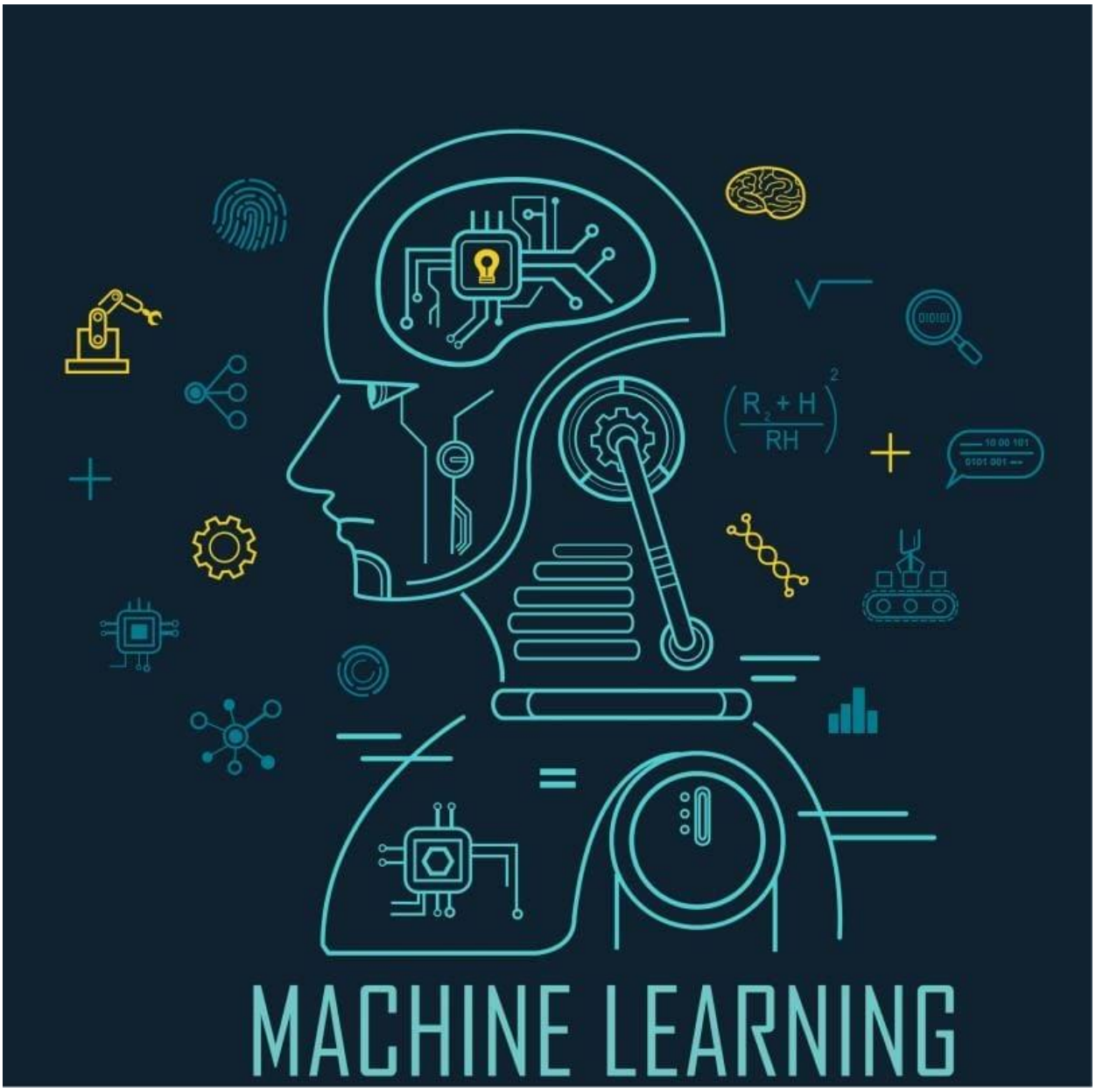
MACHINE LEARNING – CODED PROJECT



By

R. SUKANYA

CONTENT

Sl. No	TITLE	PAGE NO
	List of Figures	4
	List of Tables	5
	Scoring Rubrics	6
	Data Description	7
1.	**Problem Statement 1**	**8**
1.1	**Define the problem and perform exploratory Data Analysis**	8
1.1.1	Import the libraries	8
1.1.2	Reading and loading the dataset, Election_Data	8
1.1.3	Checking the shape of the dataset	8
1.1.4	Checking the datatypes of all the columns	9
1.1.5	Check the statistical summary	9
1.1.6	Univariate Analysis	10
1.1.6.1	Numerical columns	11
1.1.6.2	Categorical columns	12
1.1.7	Multivariate Analysis: Visualizations to identify patterns and insights	13
1.1.7.1	Categorical vs Numerical	13
1.1.7.2	Numerical vs Numerical	15
1.1.7.3	Correlation Plot	16
1.1.8	Key individual observations on individual variables and relationship between variables.	16
1.2	**Data Pre-processing – Prepare the data for modelling**	**17**
1.2.1	Prepare the data for modelling	17
	Missing value treatment	17
	Check for duplicates	18
	Outlier Detection and Treatment	18
1.2.2	Encoding the data	19
1.2.3	Split the data	20
1.2.4	Scaling the data	21
1.2.5	Train-test Split	21
1.3	**Model Building**	**21**
1.3.1	Model Building – (Naïve Bayes, KNN, Bagging and Boosting)	21
1.3.2	Metrics of Choice (Justify the Evaluation Metrics)	22

1.4	**Model Performance Evaluation**	**23**
1.4.1	Check the confusion matrix and classification metrics for all the models (for both train and test dataset)	23
1.4.2	ROC-AUC score and plot the curve:	24
1.4.3	Comment on all the model performances	25
1.5	**Model Performance Improvement**	**25**
1.5.1	Improve the model performance of bagging and boosting models by tuning the model	25
1.5.2	Comment on the model performance improvement on training and test data	27
1.6	**Final Model Selection**	**27**
1.6.1	Compare all the models built so far.	27
1.6.2	Select the final model with proper justification	28
1.6.3	Check the most important features in the final model and draw inferences.	29
1.7	**Actionable Insights and Recommendations:**	**30**
1.7.1	Compare all the models – Conclude with the key takeaways for business.	30
2.0	**Problem 2: Problem Statement**	**31**
2.1	Problem Definition - Define the problem and Perform Exploratory Data Analysis (EDA)	31
2.1.1	Importing Libraries	31
2.1.2	Importing the speeches	31
2.1.3	Find the number of Character, words & sentences in all three speeches	31
2.2	**Text Cleaning**	**31**
2.2.1	Stopword removal	33
2.2.2	Stemming	34
2.3	**Plot Word cloud of all three speeches**	**36**
2.3.1	Show the most common words used in all three speeches in the form of word clouds.	36

LIST OF FIGURES

Sl.No.	TITLE	Page No.
	PROBLEM 1	
Fig 1.1	Univariate analysis of Numeric columns	10,11,12
Fig 1.2	Univariate analysis of object type column – runqsz	13
Fig 1.3	Pairplot of Numerical vs Numerical variables	14
Fig 1.4	Barplot of runqsz (Run queue size) vs Numerical variables	15,16,17
Fig 1.5	Correlation Plot for all Numerical variables	17
Fig 1.6	Outlier detection in the Numerical fields in the dataset	19
Fig 1.7	Outlier treatment in the numerical fields of the dataset	20
Fig 1.8	ROC Curves for all Models for Training data	25
Fig 1.9	ROC Curves for all Models for Testing data	25
Fig 1.10	Feature Importance in Gradient Boosting Tuned Model	28
	PROBLEM 2	
Fig 2.1	20 Most Common words occurring in Roosevelt Speech	35
Fig 2.2	20 Most Common words occurring in Kennedy Speech	35
Fig 2.3	20 Most Common words occurring in Nixon Speech	36
Fig 2.4	Word Cloud of Roosevelt Speech	36
Fig 2.5	Word Cloud of Roosevelt Speech	37
Fig 2.6	Word Cloud of Roosevelt Speech	37

LIST OF TABLES

Sl. No	TITLE	Page No.
	PROBLEM 1	
Table 1.1	Reading the first five rows of the dataset	8
Table 1.2	Checking datatype of all columns	9
Table 1.3	Statistical summary of numerical type columns	9
Table 1.4	Statistical summary of object type columns	10
Table 1.5	Checking for missing values	17
Table 1.6	Before Encoding the data	19
Table 1.7	Checking datatypes of columns after encoding	20
Table 1.8	Columns after encoding	20
Table 1.9	X dataset with all variables except vote	20
Table 1.10	Scaled dataset	21
Table 1.11	Performance scores of all models	22
Table 1.12	Confusion Matrix and Classification Report	24
Table 1.13	AUC Scores of Training and Testing Dataset	24
Table 1.14	Model Performance with and without training	26
	PROBLEM 2	
Table 2.1	Roosevelt speech dataframe	32
Table 2.2	Kennedy speech dataframe	32
Table 2.3	Nixon speech dataframe	32
Table 2.4	Tokenization of speech1	33

Scoring guide (Rubric) - PM Project Rubric

Criteria	Points
Problem 1 - Define the problem and perform exploratory Data Analysis- Problem definition - Check shape, Data types, statistical summary - Univariate analysis - Multivariate analysis - Use appropriate visualizations to identify the patterns and insights - Key meaningful observations on individual variables and the relationship between variables	**6**
Problem 1 - Data Pre-processing Prepare the data for modelling: - Outlier Detection (treat, if needed) – Encode the data – Data split – Scale the data (and state your reasons for scaling the features.)	**2**
Problem 1- Model Building - Metrics of Choice (Justify the evaluation metrics) - Model Building (KNN, Naive bayes, Bagging, Boosting)	**10**
Problem 1 – Model Performance and Evaluation - Check the confusion matrix and classification me trics for all the models (for both train and test dataset) - ROC-AUC score and plot the curve - Comment on all the model performance	**8**
Problem 1 – Model Performance Improvement - Improve the model performance of bagging and boosting models by tuning the model - Comment on the model performance improvement on training and test data.	**9**
Problem 1 – Final Model Selection - Compare all the model built so far - Select the final model with the proper justification - Check the most important features in the final model and draw inferences.	**4**
Problem 1 - Actionable Insights & Recommendations - Compare all four models - Conclude with the key takeaways for the business	**6**
Problem 2 - Define the problem and Perform Exploratory Data Analysis - Problem Definition - Find the number of Character, words & sentences in all three speeches	**3**
Problem 2 – Text Cleaning - Stopword removal - Stemming - find the 3 most common words used in all three speeches	**3**
Problem 2 - Plot Word cloud of all three speeches - Show the most common words used in all three speeches in the form of word clouds	**3**
Business Report Quality -Adhere to the business report checklist	**6**
TOTAL	**60**

Data Description

Problem 1

1. **vote**: Party choice: Conservative or Labour
2. **age**: in years
3. **economic.cond.national**: Assessment of current national economic conditions, 1 to 5.
4. **economic.cond.household**: Assessment of current household economic conditions, 1 to 5.
5. **Blair**: Assessment of the Labour leader, 1 to 5.
6. **Hague**: Assessment of the Conservative leader, 1 to 5.
7. **Europe**: an 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Eurosceptic' sentiment.
8. **political.knowledge**: Knowledge of parties' positions on European integration, 0 to 3.
9. **gender:** female or male.

