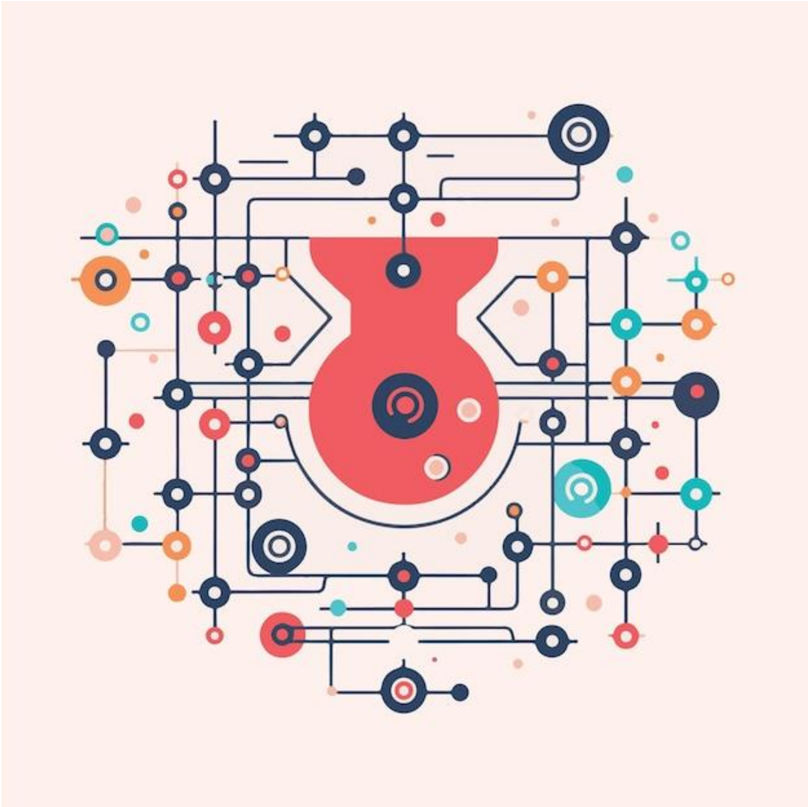

MACHINE LEARNING 2 PROJECT

By Kurt Warren Mario Gilby On July 07th 2024

Submitted to

As a part of the requirements for completion of PGP-DSBA offered in affiliation with

Table of Contents

List of Figures

List of Tables

List of Equations

No table of figures entries found.

Introduction

While doing the "**Machine Learning 2**" course, we have explored various tools and techniques that fall under machine learning. This involves analysing independent and dependent features to identify mathematical relationships between them. Specifically, we practiced clustering techniques to help us predict the outcome probability of a dataset in models such as **"Naive Bayes" and "K-Nearest Neighbours (KNN)"**. Additionally, we created ensemble methods like **"Random Forest," "Bagging," and "Boosting"** to enhance predictive accuracy. Lastly, we delved into **"Text Analytics"** to extract insights from textual data.

In the following pages, I will use these tools and techniques to review the given problem sets and answer the posed questions. The sections in this document include:

- **Overview**: A high-level overview of the problem statement/case study
- **Dataset Description**: Definition and details of the provided dataset
- **Objective**: A detailed list of the steps taken to answer the questions
- **Questions Asked**: A list of all questions with answers and supporting materials like figures and tables

The first problem set showcases regression techniques including "K-Nearest Neighbours (KNN)," "Naive Bayes," "Bagging," and "Boosting." The second problem set focuses on "Text Analytics Methods." Additionally, I will demonstrate statistical techniques and best practices learned in previous courses, such as Exploratory Data Analysis (EDA) and Data Preprocessing.

In this course, we aim to not only understand the theoretical underpinnings of these models but also to apply them effectively to solve real-world problems.

Problem 1

Overview

We have been provided with data from a comprehensive survey conducted by **CNBE**, a prominent news channel, to **deliver insightful coverage of recent elections**. The **survey** captures perspectives from **1525 voters across various demographic and socio-economic factors**. The dataset **contains 9 variables**, providing a rich source of information about voters' characteristics and preferences. The task is to use this dataset to analyse and derive insights into voter behaviour and trends.

Objective

Using the data provided, we will perform the following steps to build a predictive model for forecasting which political party a voter is likely to support:

1. **Define the problem**
2. **Explore the data**
3. **Get the statistical summary of the data**
4. **Perform data preprocessing**
5. **Apply machine learning algorithms (such as Naive Bayes and KNN)**
6. **Perform ensemble techniques (Random Forest, Bagging, Boosting)**
7. **Compare the models**
8. **Derive actionable insights and recommendations**

Dataset Description

This is the Definition of the data provided in the below table:

Variable	Description	Scale/Values
vote	Party choice	Conservative or Labour
age	Age in years	Numeric
economic.cond.national	Assessment of current national economic conditions	1 to 5
economic.cond.household	Assessment of current household economic conditions	1 to 5
Blair	Assessment of the Labour leader	1 to 5
Hague	Assessment of the Conservative leader	1 to 5
Europe	An 11-point scale that measures respondents' attitudes toward European integration	0 to 10 (Higher scores indicate 'Eurosceptic' sentiment)
political. Knowledge	Knowledge of parties' positions on European integration	0 to 3
gender	Gender	Female or Male

Table 1 Survey Data on elections.

Questions Asked

Define the problem and perform Exploratory Data Analysis.

Problem definition

We have been provided with data from a comprehensive survey conducted by CNBE, a prominent news channel, to deliver insightful coverage of recent elections. The survey captures perspectives from **1525 voters** across various demographic and socio-economic factors. The dataset contains **9 variables**, providing a rich source of information about voters' characteristics and preferences. The task is to use this dataset to analyse and derive insights into voter behaviour and trends, to do this we will **apply KNN, Naïve Bayes, Bagging and Boosting** techniques to build a predictive model.

