AUC Curve....22
Figure 44: Bagging Train Data Classification Report....23
Figure 45: Bagging Train Data Confusion Matrix....23
Figure 46: Bagging Train Data AUC Curve....23
Figure 47: Boosting Test Data Classification Report....24
Figure 48: Boosting Test Data Confusion Matrix....24
Figure 49: Boosting Test Data AUC Curve....24
Figure 50: Boosting Train Data Classification Report....25
Figure 51: Boosting Train Data Confusion Matrix....25
Figure 52: Boosting Train Data AUC Curve....25
Figure 53: Gradient Boosting Test Data Classification Report....26
Figure 54: Gradient Boosting Test Data Confusion Matrix....26
Figure 55: Gradient Boosting Test Data AUC Curve....26
Figure 56: Gradient Boosting Train Data Classification Report....27
Figure 57: Gradient Boosting Train Data Confusion Matrix....27
Figure 58: Gradient Boosting Train Data AUC Curve....27
Figure 59: Roosevelt File....30
Figure 60: Kennedy File....30
Figure 61: Nixon File....30
Figure 62: WordCloud for Roosevelt File....32
Figure 63: WordCloud for Kennedy File....32
Figure 64: WordCloud for Nixon File....33

List of Tables

Table 1: Comparison of Models (1)....28
Table 2: Comparison of Models (2)....28

1 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.

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

The csv file was read and EDA was done and the following were the inferences drawn from the EDA.

Exploratory Data Analysis

	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

Figure 1: PS:1 Sample Dataset

	Unnamed: 0	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
1520	1521	Conservative	67	5	3	2	4	11	3	male
1521	1522	Conservative	73	2	2	4	4	8	2	male
1522	1523	Labour	37	3	3	5	4	2	2	male
1523	1524	Conservative	61	3	3	1	4	11	2	male
1524	1525	Conservative	74	2	3	2	4	11	0	female

Figure 2: PS:1: Sample Tail Dataset

```
data_df.shape
```

```
(1525, 9)
```

Figure 3: PS:1: Shape of Dataset

- Dropping the redundant column ‘Unnamed:0’:

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
count	1525.000000	1525.000000	1525.000000	1525.000000	1525.000000	1525.000000	1525.000000
mean	54.182295	3.245902	3.140328	3.334426	2.746885	6.728525	1.542295
std	15.711209	0.880969	0.929951	1.174824	1.230703	3.297538	1.083315
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

Figure 4: PS:1: Data Description

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   vote                    1525 non-null   object
 1   age                     1525 non-null   int64
 2   economic.cond.national  1525 non-null   int64
 3   economic.cond.household 1525 non-null   int64
 4   Blair                   1525 non-null   int64
 5   Hague                   1525 non-null   int64
 6   Europe                  1525 non-null   int64
 7   political.knowledge     1525 non-null   int64
 8   gender                  1525 non-null   object
dtypes: int64(7), object(2)
memory usage: 107.4+ KB
```

Figure 5: PS:1: Data Info

```
VOTE   2
Conservative     460
Labour          1057
Name: vote, dtype: int64
GENDER   2
male      709
female    808
Name: gender, dtype: int64
```

Figure 6: PS:1: Unique Data

Duplicate Data Imputation:

- Number of duplicate rows found in the dataset were 8. These are dropped so as to getter a better prediction and can draw useful insights from the model.

