PROJECT 6
MACHINE LEARNING

Table of Contents

1 Problem 1 1

1.1 Define the problem and perform Exploratory Data Analysis 1

1.2 Data Pre-processing 4

1.3 Model Building 4

1.4 Model Performance Evaluation 5

1.5 Model Performance Improvement 10

1.6 Final Model Selection 11

1.7 Actionable Insights and Recommendations 11

2 Problem 2 11

2.1 Define the Problem and Perform Exploratory Data Analysis 11

2.2 Text Cleaning 12

2.3 Plot word cloud of all three speeches 12

1 Problem 1

In this problem, a predictive model will be built to predict the political party voter's support with the help of machine learning algorithms on Python Programming.

1.1 Define the problem and perform Exploratory Data Analysis

Problem Definition

It is not easy to predict the political party that is likely to win. Because, it includes various aspects for a party to win such as it's set agendas for the election, the party's popularity among the people, the maximum number of voters liking the work of the party and so on. The given dataset has the following variables:

- Vote: This variable indicates party choice, such as labour or conservative.
- Gender: This variable indicates female or male
- age: this variable indicates parties age in years.
- economic condition national: This variable indicates current national economic conditions.
- economic condition of households: This variable indicates current household economic conditions.
- Europe: This variable indicates respondent’s attitudes.
- Blair: This variable indicates the labour leader.
- Hague: This variable indicates a conservative leader.
- Political Knowledge: This variable indicates knowledge of party positions.

Exploratory Data Analysis

To begin with, explore the dataset. Firstly, import the required packages and import the dataset, as demonstrated below.

```
2 df.head()
```

	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

The information about the given dataset is demonstrated below.

```
1 df.info()
```

```
<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
```

Dataset shape is presented below.

```
1 df.shape
```

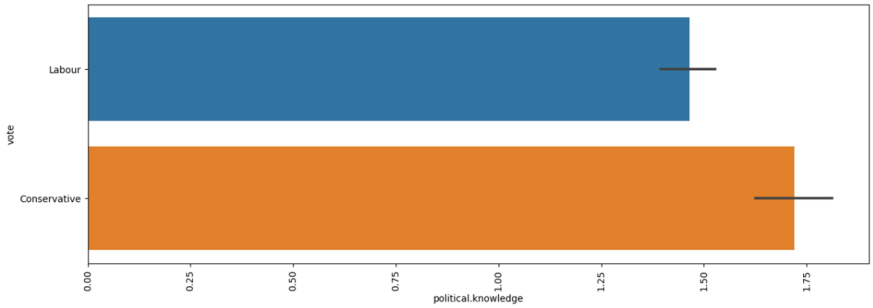
```
(1525, 9)
```

According to the above result, the given dataset consists of 1525 rows and 9 columns. The statistical information of the given dataset is shown below.