Check Shape, Data Types, Statistical Summary

- The Data has **1525** observations and **8** independent features/attributes and 1 dependant attribute
- The Data has **7** features with data type : int64, 2 features with data type : object.
- "vote" and "gender" is of data type "object", we need to check this for values counts and convert to int or float.
- There are **8** duplicated rows with the exact values, we will remove duplicates and keep the "First" records.
- The Data has **1517** observations post duplicates removal.
- No "Null" or missing values are seen in the dataset
- All the features are category except age, we change everything to category, except Age
- Statistical Summary:
    - Age has a minimum from 24 to a max of 93 with median of 53.
    - Vote has 2 unique values with Labour being around ~2/3 the values.
    - Economic.cond.national has a mode of 3 with around 1/3rd the values.
    - Economic.cond.household has a mode 3 with around 1/3rd the values.
    - Blair has a higher rating as a mode with 4 compared to 2 for Hauge, with around ½ of the values compared to 1/3rd of the values respectively.
    - Political knowledge is low with 1/3rd of the value being 2.
    - Gender mix is fairly even with around ½ being female.
- We will convert the age too to bins are then the data across could be Categorical.
    - For binning we would use the "Freedman-Diaconis Rule":
        - H number of bins = 2 multiplied by interquartile range / cube root of number of observations

Feature	Count	Unique	Top	Freq	Mean	Std	Min	25%	50%	75%	Max
vote	1517	2	Labour	1057	-	-	-	-	-	-	-
age	1517	-	-	-	54.24	15.70	24	41	53	67	93
economic.cond.national	1517	5	3	604	-	-	-	-	-	-	-
economic.cond.household	1517	5	3	645	-	-	-	-	-	-	-
Blair	1517	5	4	833	-	-	-	-	-	-	-
Hague	1517	5	2	617	-	-	-	-	-	-	-
Europe	1517	11	11	338	-	-	-	-	-	-	-
political.knowledge	1517	4	2	776	-	-	-	-	-	-	-
gender	1517	2	female	808	-	-	-	-	-	-	-

Table 2 Statistical Summary of voter Survey Data

Exploratory Data Analysis (Univariate and Bivariate)

Univariate Analysis

We perform **"Univariate"** Analysis on the all the categorical variables by plotting Count plots.

Count plots observations

- 2/3rd of the data prefers Labour party, vs 1/3rd for Conservative.
- Economic condition national and household, eco each other with most of the respondent neutral or scoring 3 out of the 5-scale rating.
- Blair does not have any neutrals and Hague too, so this data is for the most case good as there is a distinction between will vote and won't vote.
- Europe and political knowledge are at a different scale than the other scoring features which are between 0-5.
- The distribution of observations between male and female seems to be close to equal with "females" being slightly higher.
- The distribution of the data for the age bin seems to be close to right skewed normal distribution with three nodes.

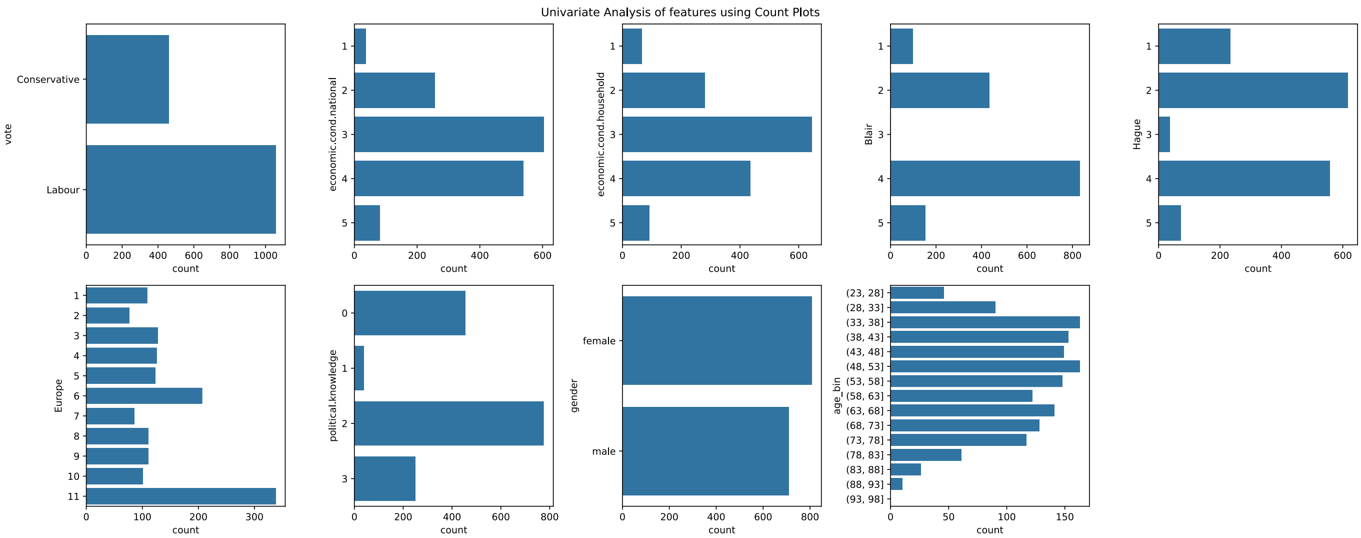

Figure 1 Survey Data Count plots

Multivariate Analysis

Using a **Count plots with the Hue of the dependant feature vote** for all the categorical variables.