Problem 2

The following speeches of the Presidents of the United States of America are taken:

1. President Franklin D. Roosevelt in 1941
2. President John F. Kennedy in 1961
3. President Richard Nixon in 1973

Code Snippet to extract the three speeches:

"

```
import nltk
nltk.download('inaugural')
from nltk.corpus import inaugural
inaugural.fileids()
inaugural.raw('1941-Roosevelt.txt')
inaugural.raw('1961-Kennedy.txt')
inaugural.raw('1973-Nixon.txt')
```

"

Problem - 1

Context

CNBE, a prominent news channel, is gearing up to provide insightful coverage of recent elections, recognizing the importance of data-driven analysis. A comprehensive survey has been conducted, capturing the perspectives of 1525 voters across various demographic and socio-economic factors. This dataset encompasses 9 variables, offering a rich source of information regarding voters' characteristics and preferences.

Objective

The primary objective is to leverage machine learning to build a predictive model capable of forecasting which political party a voter is likely to support. This predictive model, developed based on the provided information, will serve as the foundation for creating an exit poll. The exit poll aims to contribute to the accurate prediction of the overall election outcomes, including determining which party is likely to secure the majority of seats.

1.1 Define the problem and perform EDA (Exploratory Data Analysis)

1.1.1 Importing the libraries

Import the necessary libraries for data processing, data visualization, modelling, validation, data splitting, building the Naïve Bayes, KNN, Bagging and Boosting models and to check the model performance.

1.1.2 Reading and loading the dataset Election_data:

Load and read the dataset using the read_excel() function of pandas. The first five lines of the dataset is as follows:

	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 1.1: Reading the first 5 rows of the dataset

1.1.3 Checking the shape of the dataset:

Find out the number of rows and columns in the dataset using the shape command.
The dataset has 1525 rows and 10 columns.

1.1.4 Checking the datatypes of all the columns:

Use the info() command to find the datatypes of all the columns in the dataset. It also tells you if the dataset has missing values.

```
 #   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
```

Table 1.2: Checking the datatype of all the columns

Insights:

* There are 10 columns, of which 8 are numerical datatype column and 2 object datatype column.

* The 'vote' and 'gender' columns are categorical columns.

* All columns have non-null values.

* We drop the ‘Unnamed: 0’ column as it of no use in our data analysis.

1.1.5 Checking the statistical summary:

Use the describe() function to get the statistical summary of the object type columns.

	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

Table 1.3: Statistical summary of numerical type column

	count	unique	top	freq
vote	1525	2	Labour	1063
gender	1525	2	female	812

Table 1.4: Statistical summary of object type column

Insights:

- The mean age of voting is 54 years. The minimum age of the person who voted is 24 years while the maximum age is 93 years.
- The assessment of the national and household economic condition is mediocre, at around 3 to 3.25, on a scale of 1 to 5.
- The assessment of the Labour leader, Blair, seems to be higher at an average of 3.33 with the median being 4. While, the assessment of the Conservative leader, Hague, seems to be lower on an average of 2.75, the median being 2. It clearly shows that the voters like the Labour leader Blair more than Hague.
- Eurosceptic sentiment seems to be high. The average rating of Eurosceptic sentiments is 6.75, which is more than the average of 5.
- The political knowledge of the people also seems to be moderate at an average of 1.5 with a low of 0 and a high of 3.
- The vote is mainly for the Labour party with 1063 votes. The remaining votes go for the Conservatives.
- The female voters are slightly higher than the male voters. There are 812 female voters.

1.1.6 Univariate Analysis

Univariate analysis of all the fields is done using a histogram and a boxplot. First, we divide the dataset into a dataset with numerical fields and another dataset with categorical fields.