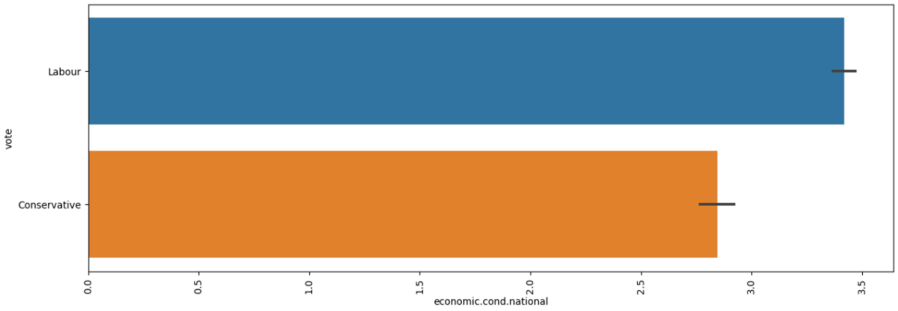
```
1 df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
age	1525.0	54.182295	15.711209	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	3.245902	0.880969	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	3.140328	0.929951	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	3.334426	1.174824	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	2.746885	1.230703	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	6.728525	3.297538	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	1.542295	1.083315	0.0	0.0	2.0	2.0	3.0

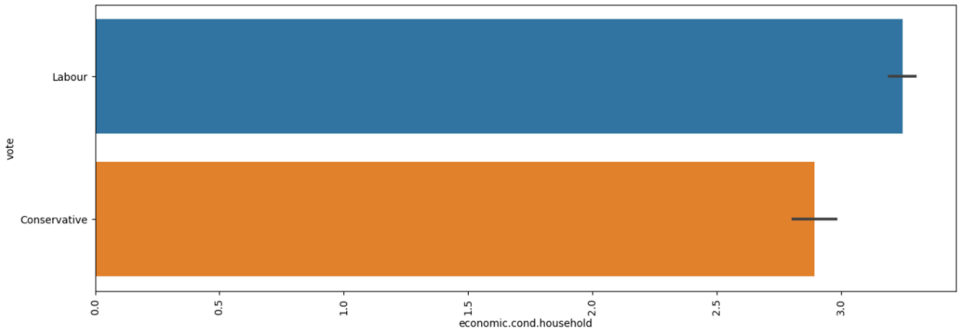
Based on the above result, the minimum age of the parties is 24, and the maximum age is 93. The average age of the parties is 54. The minimum current national economic condition is 1, and the maximum is 5. The minimum current household economic conditions is 1, and the maximum is 5. The minimum political knowledge is 0, the maximum is 3, and the average political knowledge is 3. Next, perform a univariate analysis to visualize the political knowledge of parties based on the votes, as demonstrated below.



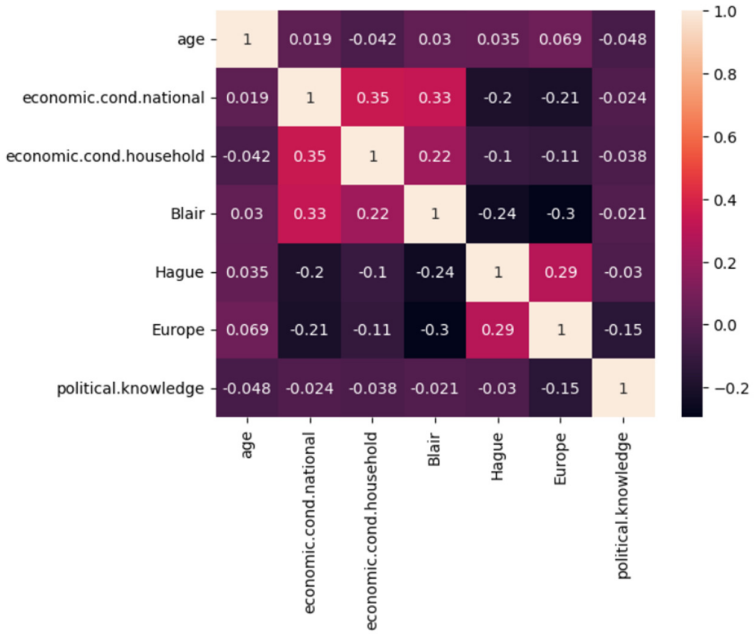
As per the above visualization, the highest number of parties have political knowledge on conservative party when compared with the labour party. Afterwards, the current national economic conditions of the parties are visualized based on the votes, as presented below.



As per the above visualization, the labour party has the highest number of national economic conditions when compared with the conservative party. Afterwards, the current household economic conditions of the parties are visualised based on the votes, as presented below.



As per the above visualization, the labour party has the highest number of national economic conditions compared with the conservative party. Next, a multivariate analysis is performed to determine the relationship between the variables. The result of the correlation matrix for the given dataset is demonstrated below.



According to the correlation result, economic household conditions, economic national conditions, Blair have positive relationship with each other. And, Hauge and Europe variable share a positive relationship with each other.

1.2 Data Pre-processing

In this section, data is prepared for building a predictive model that includes outlier detection, data encoding, data scaling and data splitting. This dataset does not have any missing values and outliers. Later, change the target variable conservative as 1, and labour as 0. Next, split the dataset into train and test dataset. After that, continue scaling the numerical features. Next comes encoding the categorical features.

1.3 Model Building

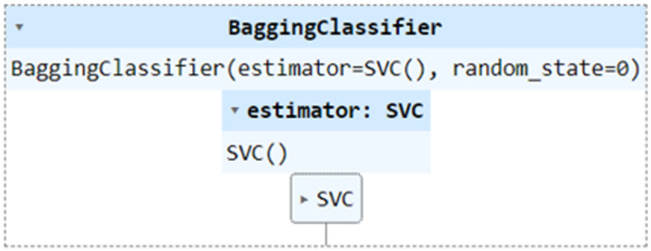
A predictive model is built to predict the political party that is likely to be supported. The selected predictive models are KNN, Naïve Bayes, Bagging classifier, and Boosting models. The K nearest neighbour classifier model for this prediction is demonstrated below.