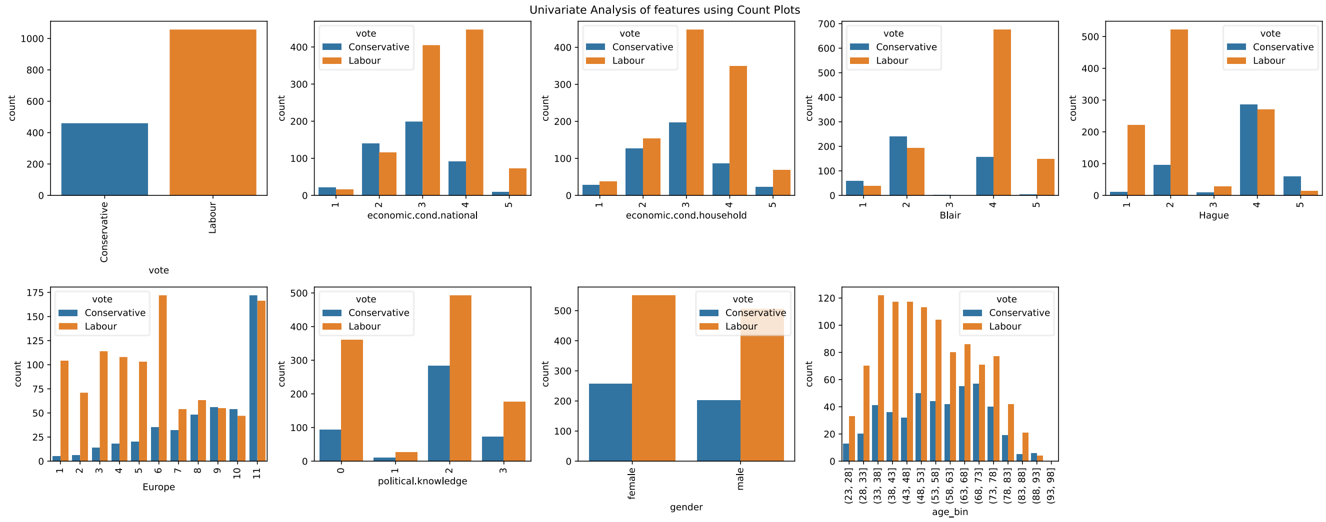

Figure 2 Survey Data split on Vote

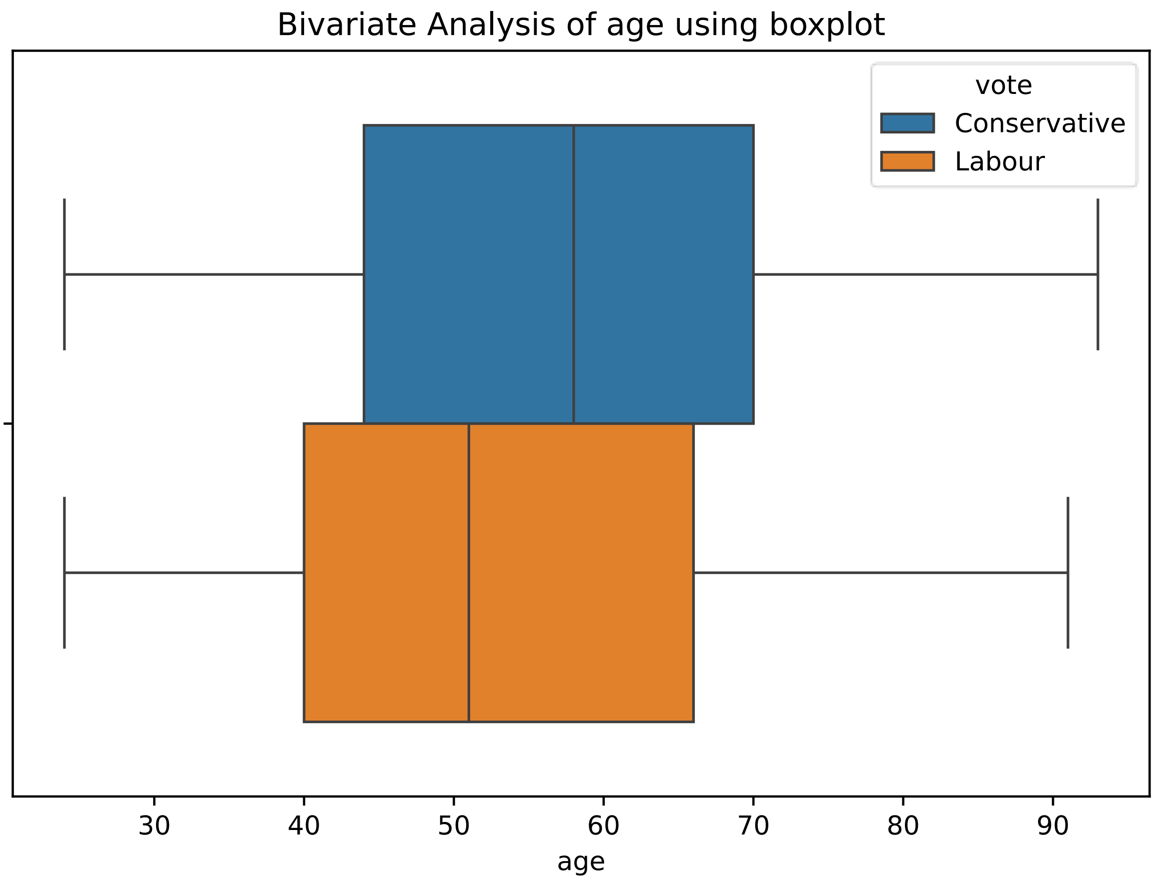

Figure 3 Age boxplots by vote

Key Observations

- A lot of conservative voters think the national and household economic condition is bad of neutral, compared to the labour voters.
- More labour voters think Blair is good compared to Hauge.
- A good part of the Labour voters are not Eurosceptics as compared to the conservative votes
    - Also, a good number both from the labour and conservative camp are highly Euro sceptical.
- The labour voters seem to be more politically knowledgeable when compared to the conservative voters.
- The gender mix seems to be the same across the conservative and the labour voters.
- In the age mix there seems to be a majority of labour voters in the ages below 60 and above 75 for the most part and in all age groups the Labour voters are higher than the conservatives.

Data Preprocessing

Prepare the data for modelling:

Missing Values Treatment:

There were no missing values, There were 8 duplicates, we removed the duplicates.

Outlier Detection (treat, if needed):

There are no Outliers and nothing to treat for the same

Encoding:

- We will encode Vote, gender and age bin.
- We will use Label encoding for Vote
    - {'Labour': 0, 'Conservative': 1}
- We will use Label encoding for gender
    - {'female': 0, 'male': 1}
- We will use one hot encoding for “age_bin”
    - This will increase the dimensions but we do not want to do label encoding to give the picture are there is some sort of order or ranking.

Spilt Data:

We split the data into Train and Test, with 70% records in Train and 30% in Test

- We have 1061 observations in Train and 456 in Test.

Scale data:

- We will **not** scale vote, gender and one hot encoded age_bins as these are pure categorical with no order or ranking.
- We will scale economic.con.national and household, Blair, Europe, and political.knowledge, as these are rated questions in the survey which have order and rank in the answer, so we would want all of these on the same scale.
    - We will use max min scaling are these series do not follow a Gaussian or normal distribution.
- We use the MinMaxScaler and fit and transform the same for the training data. And use the same scale the transform the test data.
- We do not scale the dependant variable.

Model Building

Metrics of Choice

- To evaluate between models:

  - Since we are looking to build multiple Classification Models
  - One of the best metrics to evaluate between Classification Models what is the AUC(Area under the Curve) captured by each model in the plot of the roc_auc_score.
  - The Higher the AUC of a model the better it does in discriminating/splitting the classes of the base data.
  - We do not look at "Accuracy Score" of the models and compare them with each out as a model may have a high accuracy score but a poorer performance on recall/precision /f1, for one class of the dataset.

- To evaluate between similar shortlisted models of comparable AUC(Area under the Curve) Score:

  - When we have two or more models of similar AUC(Area under the Curve) Score, we then look at the following:

    - Compare the f1 score on both the classes to see which model is doing better both for the training and Test data.
    - Compare the precision and recall on both the classes to see which model is doing better both for the training and Test data.
    - How well the models generalise to the test data, so we should calculate the accuracy scores on the Test data and compare.