1.1.6.1 Numerical columns

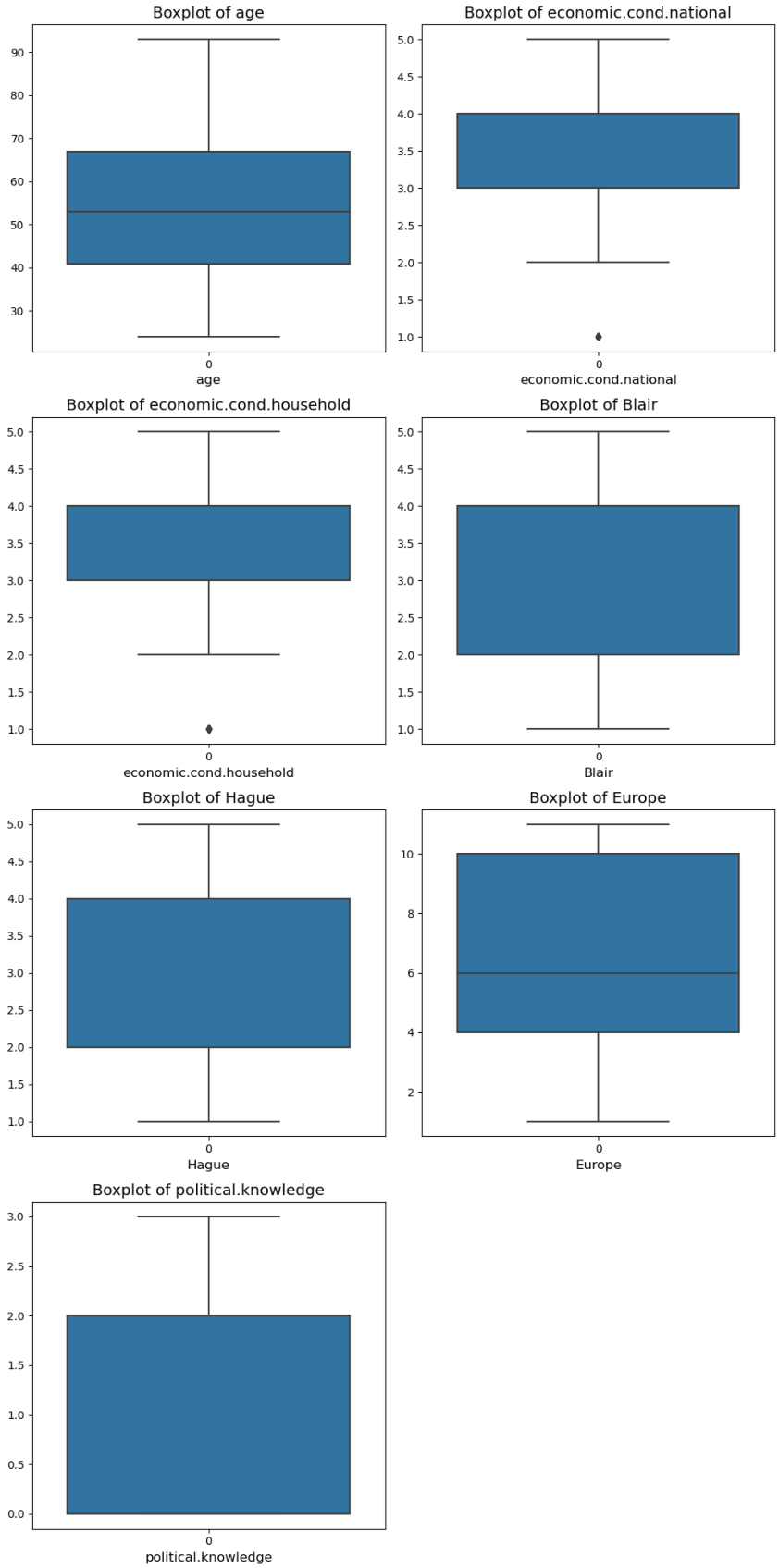


Fig 1.1: Univariate analysis of Numerical columns

1.1.6.2 Categorical columns

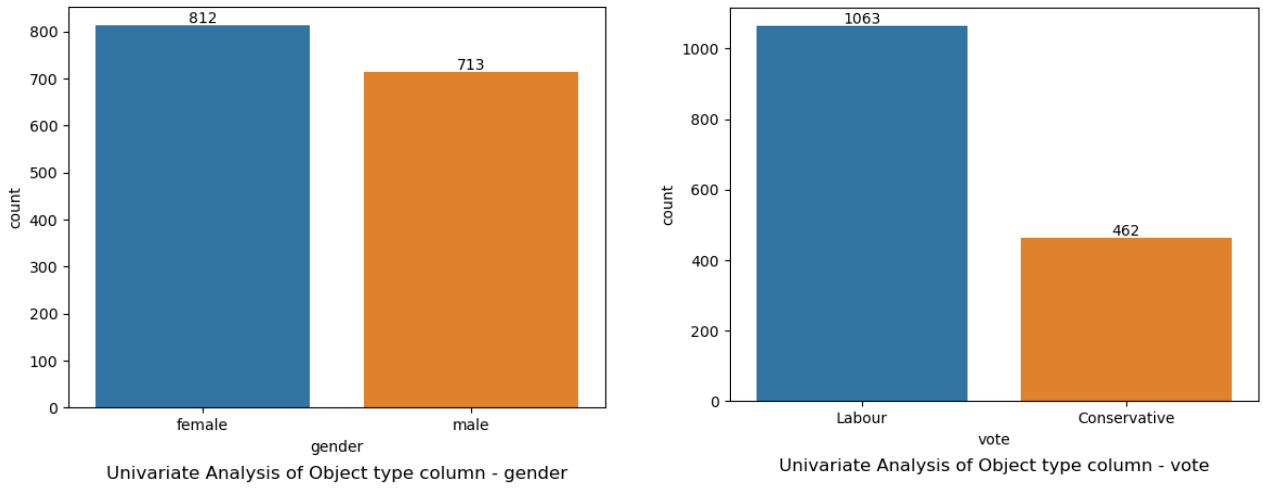


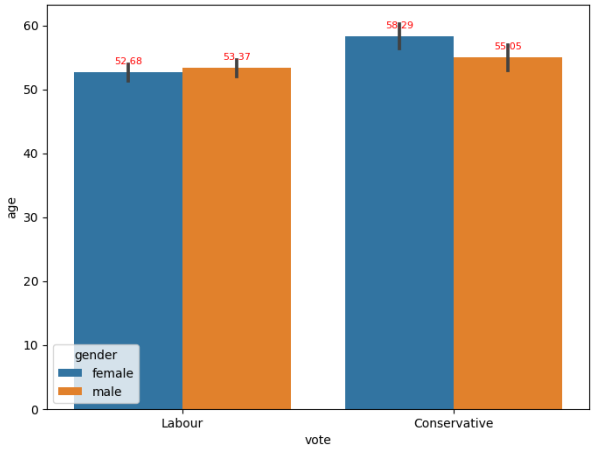
Fig 1.2: Univariate Analysis of Categorical columns

Insights:

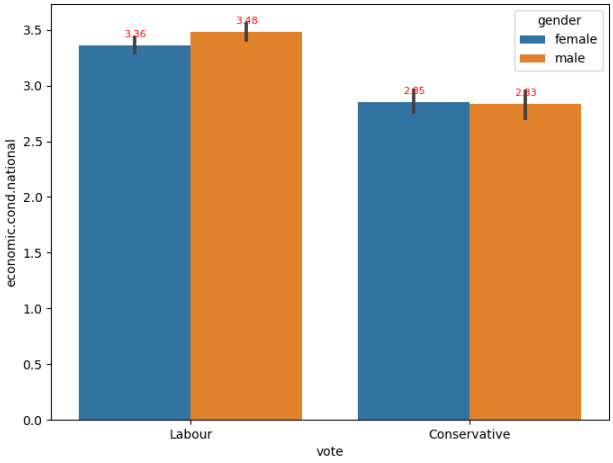
- The median age of people who voted is 53 years. The maximum age of people who have voted has crossed 90 years.
- The assessment of national economic condition is mainly from 3 to 4. The minimum assessment is 2 and the maximum assessment is 5, which means they are from very good economical background. We can see an outlier too which has an assessment of 1, which means a few voters are from very poor economic background.
- The assessment of household economic condition is mainly from 3 to 4. The minimum assessment is 2 and the maximum assessment is 5, which means they are from very good economical background.
- The support for the Labour leader, Blair and Conservative leader, Hague seem to be identical.
- The Eurosceptic sentiment seems to be at a median of 6 which is more than the average of 5.
- The political knowledge seems to be comparatively lower from a minimum of 0 to 2.
- The number of female voters are 813 and male voters are 713. The female voters are more compared to male voters.
- The people supporting the Labour party is very high, upto 1063 compared to the Conservative supporters which add up to 462.

1.1.7 Multivariate Analysis - Visualizations to identify the pattern and insights

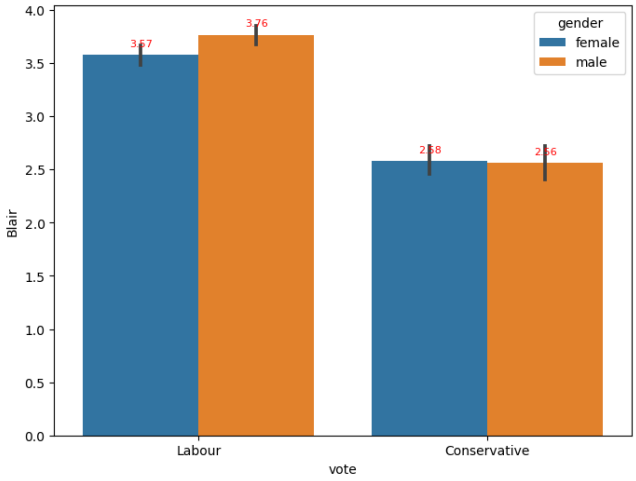
1.1.7.1 Categorical vs Numerical



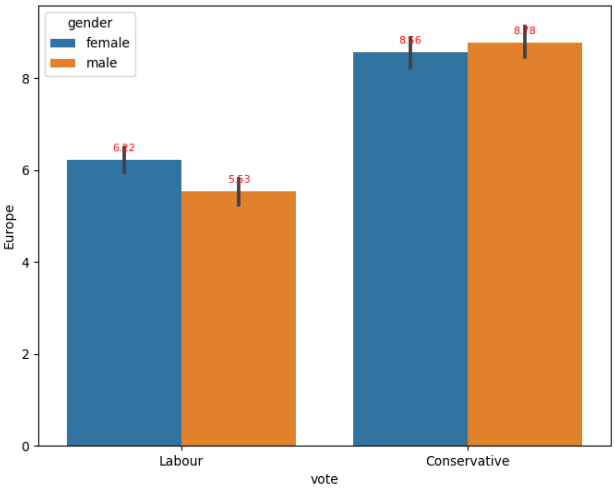
Bar plot of age against vote



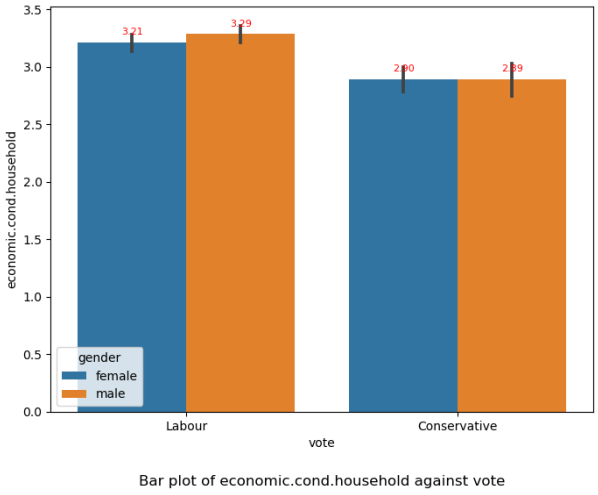
Bar plot of economic.cond.national against vote



