Image courtesy: https://blog.reffascode.de/tag/machine-learning/

Machine Learning Project

Business Report

July 07, 2024

Authored by: Kartik Trivedi

List of Contents

List of Figures

List of Tables

Data Dictionary

Problem 1

Column Name	Column Description	Data Type
vote	Party choice: Conservative or Labour	object
age	in years	int64
economic.cond.national	Assessment of current national economic conditions, 1 to 5.	int64
economic.cond.household	Assessment of current household economic conditions, 1 to 5.	int64
Blair	Assessment of the Labour leader, 1 to 5.	int64
Hague	Assessment of the Conservative leader, 1 to 5.	int64
Europe	an 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Eurosceptic' sentiment.	int64
political.knowledge	Knowledge of parties' positions on European integration, 0 to 3.	int64
gender	female or male.	object

Problem 2

Name	Description	Data Type
1941-Roosevelt.txt	Roosevelt speech 1941	object
1961-Kennedy.txt	Kennedy speech 1961	object
1973-Nixon.txt	Nixon speech 1973	object

Executive Summary

Problem 1

Background Information

In the news media industry, providing insightful election coverage is critical for attracting and retaining viewership. To achieve this, data-driven analysis is essential. CNBE News, a prominent news channel, has conducted a comprehensive survey capturing the perspectives of 1,525 voters across various demographic and socio-economic factors. Using this data, they aim to forecast which political party a voter is likely to support. This analysis will serve as the foundation for creating an exit poll that can accurately predict the overall election outcomes.

Business Objective

Exit polls are integral to election coverage, as accurate predictions can significantly benefit a news channel. With this in mind, CNBE News, a prominent news channel, aims to build a predictive model to forecast which political party a voter is likely to support. They will use survey data that captures voter perspectives across various demographic and socio-economic factors.

Problem Statement

The objective of this project is to conduct a comprehensive analysis of survey data that captures voter perspectives to identify the key factors influencing voter support for political parties. By utilizing various machine learning techniques, the goal is to build a classification model that can accurately predict election outcomes for exit poll purposes. Additionally, identifying these key factors will enhance the quality of data collected during surveys by focusing on important features, ultimately improving the predictive model's accuracy.

Model Comparison

We created 6 models using different techniques compared each model's performance for test and train data using key metrics and found that all the models are stable, here we will compare these models with each other to find the best model based on their AUC score for test data.

	Train data score	Test data score	AUC score train data	AUC score test data
Random Forest Model	0.880975	0.820961	0.940641	0.884118
Gradient Boosting Model	0.869728	0.818777	0.925820	0.880937
ADA Boosting Model	0.858482	0.812227	0.917216	0.875361
Bagging Model	0.821931	0.818777	0.890419	0.871394
Naive Bayes Model	0.837863	0.814410	0.891535	0.866767
KNN Model	0.859419	0.772926	0.929247	0.832354

Table 1: Model Comparison

Based on the above table, we can conclude that when comparing models by AUC score for the test set, random forest model is performing the best with an AUC score of 0.8841. This indicates that the random forest model has a superior ability to distinguish between classes, making it the most effective model for our classification task. We will check for the most important features which play crucial role in distinguishing between classes.

Important Features

	imp
Hague	0.291610
Blair	0.235054
Europe	0.191686
Political.knowledge	0.105977
Age	0.081862
Economic.cond.national	0.061651
Economic.cond.household	0.024827
Gender	0.007334

Table 2: Important Features

The above table indicates that the variable Hague is the most influential feature in the model, with a relative importance score of 0.291610. Other significant features include Blair and Europe, with importance scores of 0.235054 and 0.191686, respectively. In contrast, Gender has the least impact on the model, with an importance score of 0.007334.

Conclusion

1. For the current election, the most important factor is the candidate where a voter is more likely to vote for the candidate for whom they have a favorable view. In fact, just by taking columns 'Hague' and 'Blair' we can build a model with an accuracy of 0.7947 and AUC score of 0.836 which is only couple of points lower than the overall model scores.
2. On comparing different models by their AUC scores, we find that Random Forest model is performing the best with a score of 0.8841 on test data meaning there is 88.41% chance that the model will correctly distinguish a randomly chosen instance. We have used AUC score as a key metric to select the best performing model because AUC score is a robust measure for skewed data as it accounts for both the true positive rate and the false positive rate. It provides a single metric that summarizes the model's ability to distinguish between classes across all thresholds.
3. Key takeaways from EDA:

- Based on the EDA, we found that voters had a clear preference for candidates, as reflected in the ratings they gave to both candidates. This clarity significantly aided in prediction, with 'Hague' and 'Blair' emerging as the most influential features impacting the outcome.
- Another important factor was Europe where from EDA we found that voters who are more Eurosceptic are more likely to vote for the Conservative party.
- The median age of voters for Conservative party is higher than that for Labour party meaning older people are more likely to vote for the Conservative party.