Building the Models (KNN, Naive Bayes, Bagging, Boosting)

- We create the KNN and Naive Bayes models using the KNeighborsClassifier and GaussianNB methods.
- For the Bagging and Boosting we choose the same number of estimators as "100" and random state as "42" to be able to compare across these models
- We create the Bagging model using the BaggingClassifier method
- We create two models for Boosting one using the AdaBoostClassifier and other using the GradientBoostingClassifier

Model Performance evaluation

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

KNeighborsClassifier

- Training
  - f1 score of 0.89 and 0.76 on class of 0(Labour) and class of 1 (Conservative) respectively
  - Precision of 0.88 and 0.78 on class of 0(Labour) and class of 1 (Conservative) respectively, lesser FP for class 0
  - Recall of 0.90 and 0.74 class of 0(Labour) and class of 1 (Conservative) respectively, lesser FN for class 0
  - overall accuracy 0.85.
- Test
  - f1 score of 0.88 and 0.66 on class of 0(Labour) and class of 1 (Conservative) respectively
  - Precision of 0.87 and 0.69 on class of 0(Labour) and class of 1 (Conservative) respectively, lesser FP for class 0
  - Recall of 0.89 and 0.66 class of 0(Labour) and class of 1 (Conservative) respectively, lesser FN for class 0
  - overall accuracy 0.82.

GaussianNB

- Training
  - f1 score of 0.86 and 0.68 on class of 0(Labour) and class of 1 (Conservative) respectively
  - Precision of 0.84 and 0.72 on class of 0(Labour) and class of 1 (Conservative) respectively, lesser FP for class 0
  - Recall of 0.88 and 0.64 class of 0(Labour) and class of 1 (Conservative) respectively, lesser FN for class 0
  - overall accuracy 0.81.
- Test
  - f1 score of 0.86 and 0.58 on class of 0(Labour) and class of 1 (Conservative) respectively
  - Precision of 0.84 and 0.62 on class of 0(Labour) and class of 1 (Conservative) respectively, lesser FP for class 0
  - Recall of 0.88 and 0.54 class of 0(Labour) and class of 1 (Conservative) respectively, lesser FN for class 0
  - overall accuracy 0.79.

BaggingClassifier

- Training
  - f1 score of 1.00 and 1.00 on class of 0(Labour) and class of 1 (Conservative) respectively
  - Precision of 1.00 and 0.99 on class of 0(Labour) and class of 1 (Conservative) respectively, lesser FP for class 0
  - Recall of 1.00 and 1.00 class of 0(Labour) and class of 1 (Conservative) respectively, lesser FN for class 0
  - overall accuracy 1.00.
- Test
  - f1 score of 0.88 and 0.65 on class of 0(Labour) and class of 1 (Conservative) respectively
  - Precision of 0.86 and 0.69 on class of 0(Labour) and class of 1 (Conservative) respectively, lesser FP for class 0
  - Recall of 0.90 and 0.62 class of 0(Labour) and class of 1 (Conservative) respectively, lesser FN for class 0
  - overall accuracy 0.82.

AdaBoostClassifier

- Training
  - f1 score of 0.88 and 0.73 on class of 0(Labour) and class of 1 (Conservative) respectively
  - Precision of 0.87 and 0.76 on class of 0(Labour) and class of 1 (Conservative) respectively, lesser FP for class 0
  - Recall of 0.90 and 0.70 class of 0(Labour) and class of 1 (Conservative) respectively, lesser FN for class 0
  - overall accuracy 0.84.
- Test
  - f1 score of 0.89 and 0.69 on class of 0(Labour) and class of 1 (Conservative) respectively
  - Precision of 0.87 and 0.73 on class of 0(Labour) and class of 1 (Conservative) respectively, lesser FP for class 0
  - Recall of 0.91 and 0.66 class of 0(Labour) and class of 1 (Conservative) respectively, lesser FN for class 0
  - overall accuracy 0.84.

GradientBoostingClassifier

- Training
  - f1 score of 0.92 and 0.81 on class of 0(Labour) and class of 1 (Conservative) respectively
  - Precision of 0.90 and 0.85 on class of 0(Labour) and class of 1 (Conservative) respectively, lesser FP for class 0
  - Recall of 0.94 and 0.78 class of 0(Labour) and class of 1 (Conservative) respectively, lesser FN for class 0
  - overall accuracy 0.89.
- Test
  - f1 score of 0.89 and 0.69 on class of 0(Labour) and class of 1 (Conservative) respectively
  - Precision of 0.88 and 0.74 on class of 0(Labour) and class of 1 (Conservative) respectively, lesser FP for class 0
  - Recall of 0.91 and 0.66 class of 0(Labour) and class of 1 (Conservative) respectively, lesser FN for class 0
  - overall accuracy 0.84.

Traning						
	F1 Score					
	Model: ROC-AUC	KNeighborsClassifier : 0.83	GaussianNB : 0.84	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89
Classes	Labour	0.89	0.86	1.00	0.88	0.92
	Conservative	0.76	0.68	1.00	0.73	0.81
	Precision					
	Model: ROC-AUC	KNeighborsClassifier : 0.83	GaussianNB : 0.84	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89
Classes	Labour	0.88	0.84	1.00	0.87	0.90
	Conservative	0.78	0.72	0.99	0.76	0.85
	Recall					
	Model: ROC-AUC	KNeighborsClassifier : 0.83	GaussianNB : 0.84	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89
Classes	Labour	0.90	0.88	1.00	0.90	0.94
	Conservative	0.74	0.64	1.00	0.70	0.78
	Overall Accuracy					
	Model: ROC-AUC	KNeighborsClassifier : 0.83	GaussianNB : 0.84	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89
		0.85	0.81	1.00	0.84	0.89

Figure 4 Training Data Model Metrics Compare Table

Test						
	F1 Score					
	Model: ROC-AUC	KNeighborsClassifier : 0.83	GaussianNB : 0.84	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89
Classes	Labour	0.88	0.86	0.88	0.89	0.89
	Conservative	0.66	0.58	0.65	0.69	0.69
	Precision					
	Model: ROC-AUC	KNeighborsClassifier : 0.83	GaussianNB : 0.84	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89
Classes	Labour	0.87	0.84	0.86	0.87	0.88
	Conservative	0.69	0.62	0.69	0.73	0.74
	Recall					
	Model: ROC-AUC	KNeighborsClassifier : 0.83	GaussianNB : 0.84	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89
Classes	Labour	0.89	0.88	0.90	0.91	0.91
	Conservative	0.66	0.54	0.62	0.66	0.66
	Overall Accuracy					
	Model: ROC-AUC	KNeighborsClassifier : 0.83	GaussianNB : 0.84	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89
		0.82	0.79	0.82	0.84	0.84