```
Total no of duplicate values = 8

Total no of duplicate values = 0
```

Figure 7: PS:1 Data before and after duplicate Data imputation

Missing/ Null Value Treatment

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

Figure 8: PS:1: Missing Data

Inference

- On performing the descriptive analysis, we can see that there are a few columns having categorical values but are not having the data type “object”
- The Election dataset have 1525 rows and 9 columns. All the variables except vote and gender are int64 datatypes.
- ‘vote’ has two unique values Labour and Conservative, which is also a dependent variable.
- ‘gender’ has two unique values male and female.
- There are no null values in the data set.
- There are 8 duplicate rows. Even though they could represent different person with exact same profile and political outlook, we drop these rows as they are few in number and add no value to the data set.

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

Univariate Analysis

Univariate analysis refers to the analysis of a single variable. The main purpose of univariate analysis is to summarize and find patterns in the data.

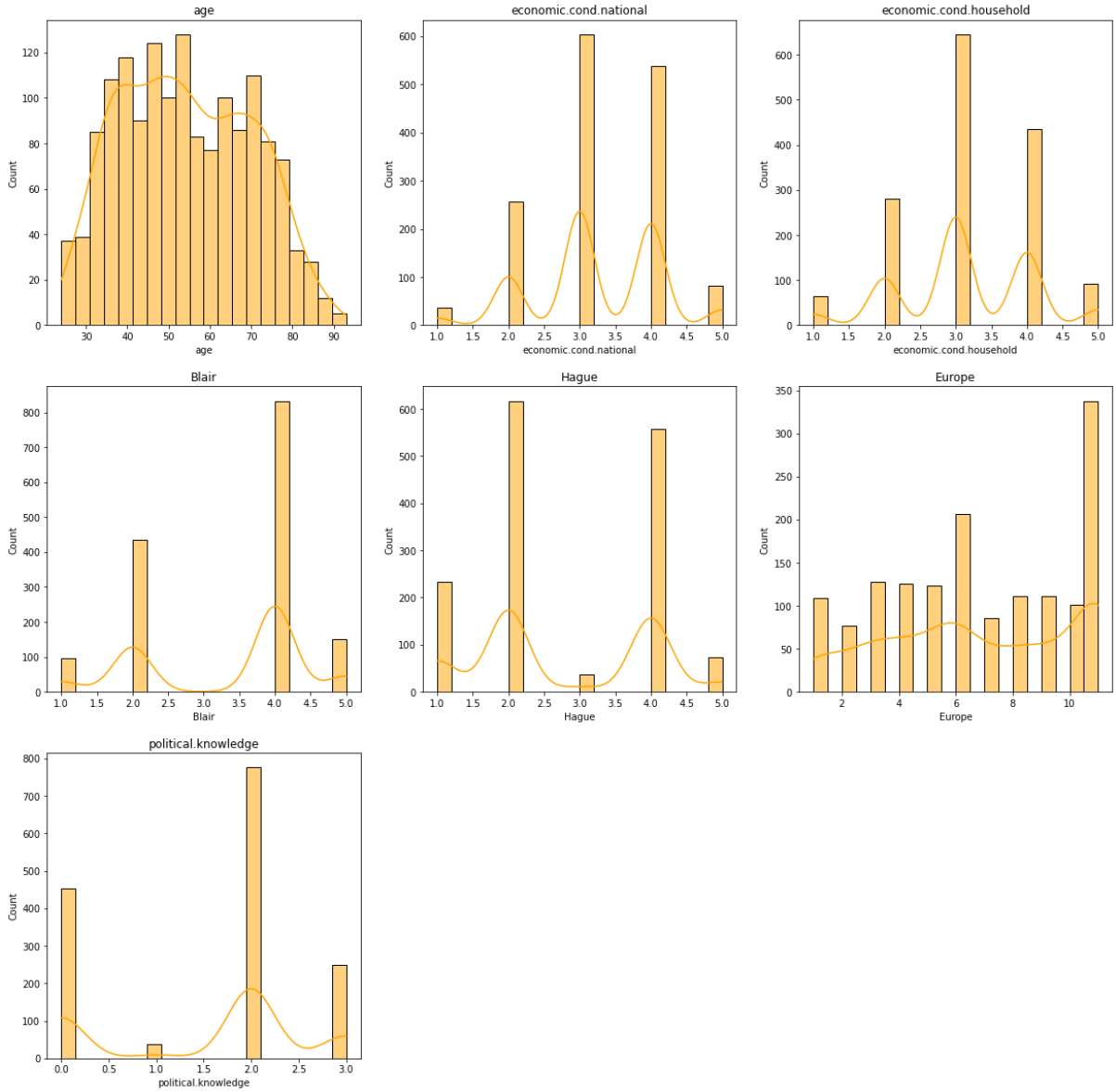

Figure 9: PS:1: Univariate Analysis

Inference:

- Distribution of "age" resembles normal distribution and is slightly right skewed. Most of the respondents in this data is in the age bracket 40-60. Mean of the age variable is greater than the median followed by mode
- 'female' voters large in number than 'male'
- Labour gets the highest voting from both female and male voters. Almost in all the categories Labour is getting the maximum votes.
- Distribution of "economic.cond.national" is not normal and it is slightly left skewed. . Mean is greater than Median. Out of the 1525 participants around 600 participants rated the national economic condition as more than average (i.e scale of 3-3.3).
- Distribution of "economic.cond.household" is not normal and it is slightly left skewed. Mean is greater than Median. Out of the 1525 participants around 650 participants rated the household economic condition as more than average (i.e scale of 3-3.4).
- Distribution of "Blair" is not normal and it is slightly left skewed. Mode is greater than mean. (Around 850 nos.) 55.74% of participants has given above average Assessment of 3.7 to 4.3 for Labour leader.
- Distribution of "Hague" is not normal and it is slightly right skewed. This variable is not normally distributed and skewed. Mean is greater than mode. (Around 625 nos.) 40.98% of participants have only given above average rating of 3- 5.
- Distribution of "Europe" is somewhat normal and it is left skewed. Mode is higher than mean followed by Median. Around (950 no's) 62.30% of participants have given rating of more than 6 in the scale, which shows that majority of the participants are much sceptical about European integration.