```
KNeighborsClassifier
KNeighborsClassifier()
```

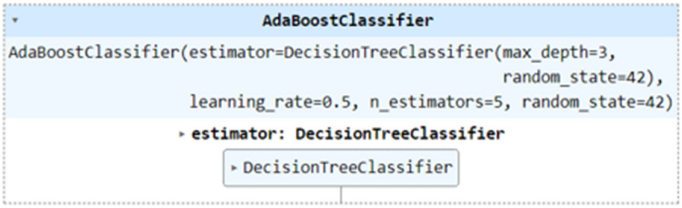
Next, a naïve bayes classifier is built as demonstrated below,

```
GaussianNB
GaussianNB()
```

Afterwards, a bagging classifier is built with SVM estimator as demonstrated below.



Next, an Ada boost classifier model is built as demonstrated below.



All the model metrices are determined and explained in the model performance section.

1.4 Model Performance Evaluation

In this section, evaluate the created model performance based on the evaluation metrices such as accuracy, confusion matrix, and ROC AUC score.

Justification of Metrics

A confusion matrix is one of the performance evaluation tools in machine learning, that represents the accuracy of a classification model.

Accuracy is a metric that helps to measure the correctness of a model. Moreover, it is easy to determine the correctness. There are also other evaluation metrics that can be used such as AUC. ROC (Receiver Operating Characteristic) curve helps in plotting TP (True Positive) rate versus the FP (False Positive) rate at various classification thresholds.

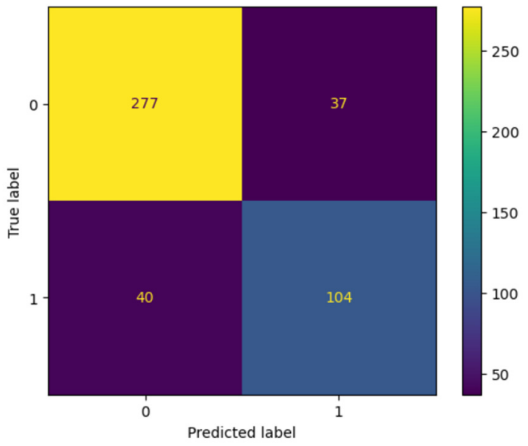
The thresholds can be referred as the probability cutoffs, which separates two classes in the binary classification. It utilizes probability for showing how effectively the model separates classes.

KNN Model Performance

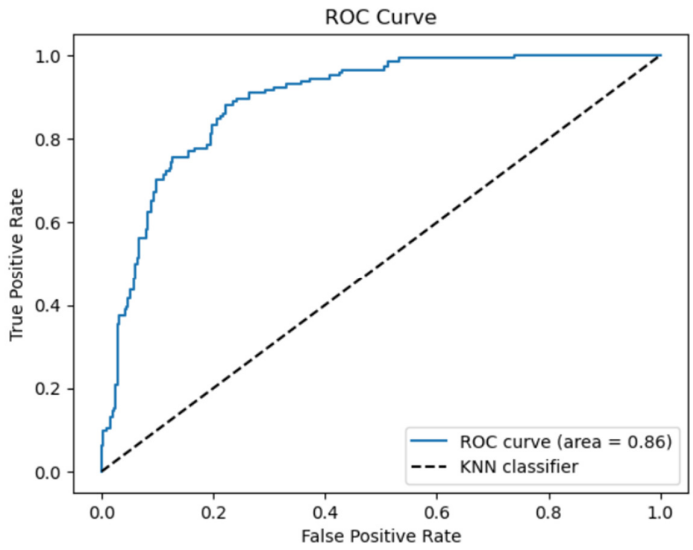
Accuracy