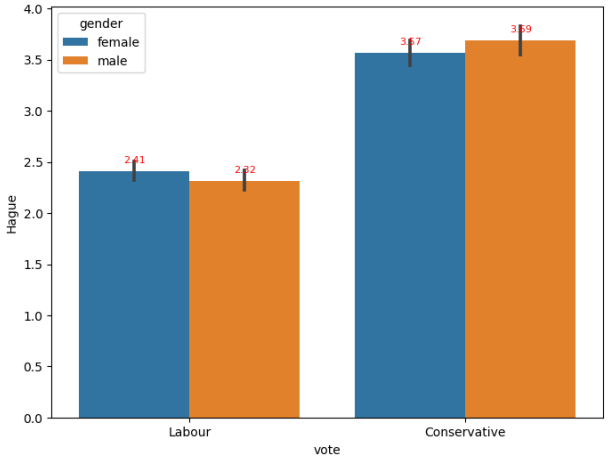
Bar plot of Blair against vote



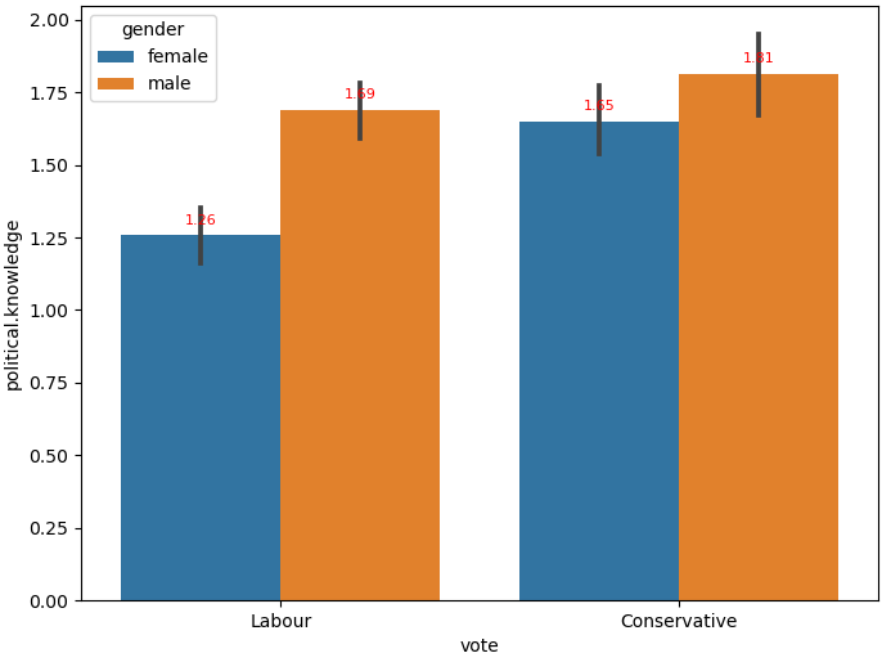
Bar plot of Europe against vote



Bar plot of economic.cond.household against vote



Bar plot of Hague against vote



Bar plot of political.knowledge against vote

Fig 1.3 Bar plot of Categorical vs Numerical columns

1.1.7.2 Numerical vs Numerical

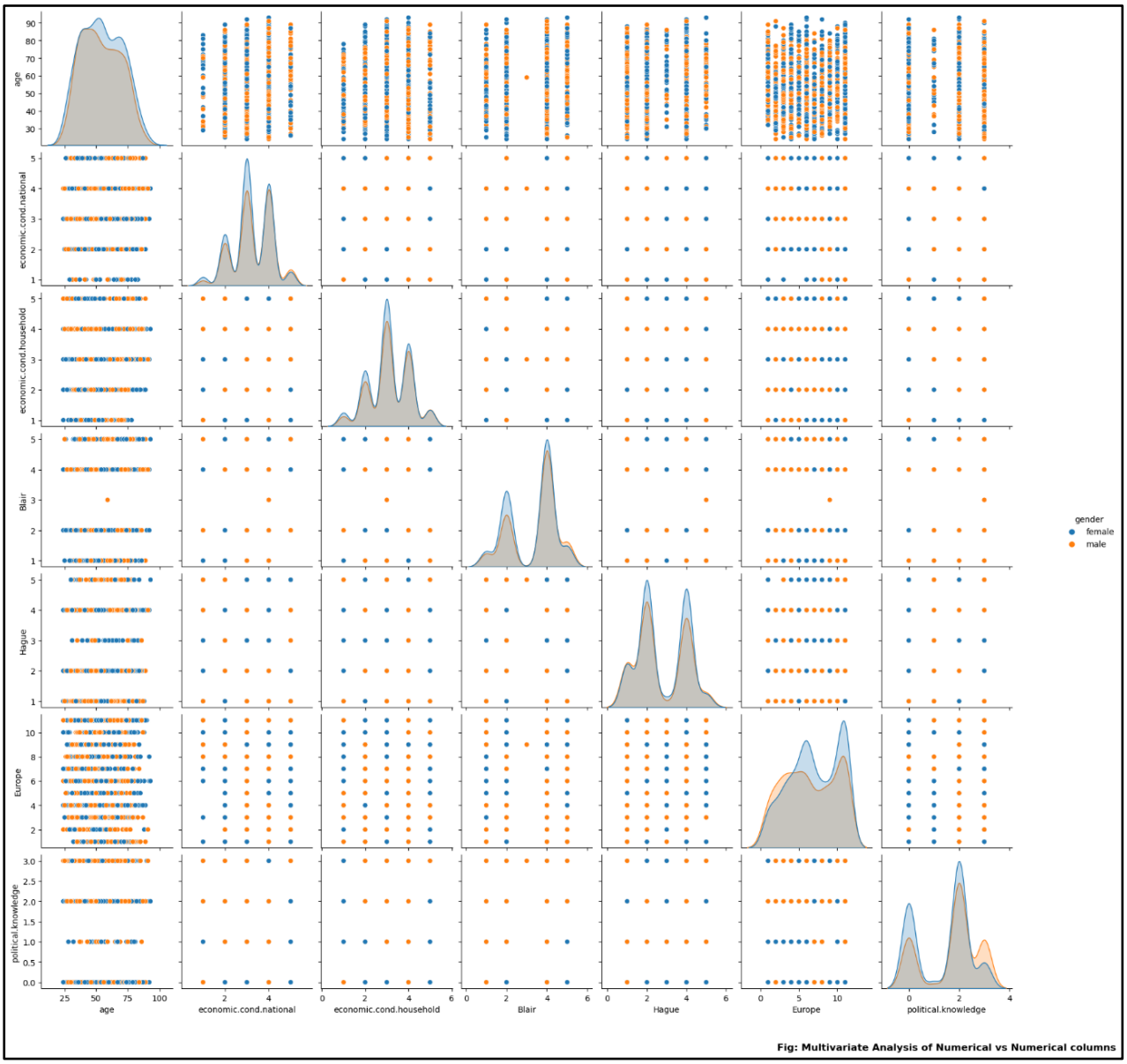


Fig 1.4: Pairplot of all the Numeric vs Numeric variables

1.1.7.3 Correlation Plot

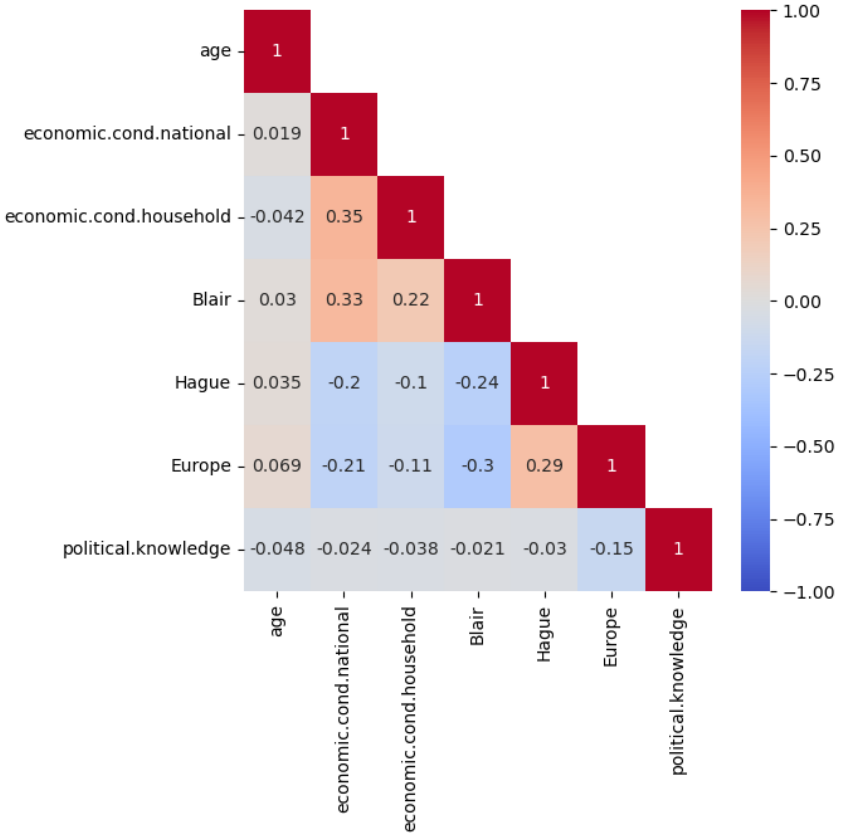


Fig 1.5: Correlation Plot of all the numerical variables

1.1.7.4 Key meaningful observations on individual variables and the relationship between variables

***** The median age of female voters voting for thhe Conservative leader is 58.29 while that of the male voters is 55 years. As we can see from the graph, this is higher than the median age of 53 years of females and 52.6 years of males voting for the Labour party.

* The national economic condition of people voting for the Labour party is more than that of the people voting for the Conservatives. That means, the richer people prefer the Labour party.

* The household economic condition of the people voting for the Labour party is around 3.2 which is more than that of the people voting for the Conservative party which is around 2.9.

* The Labour leader Blair enjoys a median vote of 3.7 from the male population while a 3.57 from the female population.

* Hague, being a Conservative leader enjoys a 3.6 median vote assessment from the male population while a 3.57 vote assessment from the female population.

* The Europe sentiment is very high among the Conservatives as compared to the Labour. The Conservative male voters constitute around 8.7 assessment and female a 8.5 in their Eurosceptic sentiment.

* The political knowledge is more among males as compared to the females. The people voting for the Conservatives seem to have a higher political knowledge as compared to the people voting for the Labour party.

* From the pairplot and the correlation plot, it is very clearly seen that there is no correlation between the numerical variables.

1.2 Data Preprocessing –

1.2.1 Prepare the data for modelling

Missing Value Treatment

Check the dataset for any missing values using the isnull() function.

```
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
```

Table 1.5: Checking for missing values

We see that there are no missing values in the dataset.

Check for duplicates

On checking for duplicates using the duplicated() function, we find that there are 8 duplicate rows.

```
<<<<<< No. of duplicate rows is 8 >>>>>>>
```

We drop the duplicate rows using the drop_duplicates() function. Now, there are no duplicate rows.

```
<<<<<<< No. of duplicate rows is 0 >>>>>>>
```

Outlier Detection and Treatment

Define a function which returns the Upper and Lower limit to detect outliers for each feature. Call the function with the column names. Cap & floor the values beyond the outlier boundaries.

Insights:

We see that all outliers have been removed and treated. We can proceed with the scaling of data.