Boxplot For Outliers Treatment

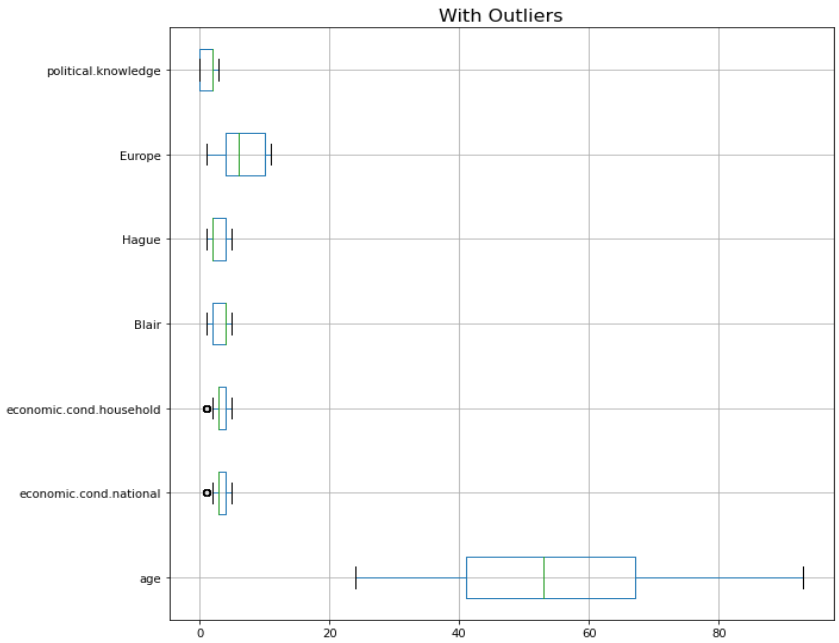

Figure 10: PS:1: Boxplot Before Outlier Treatment

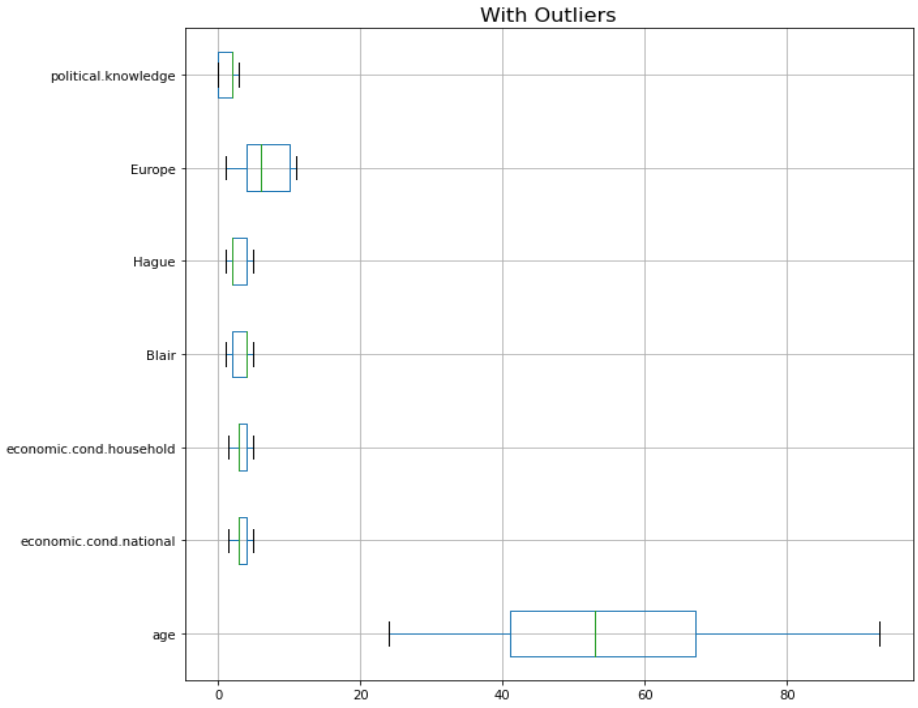

Figure 11: PS:1: Boxplot after Outlier Treatment

- The outliers are treated with percentile method with which the dataset if ready to be used for building regression model.
- The boxplot distribution of the continuous variables shows that there are marginal outliers in two variables: economic.cond.national and economic.cond.household.

Scaling (Answer of 1.3)

Scaling: Scaling is required as continuous variables are of different scales and need to normalize the data using Standard Scaler.

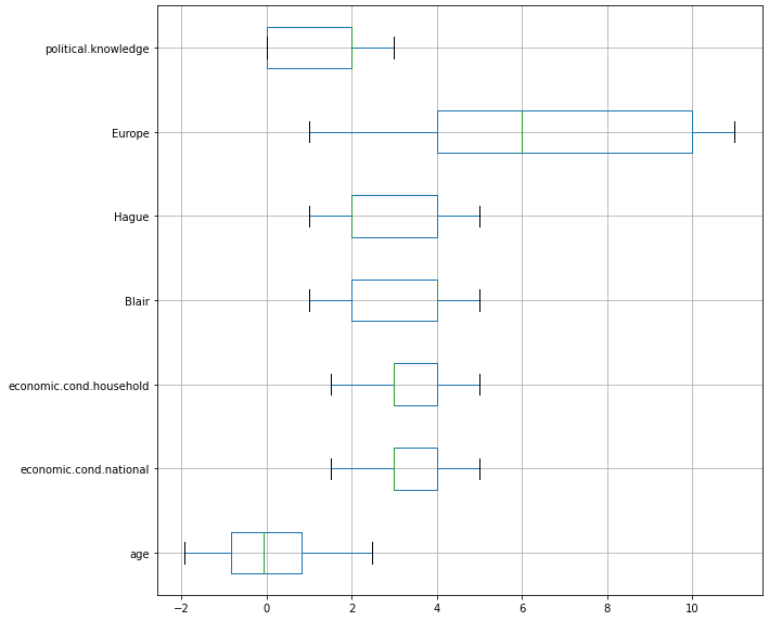

Figure 12: PS1-Data after Scaling

Bivariate Analysis

Bivariate analysis is the simultaneous analysis of two variables (attributes). It explores the concept of relationship between two variables, whether there exists an association and the strength of this association, or whether there are differences between two variables and the significance of these differences.

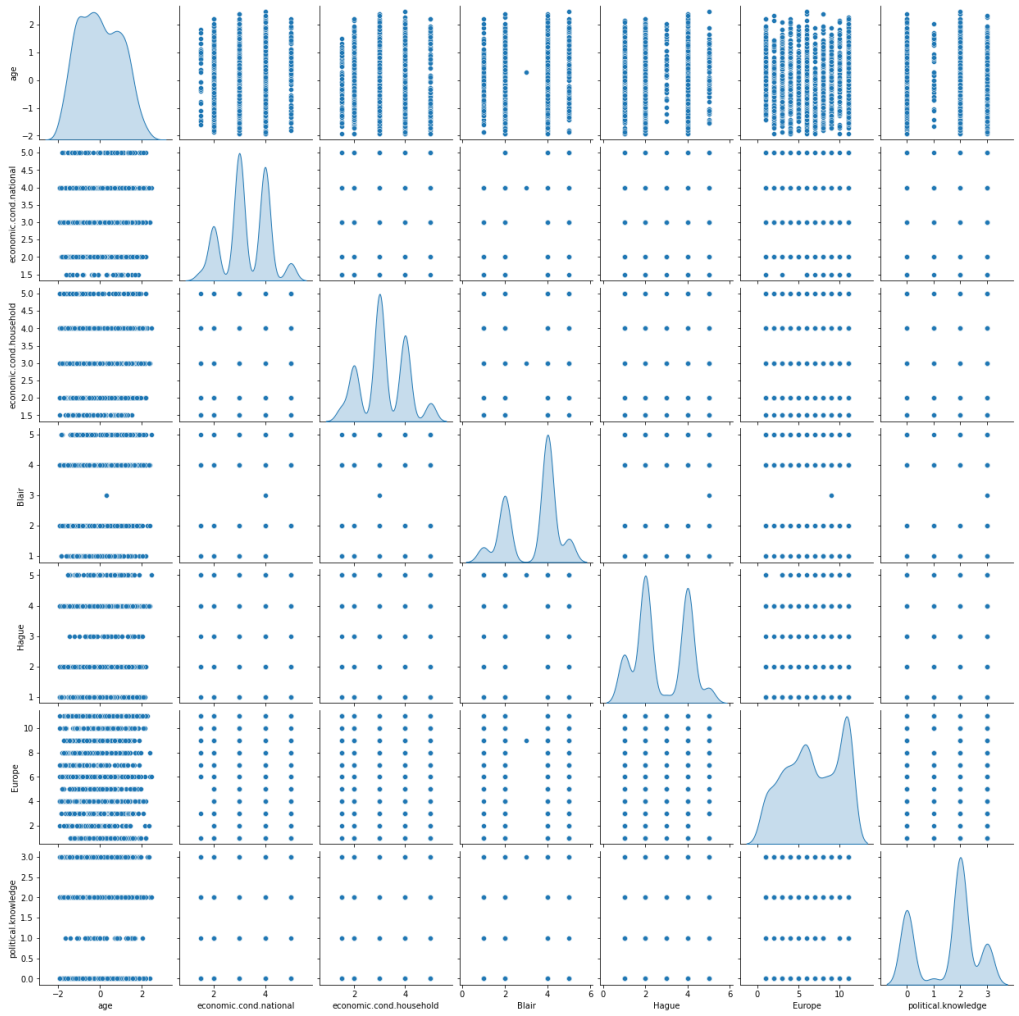

Figure 13: PS:1: Pair plot

Inference

- On performing Bivariate analysis on the column's ‘vote’ and ‘age, we can see that Younger people have less probability of voting Conservative. This pattern is clearly visible, however probability of voting conservative is low even for old age people, as per the above plot
- Majority of the population has a moderate understanding of the political situation. However, the middle-aged (35-50) population seem to have a better understanding than the others.
- The population of both middle-aged male and female is more than the other ages.
- None of variables are highly correlated with each other

- Ratings of 0, 2 & 3 on Knowledge of parties' positions on European integration has not been influenced by different age groups.
- The Eurosceptic sentiments have spread across the complete spectrum of age groups.
- Participants Eurosceptic sentiment has not influenced their assesments on national and household economic conditions
- National and household economic condition have a weak positive correlation
- Voters who rate national economic condition as high has a weak tendency to favour Labour party.
- Voters who are Eurosceptic weakly tend to favour Conservative party.
- Voters who are not Eurosceptic weakly tend to favour Labour party.

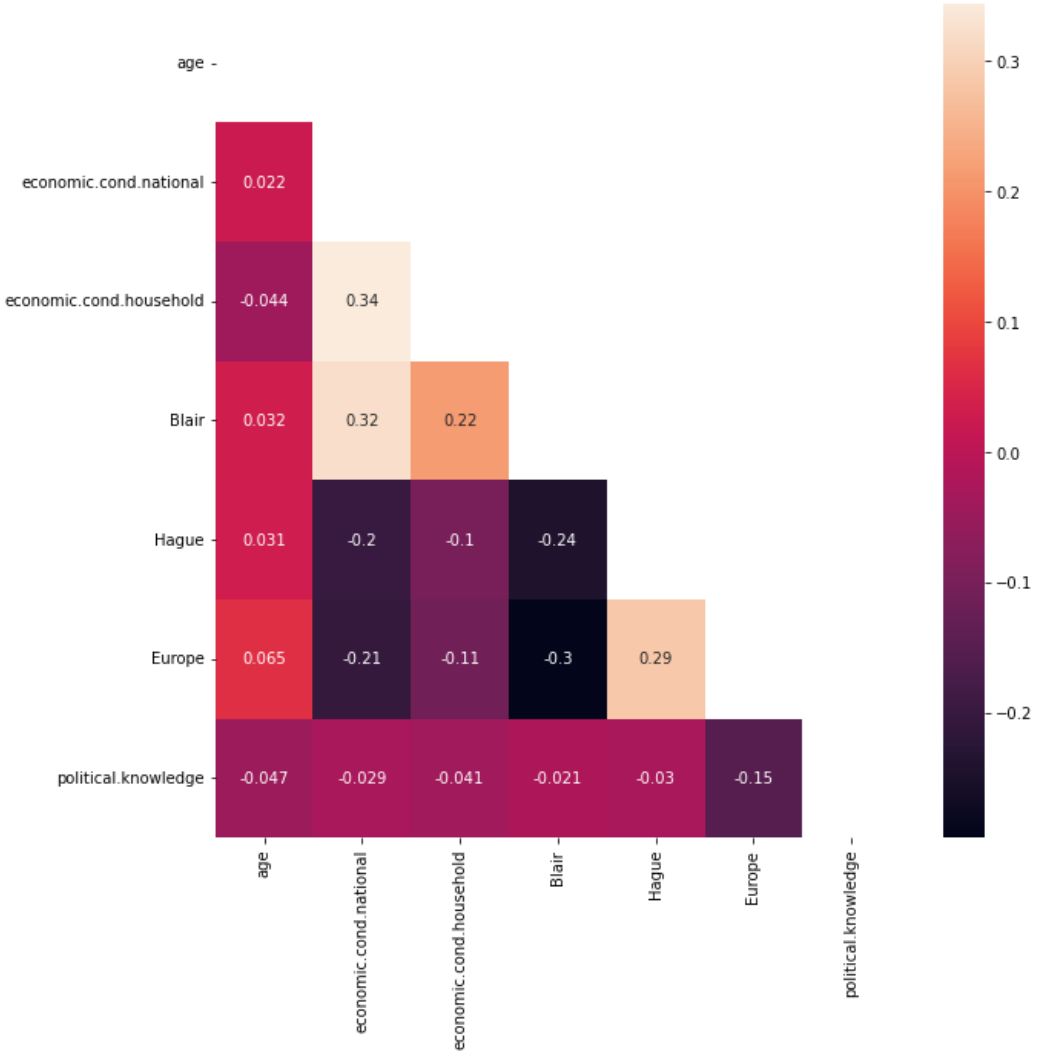

Figure 14: PS:1: Heatmap

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

Encoding String Values

- We use get_dummies () function to encode the string values for modelling, i.e., converting the categorical variables to dummy or indicator variables.
- After converting the variables, the data looks as below:

```
data_en= pd.get_dummies(data_df, columns=['vote','gender'],drop_first=True)
```

Figure 15: PS:1: Encoding

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	vote_Labour	gender_male
0	-0.716161	3.0	3.0	4	1	2	2	1	0
1	-1.162118	4.0	4.0	4	4	5	2	1	1
2	-1.225827	4.0	4.0	5	2	3	2	1	1
3	-1.926617	4.0	2.0	2	1	4	0	1	0
4	-0.843577	2.0	2.0	1	1	6	2	1	1

Figure 16: PS:1: Data Encoded Dataframe

After encoding the dataset is as below.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1517 entries, 0 to 1524
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   age                     1517 non-null   float64
 1   economic.cond.national  1517 non-null   float64
 2   economic.cond.household 1517 non-null   float64
 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   vote_Labour             1517 non-null   uint8
 8   gender_male             1517 non-null   uint8
dtypes: float64(3), int64(4), uint8(2)
memory usage: 130.1 KB
```

Figure 17: PS:1: Data Info

Test and Train Split

- We split the train and test data as 70% and 30%. We copy all the predictor variable i.e., Price in to X data frame and copy the target into y data frame.

- X frame looks like below.

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	IsMale_or_not
0	-0.716161	3.0	3.0	4	1	2	2	0
1	-1.162118	4.0	4.0	4	4	5	2	1
2	-1.225827	4.0	4.0	5	2	3	2	1
3	-1.926617	4.0	2.0	2	1	4	0	0
4	-0.843577	2.0	2.0	1	1	6	2	1

Figure 18: PS:1: X- Frame

```
X_train (1061, 8)
X_test (456, 8)
y_train (1061, 1)
y_test (456, 1)
```

Figure 19: PS:1: Train and Test Dataset

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

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

1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging), and 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. Final Model: Compare the models and write inference which model is best/optimized.

Note: The question 1.4 – 1.7 are answered below together.

Logistic Regression

The model score seems to be pretty good in both training and testing

Test Data

1. Classification Report

```
Accuracy 0.8289473684210527
              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
```

Figure 20: LR-Classification Report

2. Confusion Matrix

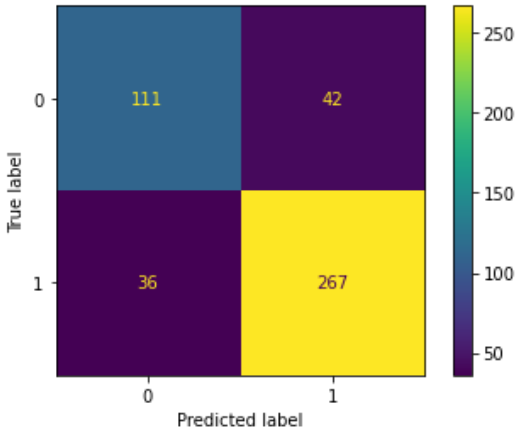

Figure 21: LR-Test Data Confusion Matrix

3. AUC

   The AUC of Test Data is 0.883

4. Accuracy

   Accuracy is 0.828

Train Data

1. Classification Report

Accuracy 0.8341187558906692

	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	0.81	0.78	0.79	1061
weighted avg	0.83	0.83	0.83	1061

Figure 22: LR-Train Data Classification Report

2. Confusion Matrix

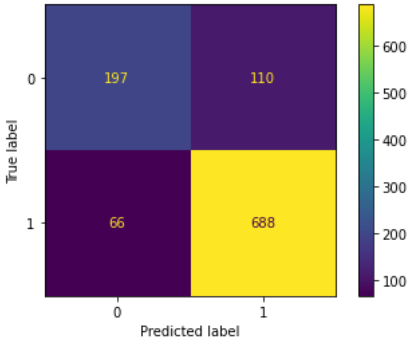

Figure 23: LR-Train Data Confusion Matrix

3. AUC

   the AUC of Train Data 0.890

4. Accuracy

   Accuracy 0.83

Linear Discriminant Analysis

Applying LDA, we see that we get a fairly good model with accuracy of about 83% approximately in the Test data

Test Data

1. Classification Report

```
----------------Test Data--------------
Accuracy 0.831140350877193
              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
```

Figure 24: LDA-Test Data Classification Report

2. Confusion Matrix

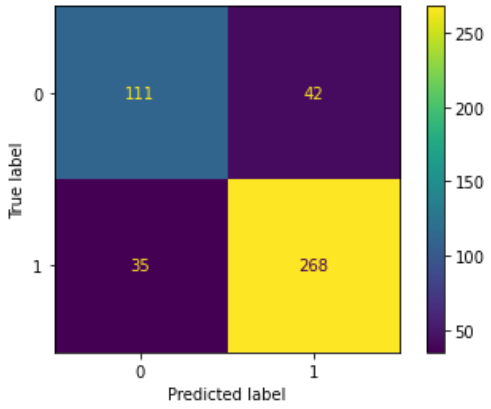

Figure 25: LDA - Test Data Confusion Matrix

3. AUC

   the AUC of Test Data is 0.888

4. Accuracy

   Accuracy 0.83

Train Data

1. Classification Report

```
----------------Train Data--------------
Accuracy 0.8341187558906692
              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
```

Figure 26: LDA - Train Data Classification Report

2. Confusion Matrix

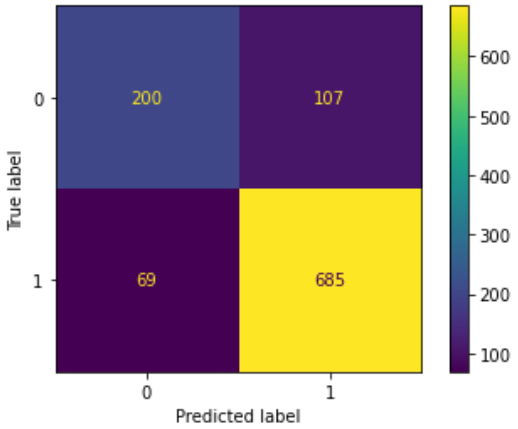

Figure 27: LDA - Train Data Confusion Matrix

3. AUC

   the AUC of Train Data 0.890

4. Accuracy

   Accuracy 0.83

KNN

```
from sklearn.neighbors import KNeighborsClassifier

KNN_model=KNeighborsClassifier(n_neighbors=5)
KNN_model.fit(X_train,y_train)
```

```
KNeighborsClassifier()
```

Figure 28: KNN

Training and Testing results show that the model is excellent with good precision and recall values. This KNN model have good accuracy and recall values

Test Data

1. Classification Report

```
----------------Test Data--------------
Accuracy 0.8223684210526315
              precision    recall  f1-score   support

           0       0.77      0.67      0.72       153
           1       0.84      0.90      0.87       303

    accuracy                           0.82       456
   macro avg       0.81      0.78      0.79       456
weighted avg       0.82      0.82      0.82       456
```

Figure 29: KNN- Test Data Classification Report

2. Confusion Matrix

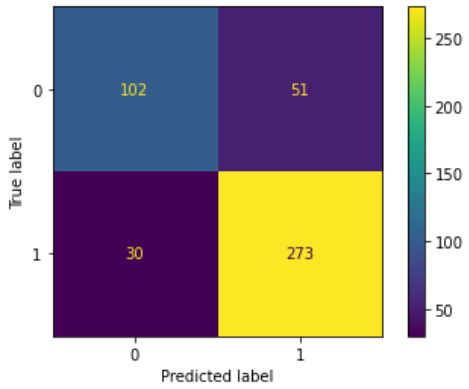

Figure 30: KNN - Test Data Confusion Matrix

3. AUC

   the AUC of Test Data is 0.881

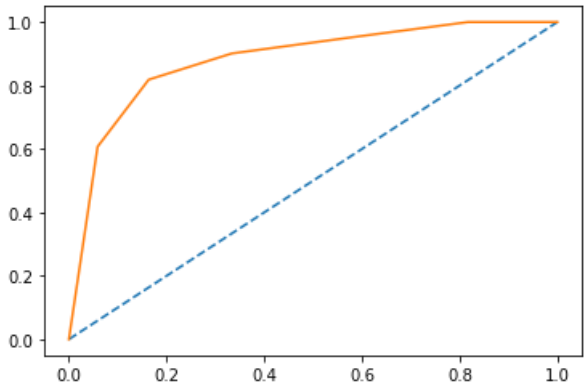

Figure 31: KNN- Test Data AUC Curve

4. Accuracy

   Accuracy 0.82

Train Data

1. Classification Report

```
----------------Train Data--------------
Accuracy 0.8586239396795476
              precision    recall  f1-score   support

           0       0.77      0.72      0.75       307
           1       0.89      0.91      0.90       754

    accuracy                           0.86      1061
   macro avg       0.83      0.82      0.82      1061
weighted avg       0.86      0.86      0.86      1061
```