Figure 5 Test Data Model Metrics Compare Table

```
Model Name: KNeighborsClassifier
Training Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.90      0.89       726
           1       0.78      0.74      0.76       335

    accuracy                           0.85      1061
   macro avg       0.83      0.82      0.83      1061
weighted avg       0.85      0.85      0.85      1061


Confusion Matrix:
               Labour  Conservative
Labour           655            71
Conservative      87           248
--------------------------------------------------
Test Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.89      0.88       331
           1       0.69      0.63      0.66       125

    accuracy                           0.82       456
   macro avg       0.78      0.76      0.77       456
weighted avg       0.82      0.82      0.82       456


Confusion Matrix:
               Labour  Conservative
Labour           296            35
Conservative      46            79
```

Figure 6 KNN Classifier Train/Test Classification Report and Confusion Matrix

```
Model Name: GaussianNB
Training Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.88      0.86       726
           1       0.72      0.64      0.68       335

    accuracy                           0.81      1061
   macro avg       0.78      0.76      0.77      1061
weighted avg       0.80      0.81      0.81      1061


Confusion Matrix:
              Labour  Conservative
Labour          642            84
Conservative    119           216
--------------------------------------------------
Test Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.88      0.86       331
           1       0.62      0.54      0.58       125

    accuracy                           0.79       456
   macro avg       0.73      0.71      0.72       456
weighted avg       0.78      0.79      0.78       456


Confusion Matrix:
              Labour  Conservative
Labour          290            41
Conservative     57            68
```

Figure 7 Naive Bayes Classifier Train/Test Classification Report and Confusion Matrix

```
Model Name: BaggingClassifier
Training Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       726
           1       0.99      1.00      1.00       335

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


Confusion Matrix:
              Labour  Conservative
Labour          724             2
Conservative      1           334
--------------------------------------------------
Test Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.90      0.88       331
           1       0.69      0.62      0.65       125

    accuracy                           0.82       456
   macro avg       0.78      0.76      0.77       456
weighted avg       0.82      0.82      0.82       456


Confusion Matrix:
              Labour  Conservative
Labour          297            34
Conservative     48            77
```

Figure 8 Bagging Classifier Train/Test Classification Report and Confusion Matrix

```
Model Name: AdaBoostClassifier
Training Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.90      0.88       726
           1       0.76      0.70      0.73       335

    accuracy                           0.84      1061
   macro avg       0.81      0.80      0.81      1061
weighted avg       0.83      0.84      0.83      1061


Confusion Matrix:
              Labour  Conservative
Labour          652            74
Conservative    100           235
-------------------------------------------------
Test Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.91      0.89       331
           1       0.73      0.66      0.69       125

    accuracy                           0.84       456
   macro avg       0.80      0.78      0.79       456
weighted avg       0.83      0.84      0.84       456


Confusion Matrix:
              Labour  Conservative
Labour          300            31
Conservative     43            82
```

Figure 9 Ada Boost Classifier Train/Test Classification Report and Confusion Matrix

```
Model Name: GradientBoostingClassifier
Training Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.94      0.92       726
           1       0.85      0.78      0.81       335

    accuracy                           0.89      1061
   macro avg       0.88      0.86      0.86      1061
weighted avg       0.88      0.89      0.88      1061


Confusion Matrix:
              Labour  Conservative
Labour          680            46
Conservative     75           260
-------------------------------------------------
Test Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.91      0.89       331
           1       0.74      0.66      0.69       125

    accuracy                           0.84       456
   macro avg       0.81      0.78      0.79       456
weighted avg       0.84      0.84      0.84       456


Confusion Matrix:
              Labour  Conservative
Labour          302            29
Conservative     43            82
```

Figure 10 Gradient Boost Classifier Train/Test Classification Report and Confusion Matrix

ROC-AUC score and plot the curve

ROC-AUC score:

- KNeighborsClassifier : 0.83
- GaussianNB : 0.84
- BaggingClassifier : 0.85
- AdaBoostClassifier : 0.89
- GradientBoostingClassifier : 0.89

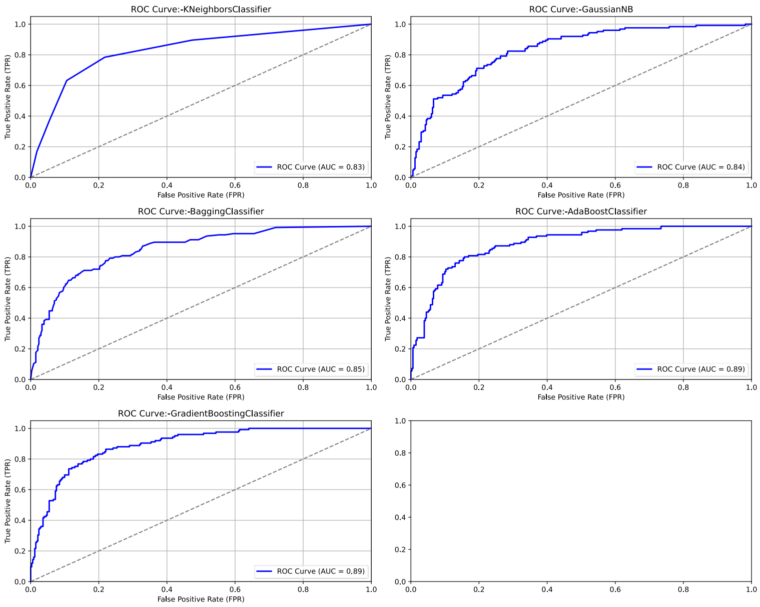

Figure 11 ROC-AUC Plots for all Models

All the model performance

KNeighborsClassifier :

- In Training
  - Does better in F1 and Precision and Recall Score for both classes than GaussianNB and AdaBoostClassifier
- On Test Data
  - Does better in F1 and Precision and Recall Score for both classes than GaussianNB and BaggingClassifier
  - ROC - AUC Score is the least among all the Models.