Before outlier treatment,

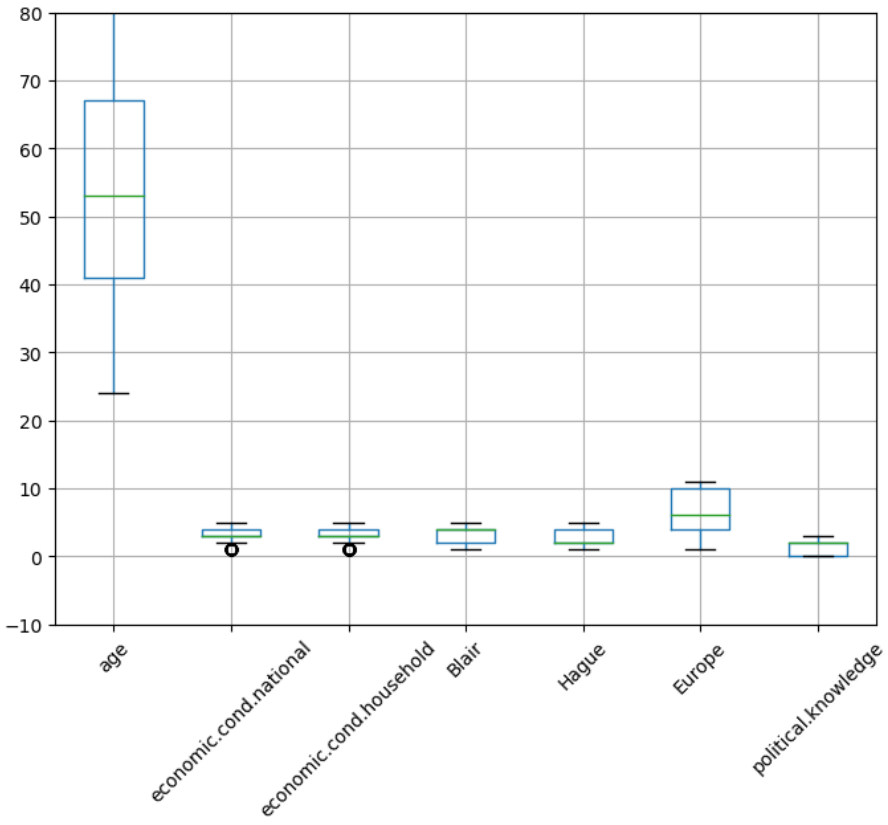


Fig 1.6: Outlier detection in the numerical fields of the dataset

After outlier treatment,

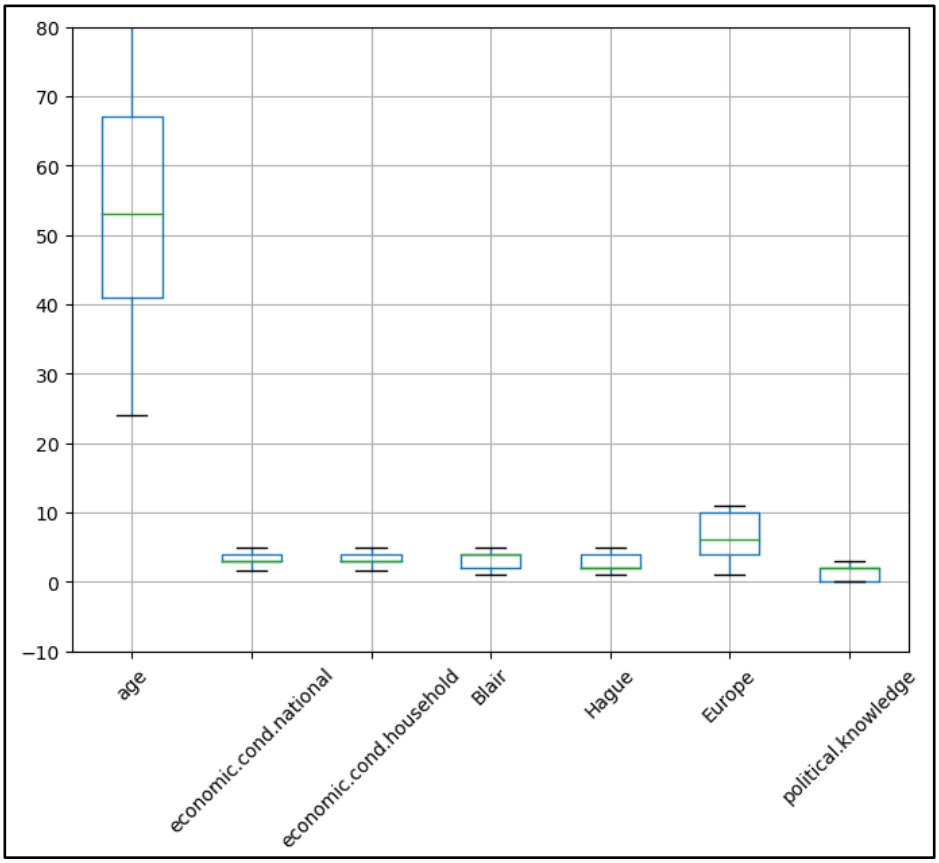


Fig 1.7 – After Outlier treatment

1.2.2 Encoding the data

One hot encoding is a technique that we use to represent categorical variables as numerical values in a machine learning model.

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

Table 1.7 - Before Encoding the data

We observe that the ‘vote’ and ‘gender’ column are object datatype. These two columns need to be changed to numerical datatype.

For encoding, we use pd.categorical.codes to change the object datatype to numerical.

```
vote                      int8
age                      int64
economic.cond.national   int64
economic.cond.household  int64
Blair                    int64
Hague                    int64
Europe                   int64
political.knowledge      int64
gender                    int8
dtype: object
```

Table 1.7 – Checking datatype of columns after encoding

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

Table 1.8 – Columns after encoding

Insights:

We have encoded the data and changed all the categorical columns into int type columns.

The female has been encoded to 0 and male has been encoded to 1.

The Conservative vote has been encoded to 0 and Labour vote has been encoded to 1.

1.2.3 Split the data

- First, split the dataset into dependent variables and independent variable. X dataset contains all the variables except the dependent variable ‘vote’ which needs to be dropped from the dataset. Y dataset contains only the dependent variable ‘vote’ in it.

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	43	3	3	4	1	2	2	0
1	36	4	4	4	4	5	2	1
2	35	4	4	5	2	3	2	1
3	24	4	2	2	1	4	0	0
4	41	2	2	1	1	6	2	1
...	...	...	...	...	...	...	...	...
1520	67	5	3	2	4	11	3	1
1521	73	2	2	4	4	8	2	1
1522	37	3	3	5	4	2	2	1
1523	61	3	3	1	4	11	2	1
1524	74	2	3	2	4	11	0	0

1517 rows × 8 columns

Table 1.9 – X dataset with all variables except ‘vote’

1.2.4 Scaling the data

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	-0.716161	3	3	4	1	2	2	0
1	-1.162118	4	4	4	4	5	2	1
2	-1.225827	4	4	5	2	3	2	1
3	-1.926617	4	2	2	1	4	0	0
4	-0.843577	2	2	1	1	6	2	1
...	...	...	...	...	...	...	...	...
1520	0.812836	5	3	2	4	11	3	1
1521	1.195085	2	2	4	4	8	2	1
1522	-1.098410	3	3	5	4	2	2	1
1523	0.430587	3	3	1	4	11	2	1
1524	1.258794	2	3	2	4	11	0	0

1517 rows × 8 columns

Table 1.10 – Scaled dataset

Insights:

On seeing the descriptive summary, the mean and median of most of the columns are almost equal, not exactly equal, indicating a normal distribution.

Scaling is necessary for the age column alone because age ranges from a minimum of 24 years to a maximum of 93 years. In this problem, age variable has a different range, compared to other variables. One variable, that is, age, might dominate another variable because it has a larger range than the other. To avoid this, we need to preprocess and scale our data so that each variable has a comparable impact on the outcome.

1.2.5 Train-Test split

- Split X and Y into training and test dataset in 70:30 ratio using the train_test_split() function. This function needs to be imported from sklearn.model_selection library.

- X_train has 1061 rows and 8 columns.
  X_test 456 rows and 8 columns.

1.3 Model Building

1.3.1 Model Building (Naïve Bayes, KNN, Bagging and Boosting)

Step 1: Invoke the GaussianNB() function for Naïve Bayes model.

Run the KNeighborsClassifier() function for the KNN model.

Run the BaggingClassifier() for the Bagging model.

For Boosting, we will carry out AdaBoosting, Gradient Boosting and XG Boost modelling.

For AdaBoosting, we run the AdaBoostClassifier() function.

For Gradient Boosting, we run the GradientBoostingClassifier() function.

For XG boosting, we run the XGBClassifier().

Step 2: We run all the models in a for loop by fitting the training set on the respective models.

Step 3: We find the Root Mean Square Error, Precision score, Accuracy score, Recall score and F1 score for both training and testing datasets.

	RMSE Train	RMSE Test	Train Precision	Test Precision	\
Naive Bayes	0.406127	0.421464	0.875486	0.865132	
KNN Classifier	0.376000	0.416228	0.889175	0.847826	
Bagging	0.106349	0.424057	0.990741	0.864686	
AdaBoosting	0.391955	0.431745	0.876433	0.847134	
Gradient Boosting	0.327789	0.405554	0.912371	0.851852	
XGB Boosting	0.092101	0.410925	0.990777	0.859873	

	Train Recall	Test Recall	Train F1_score	Test F1_score	\
Naive Bayes	0.895225	0.867987	0.885246	0.866557	
KNN Classifier	0.915119	0.900990	0.901961	0.873600	
Bagging	0.993369	0.864686	0.992053	0.864686	
AdaBoosting	0.912467	0.877888	0.894087	0.862237	
Gradient Boosting	0.938992	0.910891	0.925490	0.880383	
XGB Boosting	0.997347	0.891089	0.994052	0.875203	

	Train Accuracy	Test Accuracy
Naive Bayes	0.835061	0.822368
KNN Classifier	0.858624	0.826754
Bagging	0.988690	0.820175
AdaBoosting	0.846371	0.813596
Gradient Boosting	0.892554	0.835526
XGB Boosting	0.991517	0.831140

Table 1.11 Performance Scores of all the Models

1.3.2 Metrics of Choice (Justify the Evaluation Metrics)

Evaluation Metrics:

1) The data is not balanced as the Labour has 1063 votes and Conservatives have 462 votes. Hence, recall and precision scores are better evaluation metrics instead of accuracy scores.

2) The F1 score is the harmonic mean of precision and recall scores. It combines the precision and recall scores of a model. The F1 score is a suitable metric when recall and precision must be optimized simultaneously, especially in imbalanced datasets. The F1 score seems to be best for Naive Bayes followed by KNN and AdaBoosting models.

3) Precision score for training and testing data is closest for Naive Bayes, as in every other model, the difference of the precision scores between training and testing data is very large.

4) Recall score seems to be highest for Gradient Boosting and KNN for testing data.

5) The F1 scores are good for Naive Bayes model.

6) The Root Mean Squared Error (RMSE) should be as low as possible for the model to perform better. The RMSE values for training and test data should be close to each other to avoid overfitting or underfitting issues. Here, Naive Bayes, KNN and AdaBoosting techniques show RMSE values of train and test dataset very close to each other.

Conclusion:

We will select models:
a) which have performed approximately similar on the train and test data set.
b) which have high F1 score.
c) which have a low RMSE score.