Figure 32: KNN - Train Data Classification Report

2. Confusion Matrix

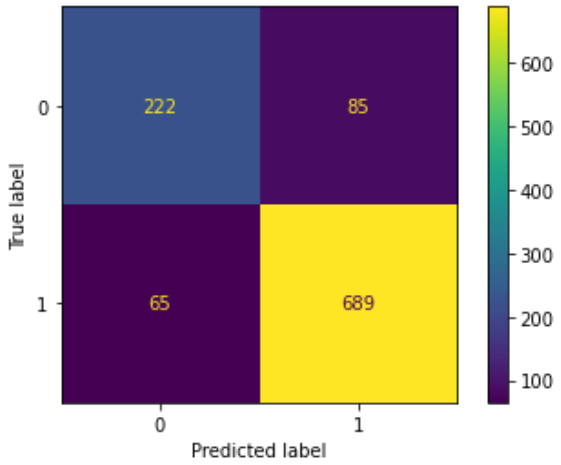

Figure 33: KNN - Train Data Confusion Report

3. AUC

   the AUC of Train Data 0.929

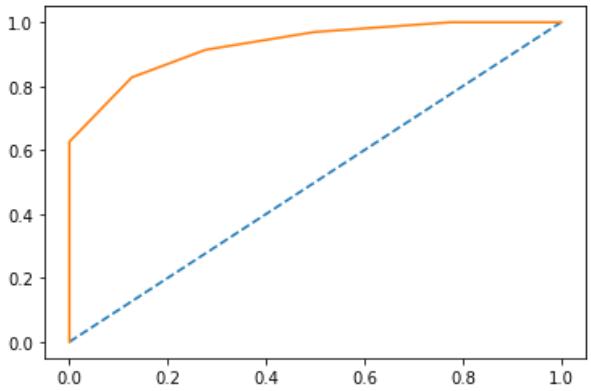

Figure 34: KNN Train Data AUC Curve

4. Accuracy

   Accuracy 0.85

Naïve Bayes Model.

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

Test Data

1. Classification Report

```
----------------Test Data--------------
Accuracy 0.8223684210526315
              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
```

Figure 35: NB Test Data Classification Reports

2. Confusion Matrix

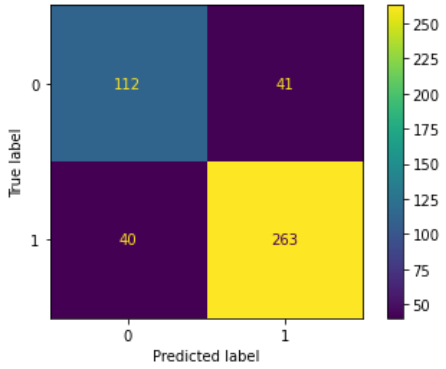

Figure 36: NB Test Data Confusion Matrix

3. AUC

   the AUC of Test Data is 0.876

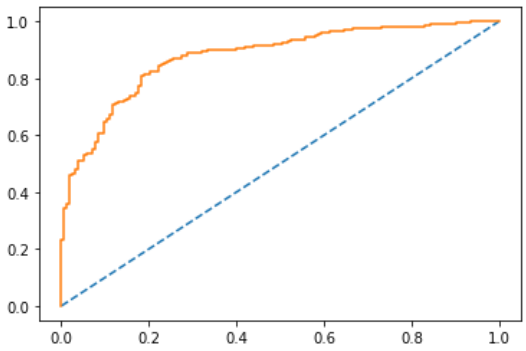

Figure 37: NB Test Data AUC Curve

4. Accuracy

   Accuracy 0.82

Train Data

1. Classification Report

```
----------------Train Data--------------
Accuracy 0.8341187558906692
              precision    recall  f1-score   support

           0       0.72      0.69      0.71       307
           1       0.88      0.89      0.88       754

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

Figure 38: NB Train Data Classification Report

2. Confusion Matrix

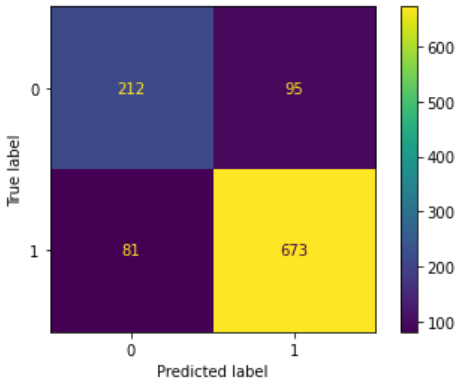

Figure 39: NB Train Data Confusion Matrix

3. AUC

   the AUC of Train Data 0.889

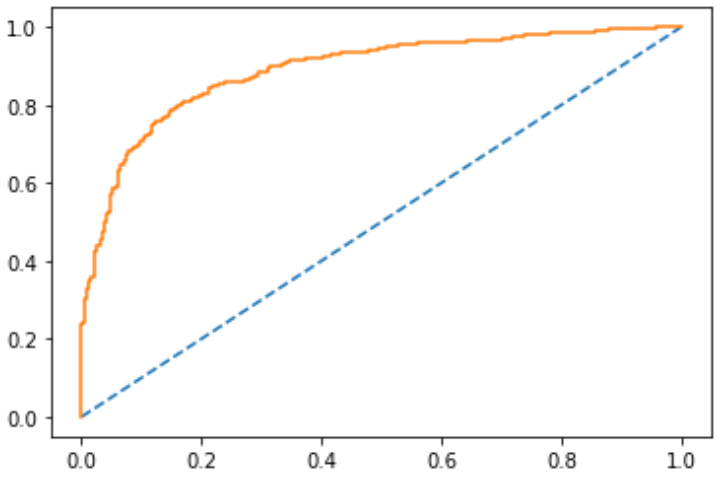

Figure 40: NB Train Data AUC Curve

4. Accuracy

   Accuracy 0.83

Bagging

Test Data

1. Classification Report

```
----------------Test Data--------------
Accuracy 0.8201754385964912
              precision    recall  f1-score   support