GaussianNB :

- In Training
  - Does badly in all the metrics all the other Models
- On Test Data
  - Does badly in all the metrics all the other Models
  - ROC - AUC Score is the only better than KNeighborsClassifier

BaggingClassifier :

- In Training
  - Over fits the data with all metrics very close to or equal to 1
- On Test Data
  - Does better in F1 and Precision and Recall Score for both classes than GaussianNB
  - ROC - AUC Score is better than KNeighborsClassifier and GaussianNB

AdaBoostClassifier :

- In Training
  - Does better in F1 and Precision and Recall Score for both classes than GaussianNB
- On Test Data
  - Does better in F1 and Precision and Recall Score for both classes against all models except GradientBoostingClassifier
  - ROC - AUC Score is better than all models except GradientBoostingClassifier

GradientBoostingClassifier :

- In Training
  - Does better in F1 and Precision and Recall Score for both classes against all models except BaggingClassifier
- On Test Data
  - Does better in F1 and Precision and Recall Score for both classes against all models.
  - ROC - AUC Score is better than all models.

Model Performance Improvement:

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

From all the models, we choose three with the highest ROC-AUC score, which are BaggingClassifier, AdaBoostClassifier and GradientBoostingClassifier. In order to check for possible improvements to these models we will run, GridsearchCV to find the best parameters to use to run these Models.

for BaggingClassifier we run the params:

- 'n_estimators': [50, 100, 150, 200, 250,300,350,400,450,500],
- 'max_samples': [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0],
- 'max_features': [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0],
- 'random_state': [42]

This resulted in the best params of:

- Best Parameters: {'max_features': 0.2, 'max_samples': 0.3, 'n_estimators': 150, 'random_state': 42}

for AdaBoostClassifier we run the params:

- 'n_estimators': [50, 100, 150, 200, 250,300,350,400,450,500],
- 'learning_rate': [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2.0],
- 'algorithm': ['SAMME', 'SAMME.R'],
- 'random_state': [42]

This resulted in the best params of:

- Best Parameters: {'algorithm': 'SAMME', 'learning_rate': 0.8, 'n_estimators': 150, 'random_state': 42}

for GradientBoostingClassifier we run the params:

- 'n_estimators': [50, 100, 150, 200, 250,300,350,400,450,500],
- 'learning_rate': [0.5,0.6,0.7,0.8,1.0],
- 'min_samples_split': [2, 4, 6, 8, 10],
- 'min_samples_leaf': [1, 2, 3, 4, 5],
- 'subsample': [0.1,0.3,0.5,0.8,1.0],
- 'random_state': [42]

This resulted in the best params of:

- Best Parameters: {'learning_rate': 0.5, 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 50, 'random_state': 42, 'subsample': 1.0}

Re-run the bagging and boosting models with the best parameters
model performance improvement on training and test data:

Tuning the model for BaggingClassifier and rerunning the new model on the Training and Test we see:

- The ROC-AUC Score increased from 0.85 to 0.89
- Both on the Training and Test Data the Model does not Overfit now but does better in Generalizations
- In all the metrics between the Train and Test of the Original and the Tuned Model there is a drop and significant drop for the class 1 (Conservative)

Tuning the model for AdaBoostClassifier and rerunning the new model on the Training and Test we see:

- The ROC-AUC Score remained the same
- Both on the Training and Test Data the model has performed the same with not much change, there is a slight bump in the Precision metric for class 1 (Conservative)
- In all the metrics between the Train and Test of the Original and the Tuned Model there is no significant drop or lift.

Tuning the model for GradientBoostingClassifier and rerunning the new model on the Training and Test we see:

- The ROC-AUC Score dropped by 1% 0.88 form 0.89
- The model on Training data has performed much better that pre-tunning with all the metrics having an improvement.
- The model on Test data however has had a drop in all metrics.
- The model is now overfitting the training data.

	Training				Training-Tuned			
	F1 Score				F1 Score			
	Model: ROC-AUC	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89	Model: ROC-AUC	BaggingClassifier : 0.89	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.88
Classes	Labour	1.00	0.88	0.92	Labour	0.84	0.89	0.95
	Conservative	1.00	0.73	0.81	Conservative	0.31	0.73	0.88
	Precision				Precision			
	Model: ROC-AUC	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89	Model: ROC-AUC	BaggingClassifier : 0.89	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.88
Classes	Labour	1.00	0.87	0.90	Labour	0.73	0.86	0.94
	Conservative	0.99	0.76	0.85	Conservative	0.95	0.78	0.90
	Recall				Recall			
	Model: ROC-AUC	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89	Model: ROC-AUC	BaggingClassifier : 0.89	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.88
Classes	Labour	1.00	0.90	0.94	Labour	1.00	0.91	0.95
	Conservative	1.00	0.70	0.78	Conservative	0.18	0.69	0.87
	Overall Accuracy				Overall Accuracy			
	Model: ROC-AUC	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89	Model: ROC-AUC	BaggingClassifier : 0.89	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.88
		1.00	0.84	0.89		0.74	0.84	0.93

Figure 12 Training Data Model vs Tuned Model Metrics Compare Table

	Test				Test- Tuned			
	F1 Score				F1 Score			
	Model: ROC-AUC	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89	Model: ROC-AUC	BaggingClassifier : 0.89	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.88
Classes	Labour	0.88	0.89	0.89	Labour	0.86	0.89	0.88
	Conservative	0.65	0.69	0.69	Conservative	0.25	0.69	0.67
	Precision				Precision			
	Model: ROC-AUC	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89	Model: ROC-AUC	BaggingClassifier : 0.89	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.88
Classes	Labour	0.86	0.87	0.88	Labour	0.76	0.87	0.87
	Conservative	0.69	0.73	0.74	Conservative	1.00	0.74	0.69
	Recall				Recall			
	Model: ROC-AUC	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89	Model: ROC-AUC	BaggingClassifier : 0.89	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.88
Classes	Labour	0.90	0.91	0.91	Labour	1.00	0.91	0.89
	Conservative	0.62	0.66	0.66	Conservative	0.14	0.65	0.65
	Overall Accuracy				Overall Accuracy			
	Model: ROC-AUC	BaggingClassifier : 0.85	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.89	Model: ROC-AUC	BaggingClassifier : 0.89	AdaBoostClassifier : 0.89	GradientBoostingClassifier : 0.88
		0.82	0.84	0.84		0.77	0.84	0.82

Figure 13 Test Data Model vs Tuned Model Metrics Compare Table

```
Model Name: BaggingClassifier
Training Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.73      1.00      0.84       726
           1       0.95      0.18      0.31       335

    accuracy                           0.74      1061
   macro avg       0.84      0.59      0.57      1061
weighted avg       0.80      0.74      0.67      1061


Confusion Matrix:
              Labour  Conservative
Labour           723             3
Conservative     274            61
--------------------------------------------------
Test Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.76      1.00      0.86       331
           1       1.00      0.14      0.25       125

    accuracy                           0.77       456
   macro avg       0.88      0.57      0.56       456
weighted avg       0.82      0.77      0.69       456


Confusion Matrix:
              Labour  Conservative
Labour           331             0
Conservative     107            18
```