```
Accuracy Score of KNN Classifier is : 0.8318777292576419
```

Confusion Matrix



ROC AUC Curve



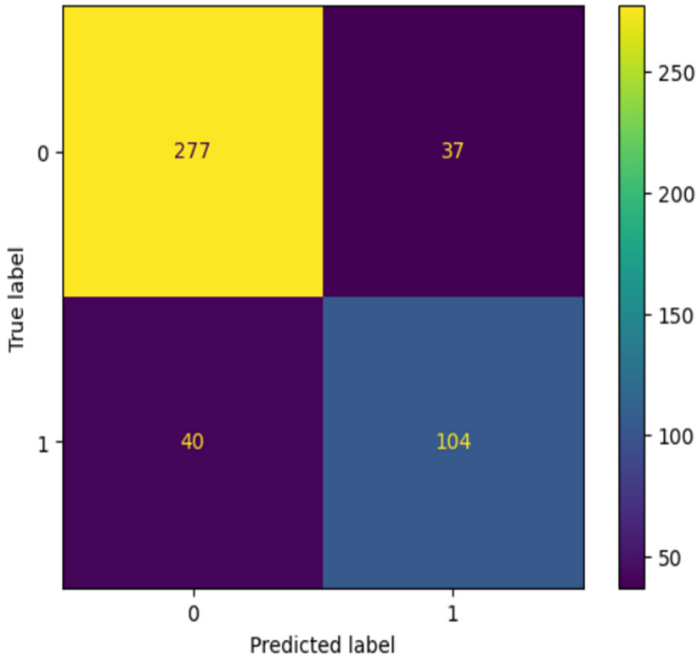
According to the above KNN model performance, it has 83.18% of accuracy for predicting the political party likely to be supported. Based on the confusion matrix, it predicts that 277 parties are interested to support the conservative party and 104 parties are interested to support the labour party. And, the ROC AUC value is 0.86, which indicates that the created model is able to predict the two classes such as political parties.

Naïve Bayes Model Performance

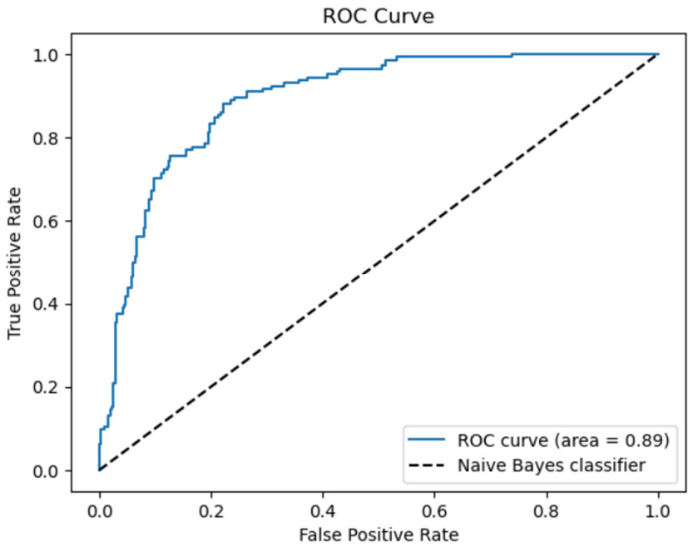
Accuracy