           0       0.74      0.71      0.72       153
           1       0.86      0.88      0.87       303

    accuracy                           0.82       456
   macro avg       0.80      0.79      0.80       456
weighted avg       0.82      0.82      0.82       456
```

Figure 41: Bagging Test Data Classification Report

2. Confusion Matrix

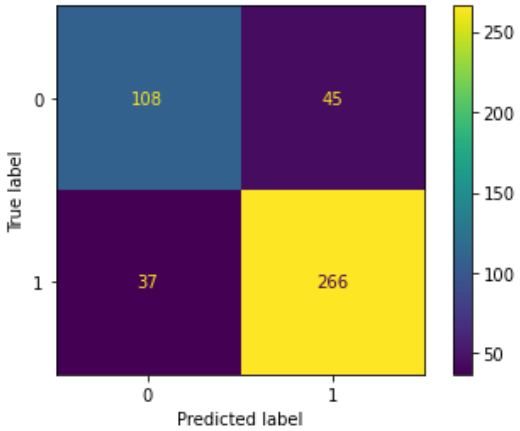

Figure 42: Bagging Test Data Confusion Matrix

3. AUC
   the AUC of Test Data is 0.882

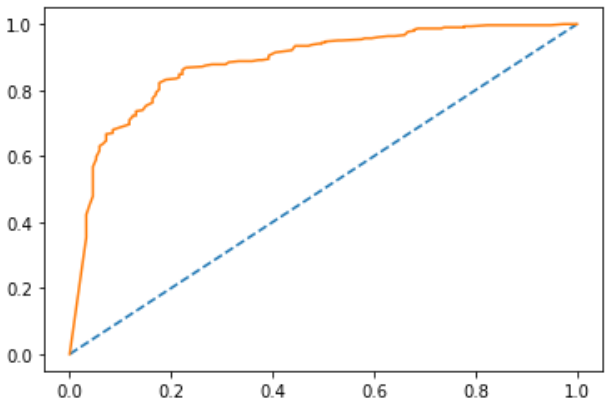

Figure 43: Bagging Test Data AUC Curve

4. Accuracy

   Accuracy 0.82

Train Data

1. Classification Report

```
----------------Train Data--------------
Accuracy 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       307
           1       1.00      1.00      1.00       754

    accuracy                           1.00      1061
   macro avg       1.00      1.00      1.00      1061
weighted avg       1.00      1.00      1.00      1061
```

Figure 44: Bagging Train Data Classification Report

2. Confusion Matrix

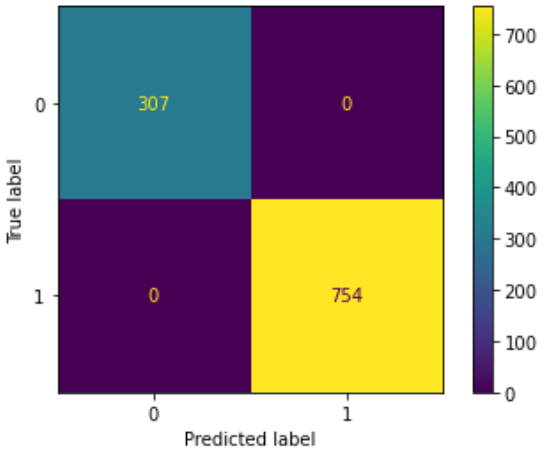

Figure 45: Bagging Train Data Confusion Matrix

3. AUC

   the AUC of Train Data 1.000

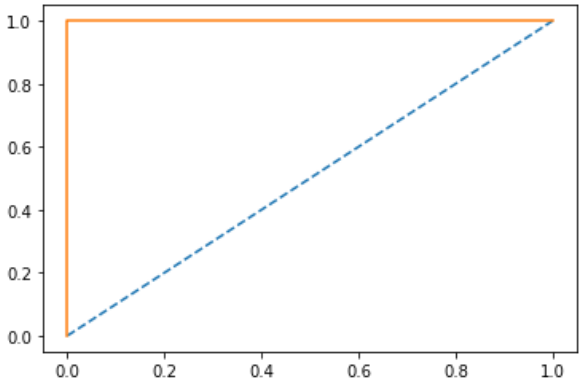

Figure 46: Bagging Train Data AUC Curve

4. Accuracy

   Accuracy 1.

Boosting

Test Data

1. Classification Report

```
----------------Test Data--------------
Accuracy 0.8135964912280702
              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
```

Figure 47: Boosting Test Data Classification Report

2. Confusion Matrix

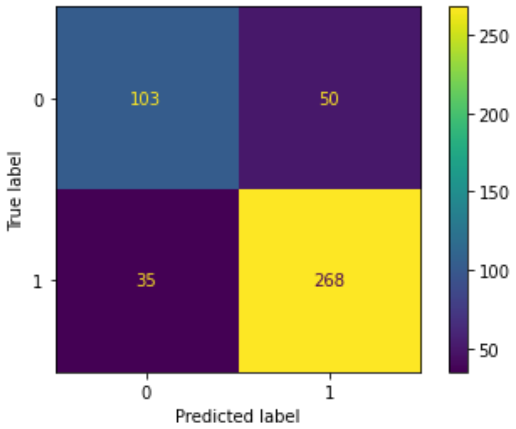

Figure 48: Boosting Test Data Confusion Matrix

3. AUC

   the AUC of Test Data is 0.877

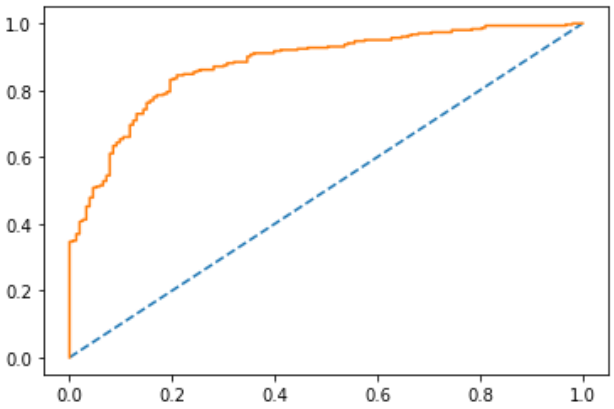

Figure 49: Boosting Test Data AUC Curve

4. Accuracy

   Accuracy 0.81

Train Data

1. Classification Report

```
----------------Train Data--------------
Accuracy 0.8501413760603205
              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
```

Figure 50: Boosting Train Data Classification Report

2. Confusion Matrix

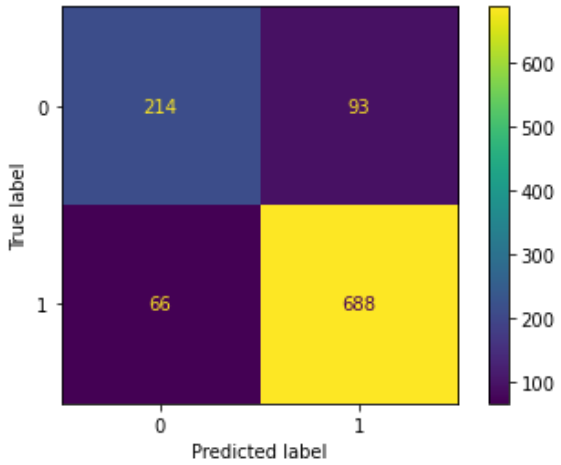

Figure 51: Boosting Train Data Confusion Matrix

3. AUC

   the AUC of Train Data 0.915

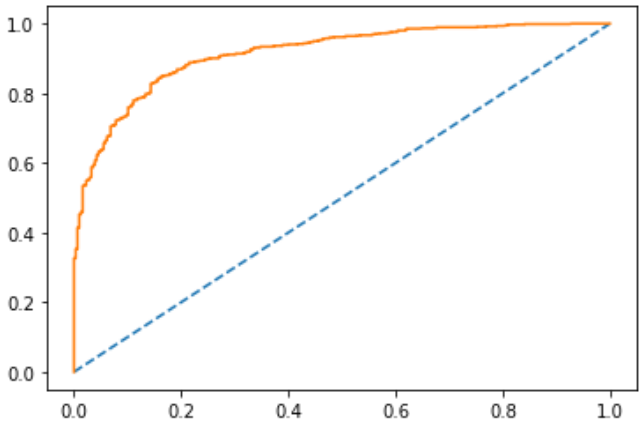

Figure 52: Boosting Train Data AUC Curve

4. Accuracy

   Accuracy 0.85

Gradient Boosting

Test Data

1. Classification Report