Figure 14 Bagging Classifier Train/Test After Hyperparameter Tuning Classification Report and Confusion Matrix

```
Model Name: AdaBoostClassifier
Training Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.91      0.89       726
           1       0.78      0.69      0.73       335

    accuracy                           0.84      1061
   macro avg       0.82      0.80      0.81      1061
weighted avg       0.84      0.84      0.84      1061


Confusion Matrix:
              Labour  Conservative
Labour           659            67
Conservative     103           232
------------------------------------------------
Test Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.91      0.89       331
           1       0.74      0.65      0.69       125

    accuracy                           0.84       456
   macro avg       0.80      0.78      0.79       456
weighted avg       0.84      0.84      0.84       456


Confusion Matrix:
              Labour  Conservative
Labour           302            29
Conservative      44            81
```

Figure 15 Ada Boosting Classifier Train/Test After Hyperparameter Tuning Classification Report and Confusion Matrix

```
Model Name: GradientBoostingClassifier
Training Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.95      0.95       726
           1       0.90      0.87      0.88       335

    accuracy                           0.93      1061
   macro avg       0.92      0.91      0.91      1061
weighted avg       0.93      0.93      0.93      1061


Confusion Matrix:
              Labour  Conservative
Labour           693            33
Conservative      45           290
------------------------------------------------
Test Data Summary:

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.89      0.88       331
           1       0.69      0.65      0.67       125

    accuracy                           0.82       456
   macro avg       0.78      0.77      0.78       456
weighted avg       0.82      0.82      0.82       456


Confusion Matrix:
              Labour  Conservative
Labour           295            36
Conservative      44            81
```

Figure 16 Gradient Boosting Classifier Train/Test After Hyperparameter Tuning Classification Report and Confusion Matrix

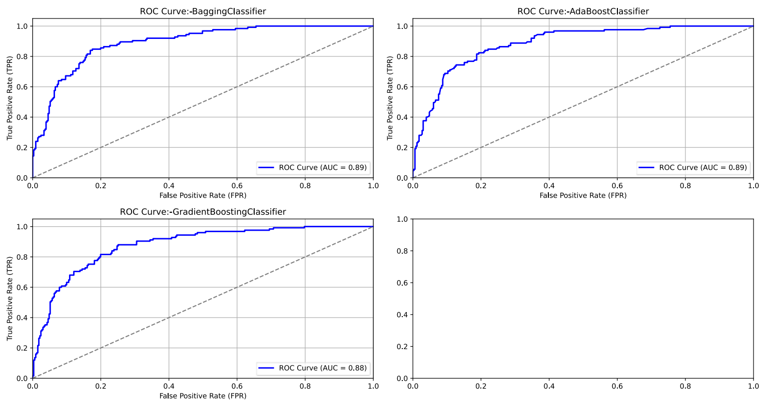

Figure 17 Bagging, Ada, Gradient Classifier ROC- AUC Plots After Hyperparameter Tuning

Final Model Selection

Comparing all the models:

- Prior to Tuning, If we compare all the models based on ROC-AUC score the three best models are BaggingClassifier, AdaBoostClassifier and GradientBoostingClassifier

Post Tuning we see that AdaBoostClassifier has:

- Highest AUC Score post parameter tuning along with BaggingClassifier
- The f1 Scores on the Training and Test are more balanced for ADAClassifier for both the classes as compared to the BaggingClassifier and GradientBoostingClassifier
- The BaggingClassifier does a good job in recall and precision for the "Labour Class" but very poor on recall for "Conservative Class" both on training and test
- ADAClassifier does a better job in regularisation and better precision and recall mix for both classes for both Training and Test Data.
- Lastly on the test data set the accuracy for the ADAClassifier is the highest

Select the final model with the proper justification

- Considering all the above points we will go with the ADAClassifier

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

- Looking at the Top Features of the Final ADABoostClassifier these are the inferences drawn:
  - Top five make-up ~80% of the features importance
    - Blair, Europe ,Hauge, age_bin(88_93], econimic.condition.national
    - The rating/popularity of "Tony Bliar" has the highest influence of how a voter would vote, if he is popular with the voter then the voter would vote for "Labour Party" else "Conservative".
    - The voter's attitude towards the European integration is a good indicator of if the voter would vote for the "Labour Party" or "Conservative", higher Europe score would indicate that the voter is more inclined towards "Conservative"
    - The rating/popularity of "William Hague" has a good influence of how a voter would vote, if he is popular with the voter then the voter would vote for "Conservative Party" else "Labour".

- Voters in the age group 88 to 92 seem to be pretty much inclined towards the conservative party, which is an exception to the other age groups, who are more "Labour party" Inclined as a proportion.

- The last top five indicator is what the voter thinks of the "econimic.condition.national" if they feel it to be not so good they tend to favour the "Conservative Party" vs the "Labour party" and vice-versa.

- Top 10 make-up ~94% of the features importance

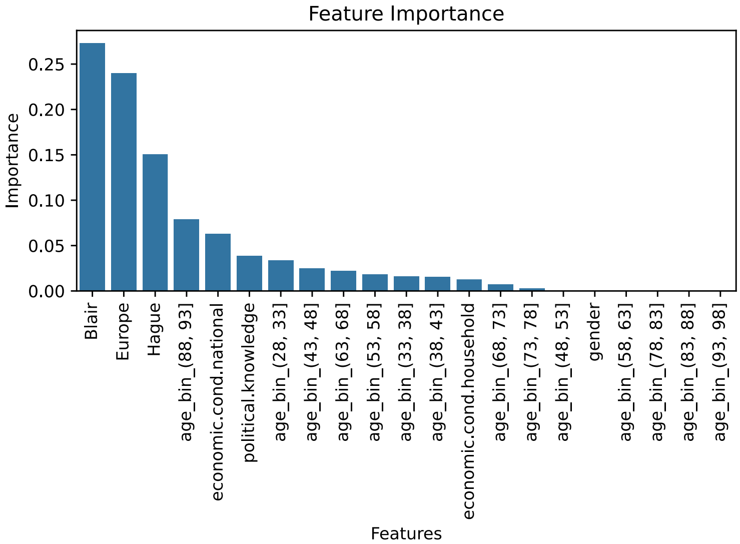

Figure 18 Descending order of Feature Importance for the Ada Boost Final Model

Features	Importance
Blair	0.273357
Europe	0.240140
Hague	0.150707
age_bin_(88, 93]	0.079071
economic.cond.national	0.062886
political.knowledge	0.038971
age_bin_(28, 33]	0.034051
age_bin_(43, 48]	0.025213
age_bin_(63, 68]	0.022402
age_bin_(53, 58]	0.018559

Figure 19 List of the Top 10 Features for the Final Ada Boost model

Actionable Insights & Recommendations

- Looking at all the four models (KNN, Naive Bayes, Bagging and Boosting):

  - We see that in this case the worse performing Model is the Naive Bayes which has the least impressive metrics both on Training and Test data, and we should not explore this further.
  - KNN does a decent job but in comparison of the Ensemble techniques of Bagging and Boosting falls short especially when we compare the ROC Area under the Curve for it against the other two.
  - Bagging overfits the training data, but is below the Boosting method on all metrics.
  - For this data set we should go with a Boosting Model.