1.4 Model Performance Evaluation

1.4.1 Check the confusion matrix and classification metrics for all the models (for both train and test dataset)

Model	Training Data	Testing Data
Naïve Bayes GaussianNB()	GaussianNB() [[211 96] [79 675]] 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	GaussianNB() [[112 41] [40 263]] 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
KNN KNeighborsClassifier()	KNeighborsClassifier() [[221 86] [64 690]] precision recall f1-score support 0 0.78 0.72 0.75 307 1 0.89 0.92 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	[[104 49] [30 273]] precision recall f1-score support 0 0.78 0.68 0.72 153 1 0.85 0.90 0.87 303 accuracy 0.83 456 macro avg 0.81 0.79 0.80 456 weighted avg 0.82 0.83 0.82 456
Bagging BaggingClassifier()	BaggingClassifier() [[303 4] [16 738]] precision recall f1-score support 0 0.95 0.99 0.97 307 1 0.99 0.98 0.99 754 accuracy 0.98 1061 macro avg 0.97 0.98 0.98 1061 weighted avg 0.98 0.98 0.98 1061	BaggingClassifier() [[110 43] [35 268]] precision recall f1-score support 0 0.76 0.72 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
Ada Boosting AdaBoostClassifier()	AdaBoostClassifier() [[210 97] [66 688]] precision recall f1-score support 0 0.76 0.68 0.72 307 1 0.88 0.91 0.89 754 accuracy 0.85 1061 macro avg 0.82 0.80 0.81 1061 weighted avg 0.84 0.85 0.84 1061	AdaBoostClassifier() [[105 48] [37 266]] precision recall f1-score support 0 0.74 0.69 0.71 153 1 0.85 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

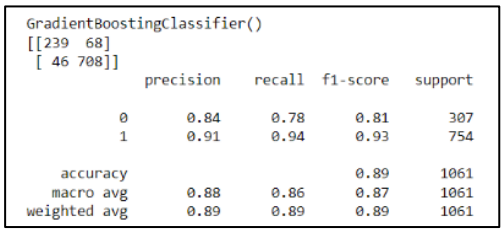
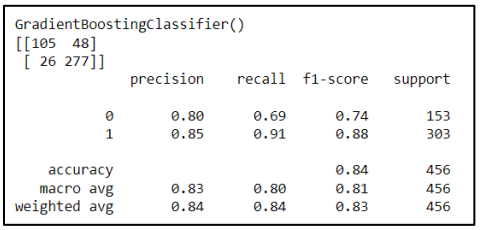
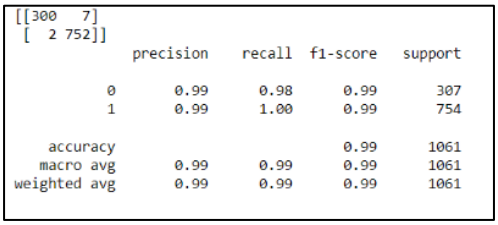
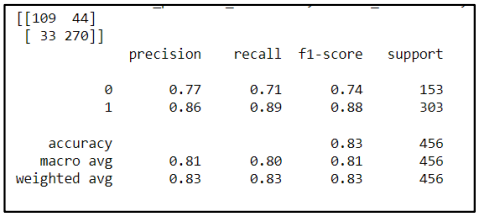
Gradient Boosting GradientBoosting Classifier()	GradientBoostingClassifier() [[239 68] [46 708]]  precision recall f1-score support 0 0.84 0.78 0.81 307 1 0.91 0.94 0.93 754 accuracy 0.89 1061 macro avg 0.88 0.86 0.87 1061 weighted avg 0.89 0.89 0.89 1061	GradientBoostingClassifier() [[105 48] [26 277]]  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 0.83 0.80 0.81 456 weighted avg 0.84 0.84 0.83 456
XG Boosting XGBClassifier()	[[300 7] [2 752]]  precision recall f1-score support 0 0.99 0.98 0.99 307 1 0.99 1.00 0.99 754 accuracy 0.99 1061 macro avg 0.99 0.99 0.99 1061 weighted avg 0.99 0.99 0.99 1061	[[109 44] [33 270]]  precision recall f1-score support 0 0.77 0.71 0.74 153 1 0.86 0.89 0.88 303 accuracy 0.83 456 macro avg 0.81 0.80 0.81 456 weighted avg 0.83 0.83 0.83 456

Table 1.12 – Confusion Matrix and Classification Report

1.4.2 ROC-AUC score and plot the curve:

The AUC scores for all the models for the training data are as follows:

Sl. No.	Model Name	AUC Score for Training dataset	AUC Score for Testing dataset
1.	Naïve Bayes	0.888	0.876
2.	KNN	0.929	0.880
3.	Bagging	0.998	0.874
4.	Ada Boosting	0.912	0.881
5.	Gradient Boosting	0.951	0.899
6.	XG Boosting	1.000	0.889

Table 1.13 – AUC Scores of Training and Testing dataset

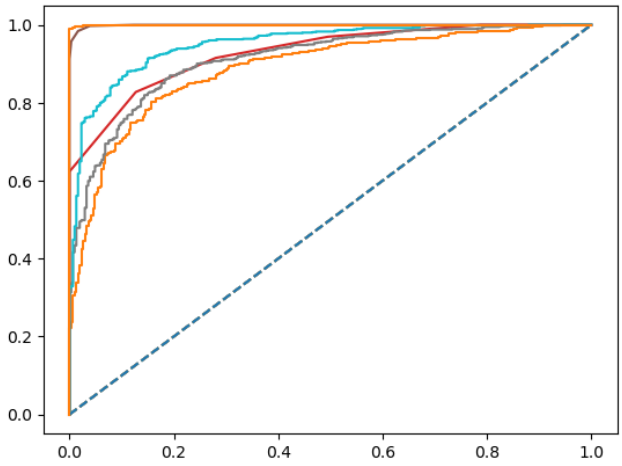


Fig 1.8 - ROC Curve for all the models for Training data

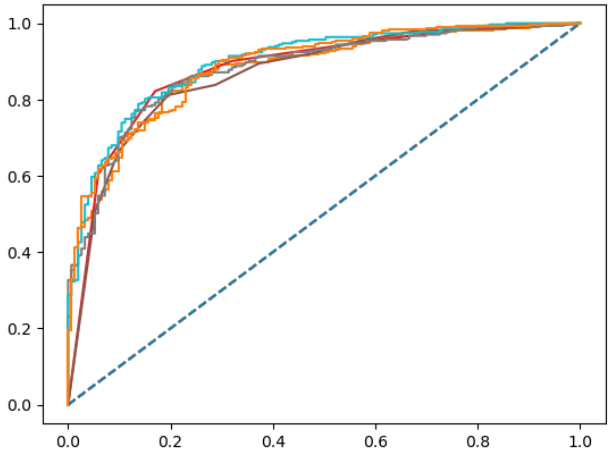


Fig 1.9 - ROC Curve for all the models for Testing data

1.4.3 Comment on all the model performances:

- From the AUC scores, it is clear that Naïve Bayes is the best model so far as the training and the testing scores are very close to each other.
- XG Boost gives a very high training AUC score as 1 but the testing AUC score is 0.889.
- The ROC curve for testing dataset for all the models is almost the same.
- For training dataset, the ROC curve is the best for XG Boost model as it covers the maximum area.
- From the confusion matrix and the classification report, it is very clear that the F1 score is the best and very close to each other for training and testing datasets for Naïve Bayes model followed by KNN model, followed by Ada Boosting.
- The classification reports for Bagging, Gradient Boosting and XG Boosting show that the F1 scores for testing dataset is very low compared to that of the training dataset, causing issues of overfitting or underfitting.

1.5 Model Performance Improvement

1.5.1 Improve the model performance of bagging and boosting models by tuning the model

Bagging Tuning:

The hyperparameters like n_estimators, max_features and max_samples in bagging were tuned. The optimum values that we got are:

```
{'max_features': 0.95, 'max_samples': 8, 'n_estimators': 500}
```

Thus, the BaggingClassifier() function was run with the above values of hyperparameters.

AdaBoost Tuning:
The hyperparameters that were tuned were n_estimators and learning_rate. The optimum values that we got are:

```
{'learning_rate': 0.1, 'n_estimators': 100}
```

Thus, the AdaBoostClassifier() function was run with the above values of hyperparameters.

GradientBoost Tuning
The hyperparameters that were tuned were n_estimators, subsample and max_features. The optimum values that we got are:

```
{'max_features': 1, 'n_estimators': 50, 'subsample': 0.2}
```

The GradientBoostingClassifier() was run with the above values of hyperparameters.

XG Boost Tuning
The hyperparameters that were tuned were n_estimators, colsample_bylevel, colsample_bytree, gamma, learning_rate, scale_pos_weight and subsample. The optimum values that we got are:

```
{'colsample_bylevel': 0.5, 'colsample_bytree': 0.5, 'gamma': 0,
'learning_rate': 0.01, 'n_estimators': 10, 'scale_pos_weight': 1,
'subsample': 0.5}
```

The XGBClassifier() was run with the above values of hyperparameters.

The values of accuracy, precision, recall and F1 scores were calculated for the tuned models and a table was created.

	Train Precision	Test Precision	Train Recall	Test Recall	Train F1_score	Test F1_score	Train Accuracy	Test Accuracy
Naive Bayes	0.88	0.87	0.90	0.87	0.89	0.87	0.84	0.82
KNN Classifier	0.89	0.85	0.92	0.90	0.90	0.87	0.86	0.83
Bagging	1.00	0.87	0.99	0.83	0.99	0.85	0.99	0.80
Bagging Tuned	0.79	0.77	0.96	0.96	0.87	0.86	0.79	0.79
AdaBoosting	0.88	0.85	0.91	0.88	0.89	0.86	0.85	0.81
AdaBoosting Tuned	0.85	0.83	0.93	0.89	0.89	0.86	0.84	0.81
Gradient Boosting	0.91	0.85	0.94	0.91	0.93	0.88	0.89	0.84
Gradient Boosting Tuned	0.90	0.84	0.93	0.91	0.91	0.87	0.88	0.83
XG Boosting	0.99	0.86	1.00	0.89	0.99	0.88	0.99	0.83
XG Boosting Tuned	0.71	0.66	1.00	1.00	0.83	0.80	0.71	0.66

Table 1.14 – Model Performance with and without tuning

1.5.2 Comment on the model performance improvement on training and test data

Bagging:

After performance tuning, the precision and accuracy have gone down, recall of the test data has improved and F1 score of test data remains the same while that for training data has reduced.

AdaBoosting:

After tuning, Almost all the parameters except Precision remains the same. Precision has reduced slightly after tuning.

Gradient Boosting:

All the evaluation metrics almost remain the same after tuning.

XG Boosting:

The Precision, F1 score and Accuracy have reduced after performance tuning. However, the Recall is 100% for training and testing dataset after tuning.

1.6 Final Model Selection

1.6.1 Compare all the models built so far.

After tuning, we observe the following:

On the basis of AUC score: Gradient Boost performs best on the test data.

On the basis of Precision: Naive Bayes is the best followed by AdaBoosting.

On the basis of Recall: XG Boosting Tuned performs the best followed by Bagging tuned.

On the basis of F1 score: Gradient Boosting Tuned performs the best followed by Naive Bayes and KNN.

On the basis of Accuracy: Gradient Boosting performs the best with 84% accuracy.