```
Accuracy Score of Naive Bayes Classifier is : 0.8318777292576419
```

Confusion Matrix



ROC AUC Curve



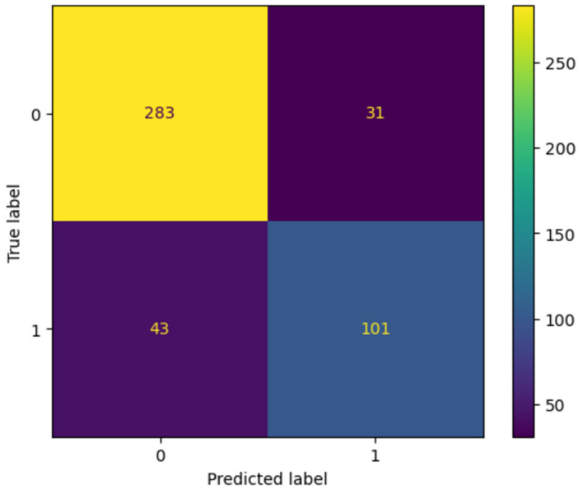
According to the above naïve bayes model's performance, it has 83.18% of accuracy for predicting the political party likely to be supported. Based on the confusion matrix, it predicts that 277 parties are interested to support the conservative party and 104 parties are interested to support the labour party. And, the ROC AUC value is 0.89, which indicates that the created model is able to predict the two classes such as political parties.

Bagging Model Performance

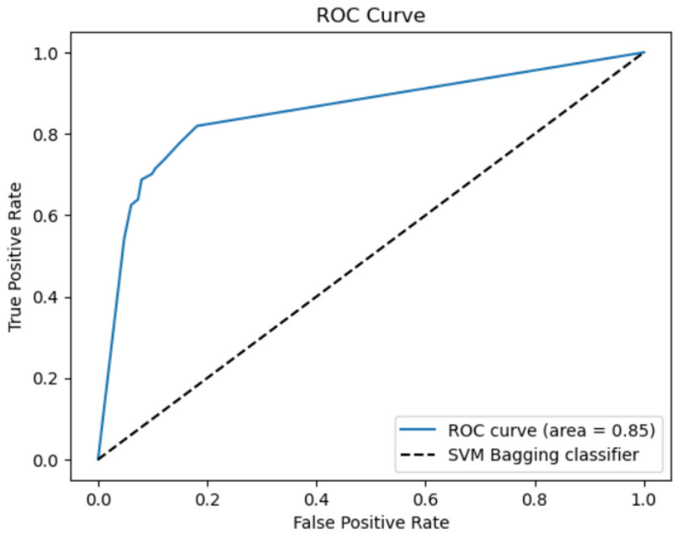
Accuracy

```
Accuracy Score of SVM is : 0.8384279475982532
```

Confusion Matrix



ROC AUC Curve



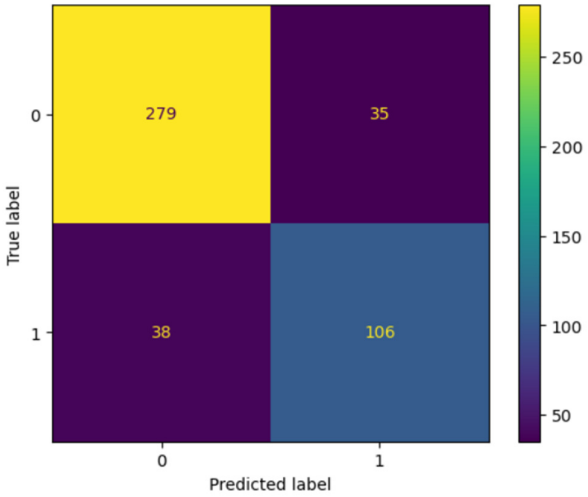
According to the above Bagging model's performance, it has 83.84% of accuracy for predicting the political party likely to be supported. Based on the confusion matrix, it predicts that 283 parties are interested to support the conservative party and 101 parties are interested to support labour party. And, the ROC AUC value is 0.85, which indicates that created model is able to predict the two classes such as political parties.

Adaboost Classifier Performance

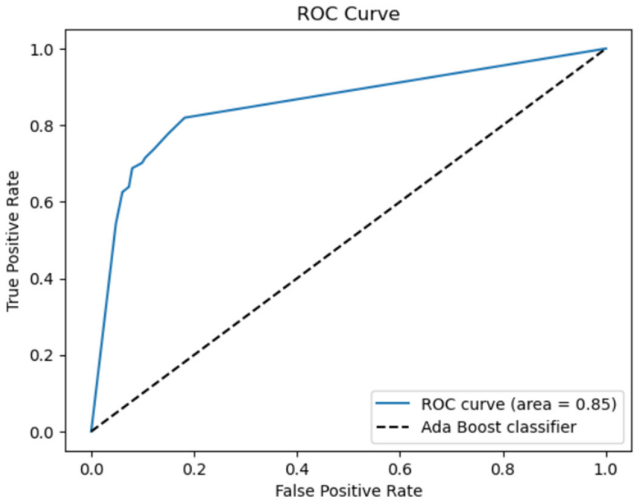
Accuracy