```
----------------Test Data--------------
Accuracy 0.8355263157894737
              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
```

Figure 53: Gradient Boosting Test Data Classification Report

2. Confusion Matrix

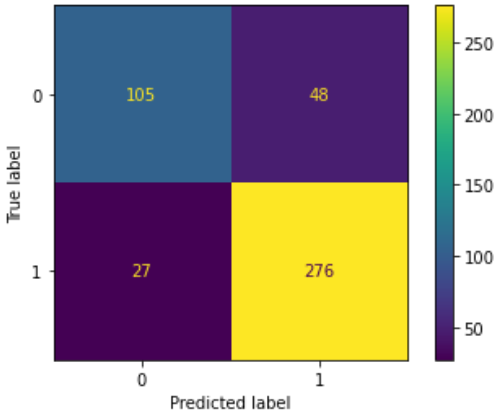

Figure 54: Gradient Boosting Test Data Confusion Matrix

3. AUC

   the AUC of Test Data is 0.899

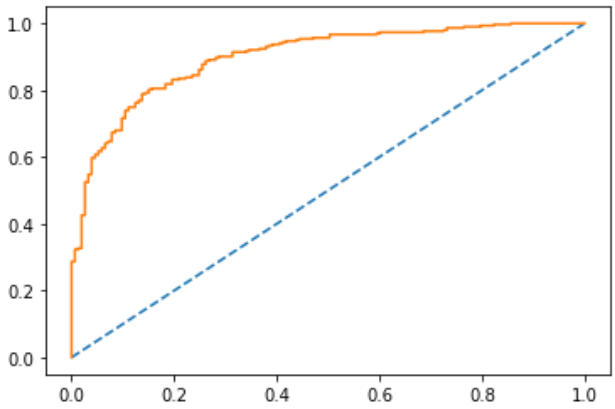

Figure 55: Gradient Boosting Test Data AUC Curve

4. Accuracy

   Accuracy 0.83

Train Data

1. Classification Report

```
----------------Train Data--------------
Accuracy 0.8925541941564562
              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
```

Figure 56: Gradient Boosting Train Data Classification Report

2. Confusion Matrix

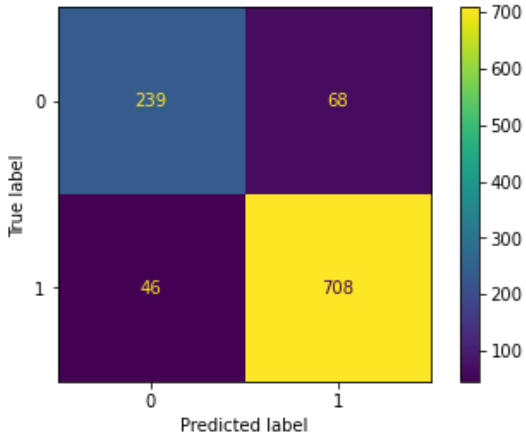

Figure 57: Gradient Boosting Train Data Confusion Matrix

3. AUC

   the AUC of Train Data 0.951

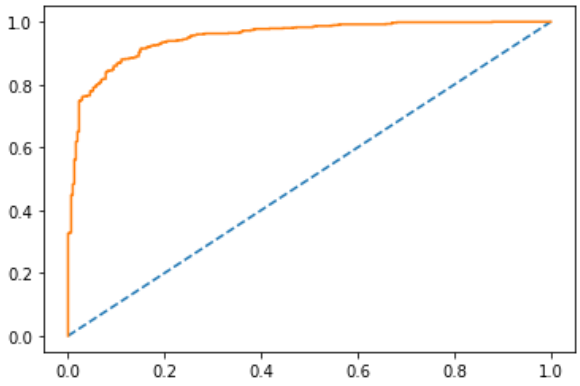

Figure 58: Gradient Boosting Train Data AUC Curve

4. Accuracy

   Accuracy 0.89

Comparison of Models:

Table 1: Comparison of Models (1)

	Logistic Regression		LDA		KNN	
	Train Set	Test Set	Train Set	Test Set	Train Set	Test Set
Accuracy	0.83	0.82	0.83	0.83	0.858	0.82
AUC	0.89	0.88	0.89	0.888	0.929	0.88
Recall	0.91	0.88	0.91	0.88	0.91	0.9
Precision	0.86	0.86	0.86	0.86	0.89	0.84
F-1 Score	0.89	0.87	0.89	0.87	0.9	0.87

Table 2: Comparison of Models (2)

	Naives Bayes		Bagging		Boosting		Gradient Boosting	
	Train Set	Test Set	Train Set	Test Set	Train Set	Test Set	Train Set	Test Set
Accuracy	0.83	0.82	1	0.82	0.85	0.81	0.89	0.83
AUC	0.889	0.876	1	0.88	0.91	0.87	0.95	0.89
Recall	0.89	0.87	1	0.88	0.91	0.88	0.94	0.91
Precision	0.88	0.87	1	0.86	0.88	0.84	0.91	0.85
F-1 Score	0.88	0.87	1	0.87	0.9	0.86	0.93	0.88

1.8 Based on these predictions, what are the insights?

- Accuracy is almost similar – in between 80 – 83%
- Recall for train set is same for LDA than logistic Regression, on the test set its performance is good.
- Accuracy & prediction is good. Hence both the model's performance is approx. equal.
- Either of the two models, Logistic Regression or LDA can be used to make predictions on the exit poll as to whether a particular voter would vote the Conservative or the Labour party based on the information provided. .
- The accuracy of KNN model is very good compared to other models on train set.
- Naive Bayes model better for both train and test datasets.
- These Models should perform even better
- Even after applying Model Tuning, bagging & boasting did not improve performance
- of models.
- Majority of the population is between the ages 35-60 with considerable political knowledge and would vote mostly for Labour party.
- Bagging model is not very suitable and has overfitting issues

2 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:

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

We are importing the necessary libraries for the data set analysis and then the necessary data sets from the cloud.

Number of Characters in each file

- Number of characters in Roosevelt file: 7571
- Number of characters in Kennedy file: 7618
- Number of characters in Nixon file: 9991

No of Words in each text file

- Number of words in Kennedy file: 1390
- Number of words in Nixon file: 1819
- Number of words in Roosevelt file: 1819

Number of Sentences in each text File

- Number of Sentences in Kennedy file: 52
- Number of Sentences in Nixon file: 68
- Number of Sentences in Roosevelt file: 67

2.2 Remove all the stopwords from all three speeches.

Removing stop words is important in language processing as it removes words without any meaning. Refer the python file for the code snippet.

After removing the stopwords from the file, the output looks like

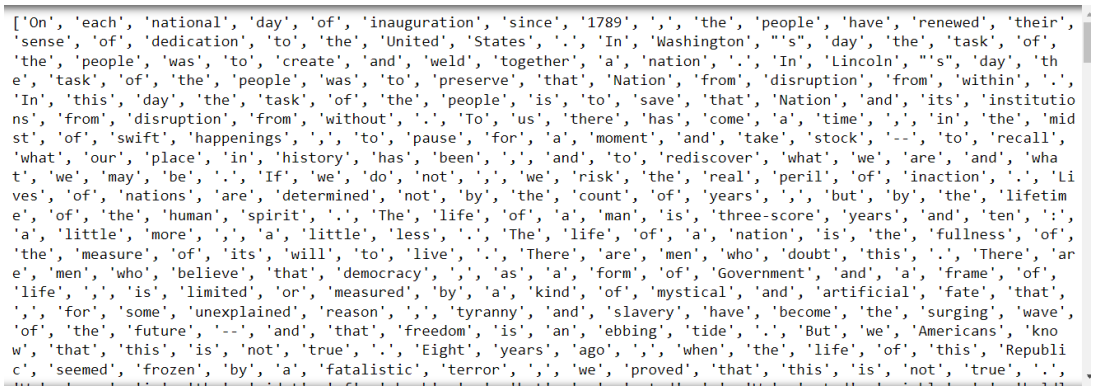

Figure 59: Roosevelt File

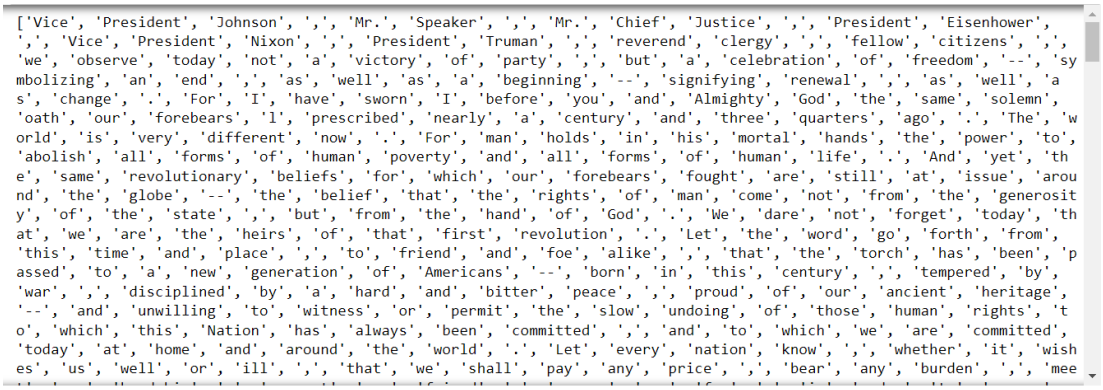

Figure 60: Kennedy File

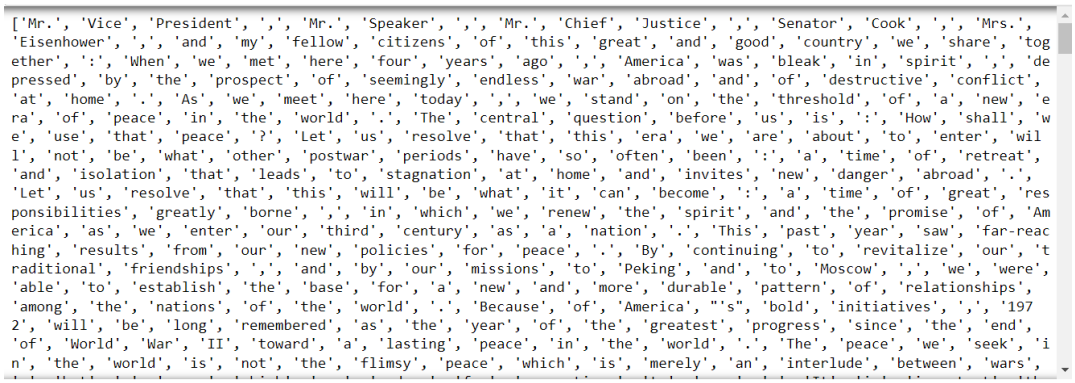

Figure 61: Nixon File

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)

The words having highest frequency in each file are found out. Refer python file for the snippet.

Roosevelt:

```
[('Nation', 12),
 ('Know', 10),
 ('Spirit', 9),
 ('Life', 9),
 ('Democracy', 9),
```

Nation is the word occurring 12(highest) number of times.

Kennedy:

```
[('Let', 16),
 ('Us', 12),
 ('World', 8),
 ('Sides', 8),
```

Let is the word occurring 16(highest) number of times.

Nixon:

```
[('Us', 26),
 ('Let', 22),
 ('America', 21),
 ('Peace', 19),
 ('World', 18),
```

Us is the word occurring 26(highest) number of times.

2.4 Plot the word cloud of each of the speeches of the variable.

Generating the word cloud for all three of them

Roosevelt:

Figure 62: WordCloud for Roosevelt File

Kennedy

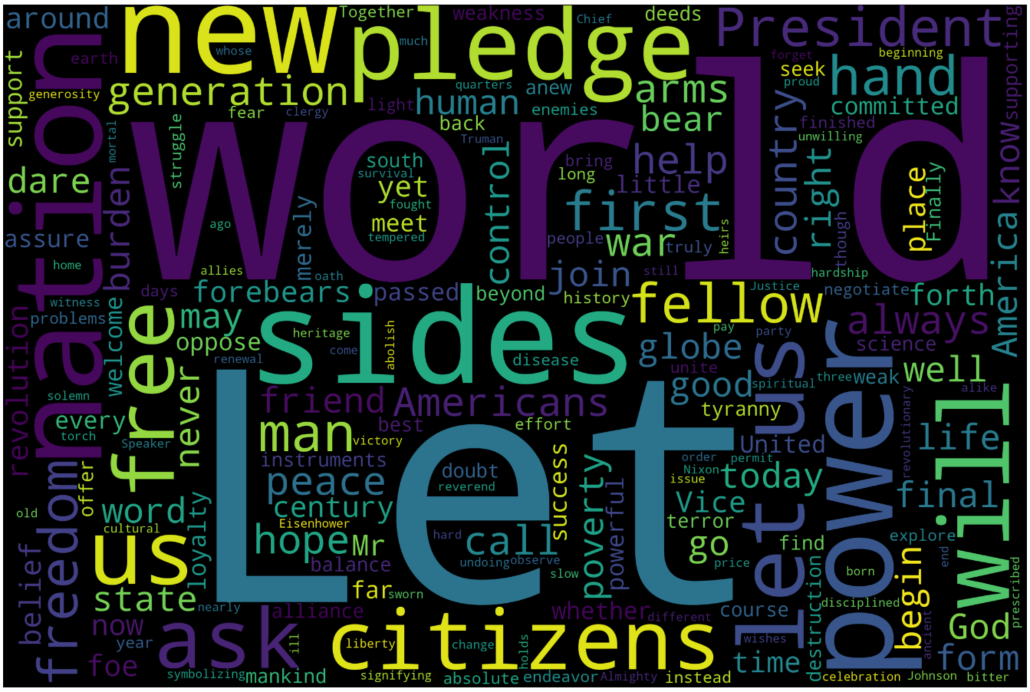

Figure 63: WordCloud for Kennedy File

Nixon

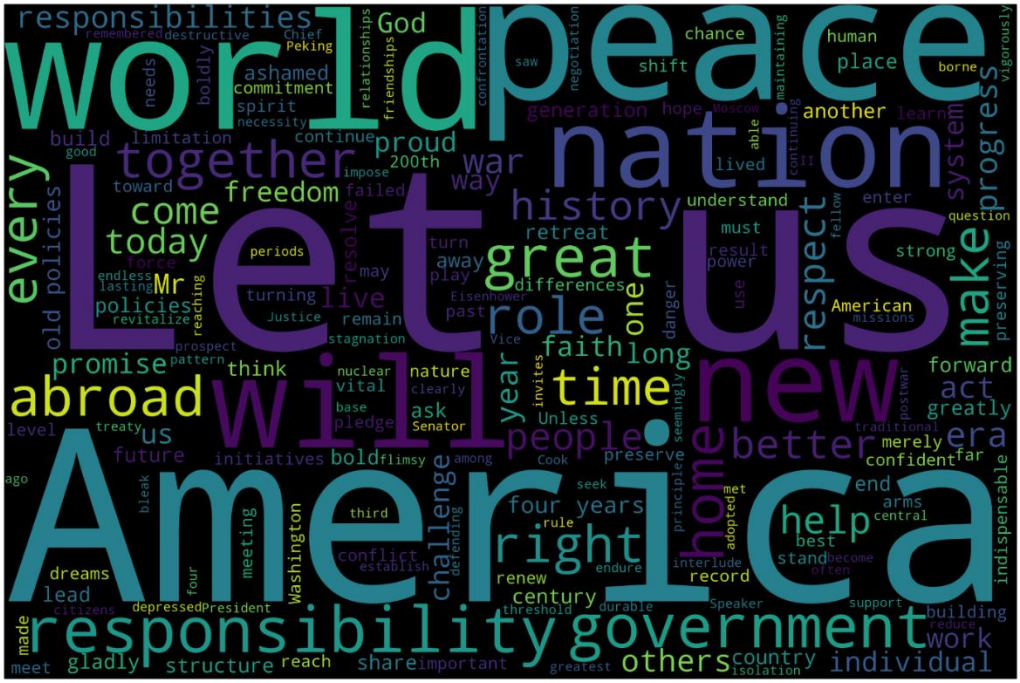

Figure 64: WordCloud for Nixon File

The End!