1.6.2 Select the final model with proper justification

* We have an imbalanced dataset, with around 1063 voting for Labour and 462 voting for Conservatives.

* Accuracy tells us the percentage of correctly classified observations, however, it is misleading for imbalanced datasets as it will lean towards the majority values.

* When a binary classification model makes a prediction, we have 4 outcomes, true positives and true negatives (correct answers) and false positives and false negatives (incorrect answers).

* Precision is the fraction of correct answers among observations for which our prediction is true, while Recall is the fraction of correct answers among all positive observations. The F1 score is commonly used as an evaluation metric in binary and multi-class classification. It's primarily used to compare the performance of two classifiersn.

* When the data is imbalanced and one class is a majority class, the best evaluation metric is F1 score. F1 score is used as an evaluation metric when False Positives and False Negatives are more important than the True Positives and True Negatives.

* On the basis of the F1 score, the best final model is **GRADIENT BOOSTING TUNED**. It has a F1 score of 88% on the testing data and 91% on the training data. Both the values are very near to each other.

1.6.3 Check the most important features in the final model and draw inferences.

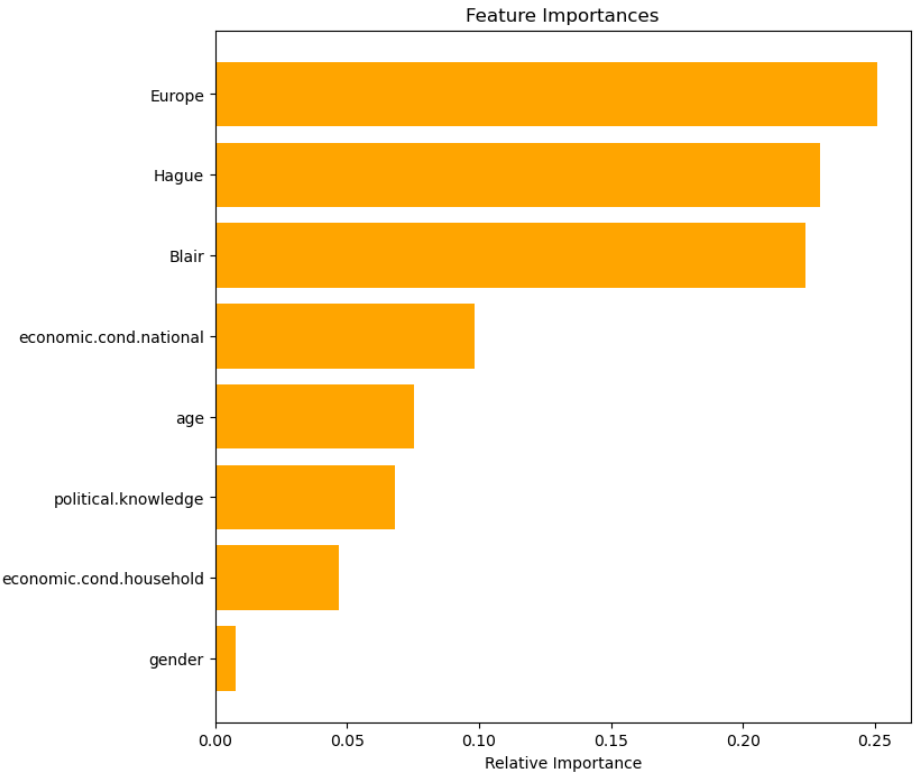


Fig 1.10 Feature Importance in Gradient Boosting Tuned Model

Insights:

- Europe followed by Hague and Blair are the most important features in Gradient Boosting Tuned model.
- Respondents' attitudes toward European integration has played a major role in the voting pattern. 'Eurosceptic' sentiment has been the key factor in predicting the party for which the voter votes for.
- The Conservative Leader Hague's assessment has played the next major role in voter prediction. The people's assessment of the work done by Hague determines the party for which the voter casts his vote for.
- Thirdly, The Labour Leader Blair's assessment has played the next major role in voter prediction. The people's assessment of the work done by Blair determines the party for which the voter casts his vote for.
- Next is the national economic conditions. This has played a significant role in determining who the voter votes for.
- Age and Political knowledge follow next. The age of the voter and his/her political knowledge plays a major role in determining the vote of the voter.

1.7 Actionable Insights and Recommendations:

1.7.1 Compare all the models – Conclude with the key takeaways for business.

- On the basis of AUC score: Gradient Boost performs best on the test data.
- On the basis of Precision: Naive Bayes is the best followed by AdaBoosting.
- On the basis of Recall: XG Boosting Tuned performs the best followed by Bagging tuned.
- On the basis of F1 score: Gradient Boosting Tuned performs the best followed by Naive Bayes and KNN.
- On the basis of Accuracy: Gradient Boosting performs the best with 84% accuracy.

- Our main business objective is to build a model to predict which party the voter will vote for – the Labour or the Conservatives to create an exit poll that will help in predicting the overall win and seats won by a particular party.

- Use Gradient Boosting with hyperparameter tuning to build the model as it gives the best optimised performance.
- We have tuned the parameters as per our understanding and limitations. Huge combinations require large processing power.
- Gathering more data may help in training the model and thus improve the predicting powers.
- The features of importance are Europe, Hague and Blair followed by national economic conditions, age and political knowledge.
- Eurosceptic sentiments seem to be the most important factor determining the vote going to which party. What the voter feels about the European integration plays the most important role in determining which party he will be voting for.
- The assessment of the leaders Hague and Blair by the voter determines the party for whom he is casting his vote for.
- The economic conditions of the country is the next important factor which tells the voter who to vote for.
- Gender hardly plays a role in casting vote for the voters.

Problem 2:

Problem Statement

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

Code Snippet to extract the three speeches:

" import nltk
nltk.download('inaugural') from nltk.corpus
import inaugural inaugural.fileids()
inaugural.raw('1941-Roosevelt.txt')
inaugural.raw('1961-Kennedy.txt')
inaugural.raw('1973-Nixon.txt') "

2.1 Problem Definition - Define the problem and Perform Exploratory Data Analysis (EDA)

2.1.1 Importing Libraries

Import the necessary libraries for data processing, data visualization and modelling, for example, import nltk, string, random, stopwords, re etc.

2.1.2 Import the speeches:

Run the code snippet given in the Problem statement to download all the three speeches, namely, ‘1941-Roosevelt.txt', '1961-Kennedy.txt', ‘1973-Nixon.txt’

2.1.3 Find the number of Character, words & sentences in all three speeches:

Use the length of raw() function to get the number of characters in the text.
Use the length of words() function to get the number of words in the text.
Use the length of sents() function to get the number of sentences in the text.

Roosevelt Speech:

```
The numbers of characters in Roosevelt 1941 speech is: 7571
The numbers of words in Roosevelt 1941 speech is: 1536
The numbers of sentences in Roosevelt 1941 speech is: 68
```

Kennedy Speech:

```
The numbers of characters in Kennedy 1961 speech is: 7618
The numbers of words in Kennedy 1961 speech is: 1546
The numbers of sentences in Kennedy 1961 speech is: 52
```

Nixon Speech:

```
The numbers of characters in Nixon 1973 speech is: 9991
The numbers of words in Nixon 1973 speech is: 2028
The numbers of sentences in Nixon 1973 speech is: 69
```

Use the info() command to find the datatypes of all the columns in the dataset. It also tells you if the dataset has missing values. Put all the speeches into a dataframe, namely, speech1, speech2 and speech3.

	Roosevelt_speech
characters	7571
words	1536
sentences	68

Table 2.1: Roosevelt Speech Dataframe

	Kennedy_speech
characters	7618
words	1546
sentences	52

Table 2.2: Kennedy Speech Dataframe

	Nixon_speech
characters	9991
words	2028
sentences	69

Table 2.3: Nixon Speech Dataframe

2.2 Text Cleaning:

2.2.1 Stopword removal:

Step 1: Convert the text to lowercase using the lower() function.

Step 2: Get the list of stopwords from nltk corpus in English and list of punctuations and store them in a list called stopwords.

Step 3: Cleaning the special characters (/@$: etc) from Regular Expressions package. Use substitution function re.sub() function for string substitution using regular expressions except(Not) a-z, A-z, replace everything with a space in the raw file.

Step 4: **Tokenization:** Split the text file into words, called tokenization, for all the three speeches. After tokenization, the words in the three speeches are:

Roosevelt speech – 1348 words

Kennedy speech – 1371 words

Nixon speech – 1819 words

An excerpt of tokenization is as below:

```
['on', 'each', 'national', 'day', 'of', 'inauguration', 'since', 'the', 'people', 'have', 'renewed', 'their', 'sense', 'of', 'd
edication', 'to', 'the', 'united', 'states', 'in', 'washington', 's', 'day', 'the', 'task', 'of', 'the', 'people', 'was', 'to',
'create', 'and', 'weld', 'together', 'a', 'nation', 'in', 'lincoln', 's', 'day', 'the', 'task', 'of', 'the', 'people', 'was',
'to', 'preserve', 'that', 'nation', 'from', 'disruption', 'from', 'within', 'in', 'this', 'day', 'the', 'task', 'of', 'the', 'p
eople', 'is', 'to', 'save', 'that', 'nation', 'and', 'its', 'institutions', 'from', 'disruption', 'from', 'without', 'to', 'u
s', 'there', 'has', 'come', 'a', 'time', 'in', 'the', 'midst', 'of', 'swift', 'happenings', 'to', 'pause', 'for', 'a', 'momen
t', 'and', 'take', 'stock', 'to', 'recall', 'what', 'our', 'place', 'in', 'history', 'has', 'been', 'and', 'to', 'rediscover',
'what', 'we', 'are', 'and', 'what', 'we', 'may', 'be', 'if', 'we', 'do', 'not', 'we', 'risk', 'the', 'real', 'peril', 'of', 'in
action', 'lives', 'of', 'nations', 'are', 'determined', 'not', 'by', 'the', 'count', 'of', 'years', 'but', 'by', 'the', 'lifeti
me', 'of', 'the', 'human', 'spirit', 'the', 'life', 'of', 'a', 'man', 'is', 'three', 'score', 'years', 'and', 'ten', 'a', 'litt
le', 'more', 'a', 'little', 'less', 'the', 'life', 'of', 'a', 'nation', 'is', 'the', 'fullness', 'of', 'the', 'measure', 'of',
'its', 'will', 'to', 'live', 'there', 'are', 'men', 'who', 'doubt', 'this', 'there', 'are', 'men', 'who', 'believe', 'that', 'd
emocracy', 'as', 'a', 'form', 'of', 'government', 'and', 'a', 'frame', 'of', 'life', 'is', 'limited', 'or', 'measured', 'by',
'a', 'kind', 'of', 'mystical', 'and', 'artificial', 'fate', 'that', 'for', 'some', 'unexplained', 'reason', 'tyranny', 'and',
'slavery', 'have', 'become', 'the', 'surging', 'wave', 'of', 'the', 'future', 'and', 'that', 'freedom', 'is', 'an', 'ebbing',
'tide', 'but', 'we', 'americans', 'know', 'that', 'this', 'is', 'not', 'true', 'eight', 'years', 'ago', 'when', 'the', 'life',
```

Table 2.4 – Tokenization of Speech 1

Step 5: Remove the words that are not useful is the next step. After stopwords removal, the lengths of the three speeches are as follows:

Roosevelt speech: 625 words

Kennedy speech: 688 words

Nixon speech: 833 words

2.2.2 Stemming

From nltk, download wordnet and omw-1.4

Stemming Using Lemmatizer - Stem the words to its root word from n1tk import WordNetUemmatizer.