```
Accuracy Score of Adaboost is : 0.8406113537117904
```

Confusion Matrix



ROC AUC Curve



According to the above Ada boosting model's performance, it has 84.06% of accuracy for predicting the political party likely to be supported. Based on the confusion matrix, it predicts that 279 parties are interested to support the conservative party and 106 parties are interested to support the labour party. And, the ROC AUC value is 0.85, which indicates that created model is ability to predicting the two classes such as political parties.

1.5 Model Performance Improvement

In this section, the bagging and boosting models will be improved.

Bagging Classifier with Hyper Parameters

The hyper parameter for bagging classifier is as follows:

- C: 1,10,100,1000
- Gamma: 1,0.1, 0.001, 0.0001
- Kernel: rbf and linear

The hyper tunned bagging classifier accuracy score is demonstrated below.

```
 Improved Accuracy Score of SVM bagging is : 0.8384279475982532
Classification Report :
              precision    recall  f1-score   support

           0       0.86      0.91      0.89       314
           1       0.77      0.69      0.73       144

    accuracy                           0.84       458
   macro avg       0.82      0.80      0.81       458
weighted avg       0.84      0.84      0.84       458
```

Boosting Classifier with Hyper Parameters

The hyper parameter for bagging classifier is as follows:

- Number of estimators: 10,100,200,250
- Algorithm: SAMME, SAMME.R
- Learning rate: 0.05, 0.5, 1.5, 205

The hyper tunned bagging classifier accuracy score for the boosting classifier model is demonstrated below.

```
 Improved Accuracy Score of Adaboosting Model is : 0.8449781659388647
Classification Report :
              precision    recall  f1-score   support

           0       0.89      0.88      0.89       314
           1       0.74      0.77      0.76       144

    accuracy                           0.84       458
   macro avg       0.82      0.82      0.82       458
weighted avg       0.85      0.84      0.85       458
```

1.6 Final Model Selection

According to all the models, the hyper tunned Adaboost model is the best model for predicting the political party, because it has a high accuracy that is 84.49%, which is high accuracy when compared with the other models. It predicted that 279 parties are interested to support the conservative party and 106 parties are interested to support the labour party. And, the ROC AUC value is 0.85, which indicates that the created model is able to predict the two classes such as political parties. Therefore, it is stated that the Adaboost model is the best model for this prediction.

1.7 Actionable Insights and Recommendations

It is recommended that the labour party must work on its ideologies, improve their party's position in the society, and work on the betterment of the part by contributing to the different sects of the society like the upper and lower classes. This helps the party to increase its supporters.

2 Problem 2

Based on this problem, text analysis will be conducted on the provided president's speeches with the help of Python programming language.

2.1 Define the Problem and Perform Exploratory Data Analysis

Problem Definition

The problem is that, it is not easy to find the unique words from the speeches of the USA's presidents, or determine the number of times a word is repeated in the inaugural address.

Exploratory Data Analysis

To begin with, explore the president speeches dataset by importing the required packages and the dataset, as demonstrated below.

```
2 df1.head()
```

	Name	Speech
0	Roosevelt	On each national day of inauguration since 178...
1	Kennedy	Vice President Johnson, Mr. Speaker, Mr. Chief...
2	Nixon	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...

Next, determine the number of characters, words and sentences. As per the result, Number of characters is 204957, number of sentence count is 9 and number of words is 41871.

2.2 Text Cleaning

Next, text cleaning is performed that includes stop word removal, stemming, and find the three most common words used in all the three speeches. And, the determined most common words.

```
from collections import Counter
split_it = df1["Speech"].sum()
Counter = Counter(split_it)
# most_common() produces k frequently encountered
most_occur = Counter.most_common(3)
most_occur
```

```
[(' ', 7458), ('e', 4311), ('n', 3093)]
```

2.3 Plot word cloud of all three speeches

Next, the word count is plotted for all the three speeches as demonstrated below.