    - we should explore more hyper parameter tuning and other boosting models like XGboost too and would benefit from this.

Key Take Aways:

- If we know the preference of the voter in terms of :
  - their views on Blair, Europe ,Hauge, and econimic.condition.national and the age of the voter.
  - we currently should be able to predict how the voter would vote with an 80% accuracy.
- we should invest more time to explore how to improve this prediction model and explore other classifiers.

Problem 2

Problem Definition

Looking at the three speeches made "President Franklin D. Roosevelt in 1941", "President John F. Kennedy in 1961" and "President Richard Nixon in 1973", we want to run text analytics to find out what are these speeches made up of and what is the common themes between them.

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

1941-Roosevelt

- The Number of "Characters": 7571, The Number of "Words": 1526, The Number of "Sentences": 68

1961-Kennedy

- The Number of "Characters": 7618, The Number of "Words": 1543, The Number of "Sentences": 52

1973-Nixon

- The Number of "Characters": 9991, The Number of "Words": 2006, The Number of "Sentences": 68

For all three speeches:

- The Number of "Characters": 25180, The Number of "Words": 5075, The Number of "Sentences": 188

Text cleaning

- using the stop words and other methods provided by the nltk(natural language tool kit), we remove:
  - common words, stop words such as e.g. 'he, she, they, us, and, the' etc
  - we also remove "punctuation" and some symbols "{'--', '``', """"}"
  - using stemming, we bring the words to the base root word.

3 most common words used in all three speeches:

1941-Roosevelt

- The top three common words are “nation”: 17 “know”: 10 “people”: 9

1961-Kennedy

- The top three common words are “let”: 16 “us”: 12 “power”: 9

1973-Nixon

- The top three common words are “us”: 26 “let”: 22 “america”: 21

For all three speeches:

- The top three common words are “us”: 46 “nation”: 40 “let”: 39

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

I have generated and displayed this for each Speech and all the three speeches below:

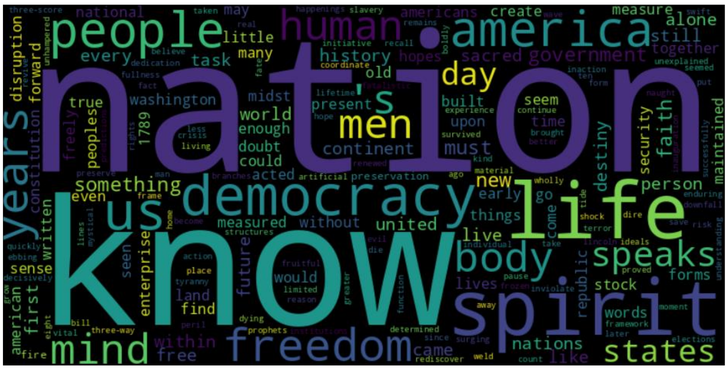

Figure 20 1941-Roosevelt Word Cloud

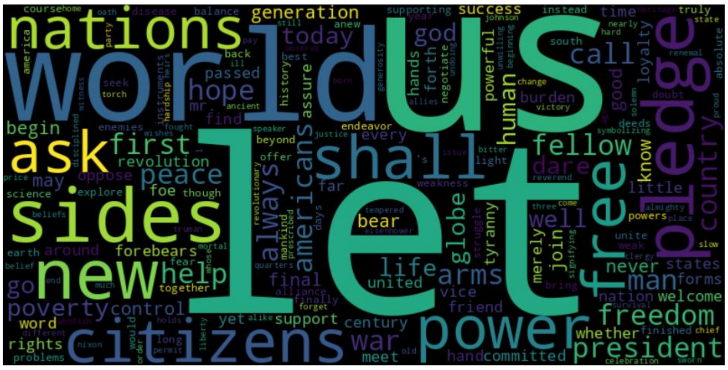

Figure 21 1961-Kennedy Word Cloud

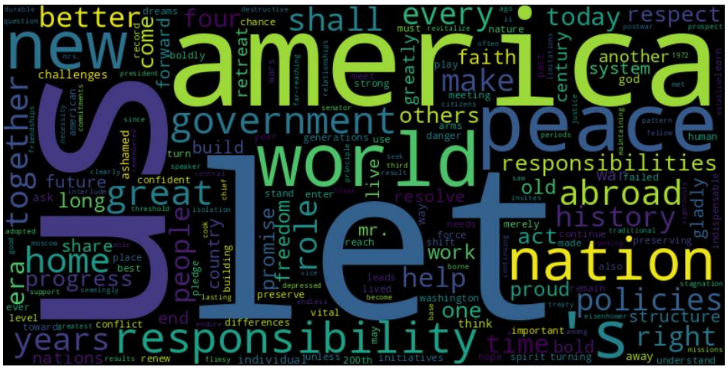

Figure 22 1973-Nixon Word Cloud

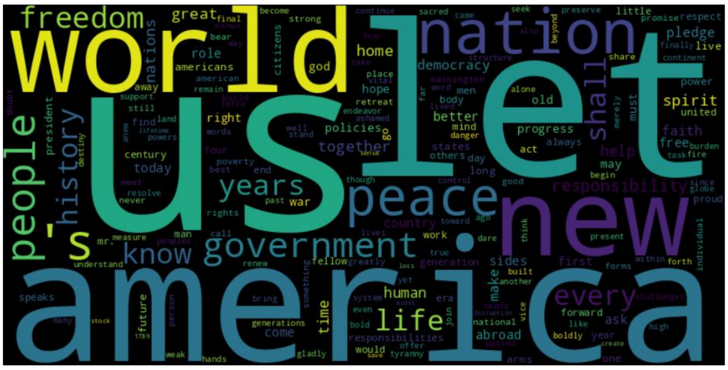

Figure 23 All Three Speeches Word Cloud