Run the WordNetLemmetizer() function to stem the words to its root word.

2.2.3 Find the 3 most common words used in all three speeches

Use the frequency distribution function FreqDist() function to find the most common words in all the three speeches.

Roosevelt speech:

```
[('nation', 15), ('life', 11), ('know', 10)]
```

Kennedy speech:

```
[('let', 16), ('u', 12), ('world', 8)]
```

Nixon speech:

```
[('u', 26), ('let', 22), ('america', 21)]
```

We see that:

In Roosevelt speech, the three most common words are **nation** which occurs 15 times, **life** which occurs 11 times and **know** which occurs 10 times in the text.

In Kennedy speech, the three most common words are **let** which occurs 16 times, **u** which occurs 12 times and **world** which occurs 8 times in the text.

In Nixon speech, the three most common words are **u** which occurs 26 times, **let** which occurs 22 times followed by **america** which occurs 21 times in the text.

Plot the top 20 words used most frequently in all the three speeches.

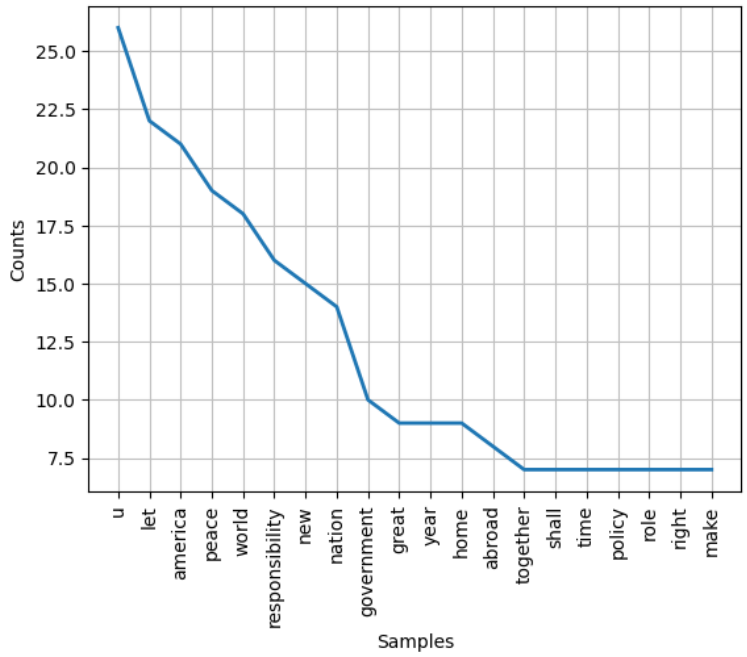


Fig 2.1 – 20 Most common words occuring in Roosevelt speech

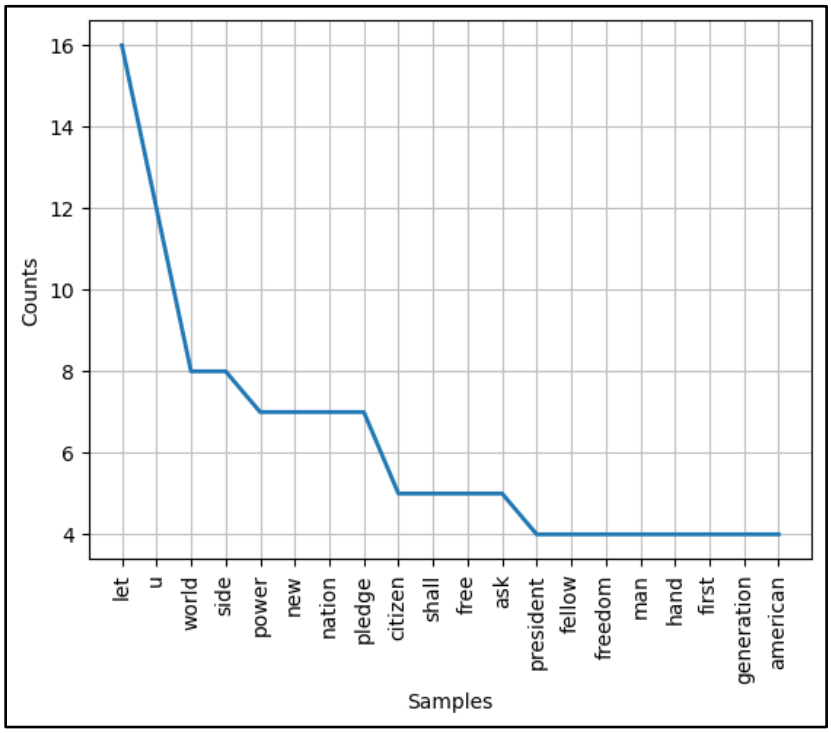


Fig 2.2 – 20 Most common words occuring in Kennedy speech

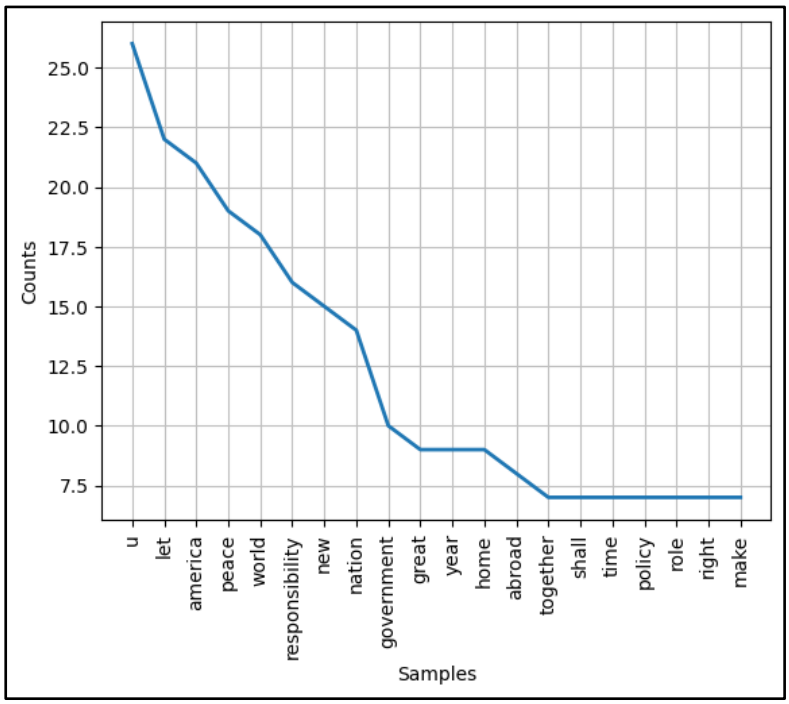


Fig 2.3 – 20 Most common words occuring in Nixon speech

2.3 Plot Word cloud of all three speeches

2.3.1 Show the most common words used in all three speeches in the form of word clouds.

Word cloud of Roosevelt speech:

Fig 2.4 - WORD CLOUD - ROOSEVELT SPEECH

Word cloud of Kennedy speech:

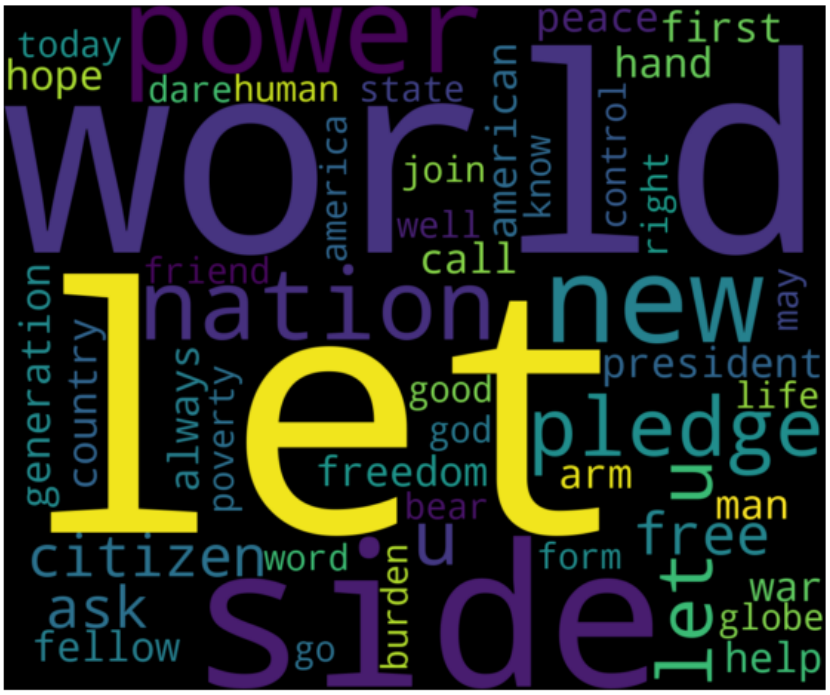


Fig 2.5 - WORD CLOUD - KENNEDY SPEECH

Word cloud of Nixon speech:

Fig 2.6 - WORD CLOUD - NIXON SPEECH

****