4. While we successfully built and selected the best-performing predictive model with approximately 82% accuracy and an AUC score of 88%, there is potential for further improvement. By expanding the sample to include more features that clearly capture a voter's preference, similar to the features 'Hague' and 'Blair', we can enhance the model's performance. Additionally, recording which factors are most important to voters such as candidate preference, Europe, or economic factors would allow us to assign weights to each factor, thereby refining our model further.

Problem 2

Business Context

To analyze the speeches of three different Presidents of the United States and find the most frequent words used which might give us the understanding of their priorities and provide some insight into their policy making.

Problem Statement

The objective of this analysis is to study speeches for three different Presidents of the United States belonging from three different decades and find the most common words used by each of them in these speeches.

Most Common Words

On doing a frequency count for words in each speech we found the most common words in each speech.

- Top 3 words used by President Roosevelt in speech of 1941
  [('nation', 17), ('know', 10), ('peopl', 9)]
- Top 3 words used by President Kennedy in speech of 1961
  [('let', 16), ('us', 12), ('power', 9)]
- Top 3 words used by President Nixon in speech of 1973
  [('us', 26), ('let', 22), ('america', 21)]

Word Cloud

We analyzed the text of each speech and presented the most common words in the form of a word cloud.

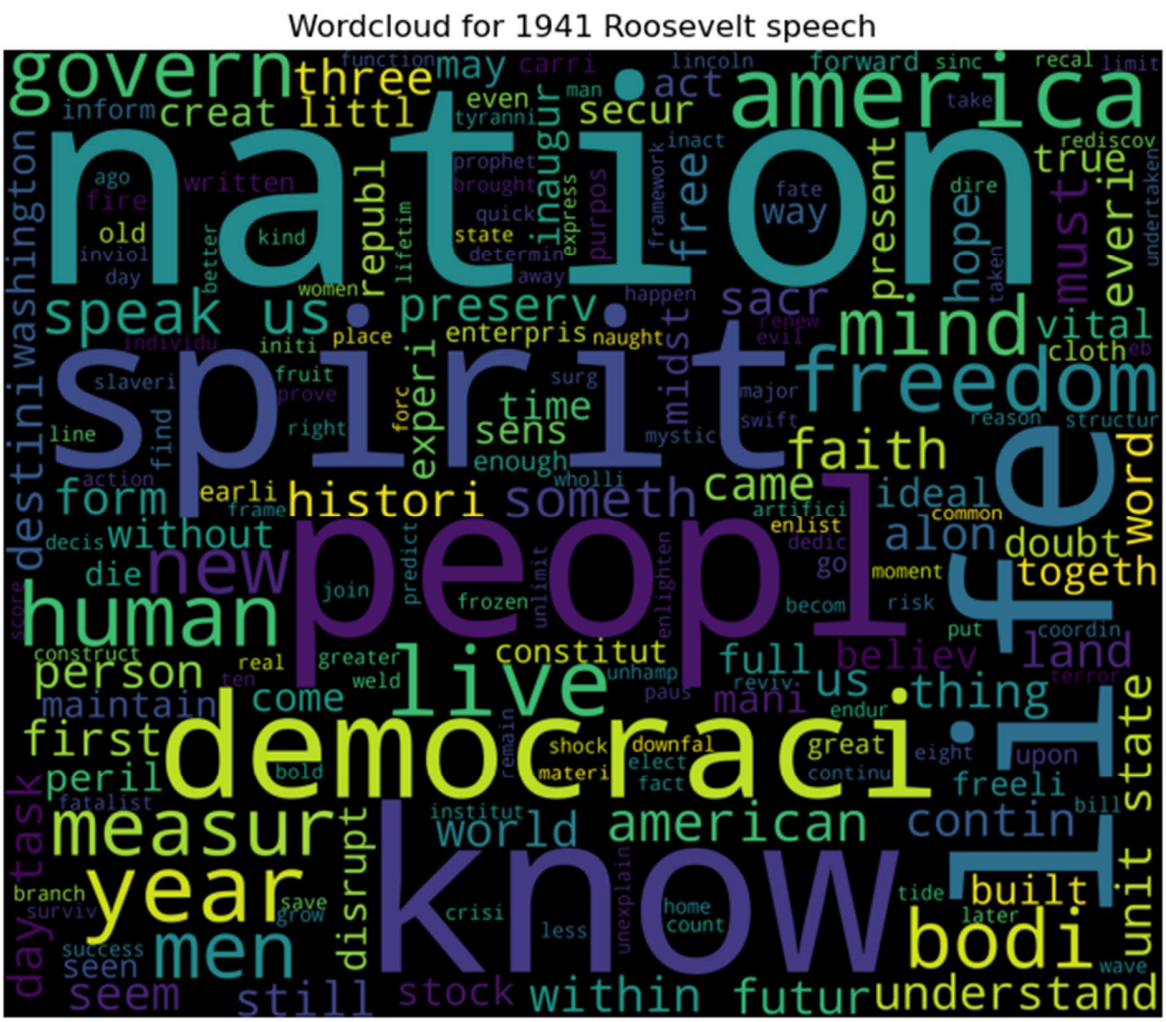


Figure 1: Word cloud

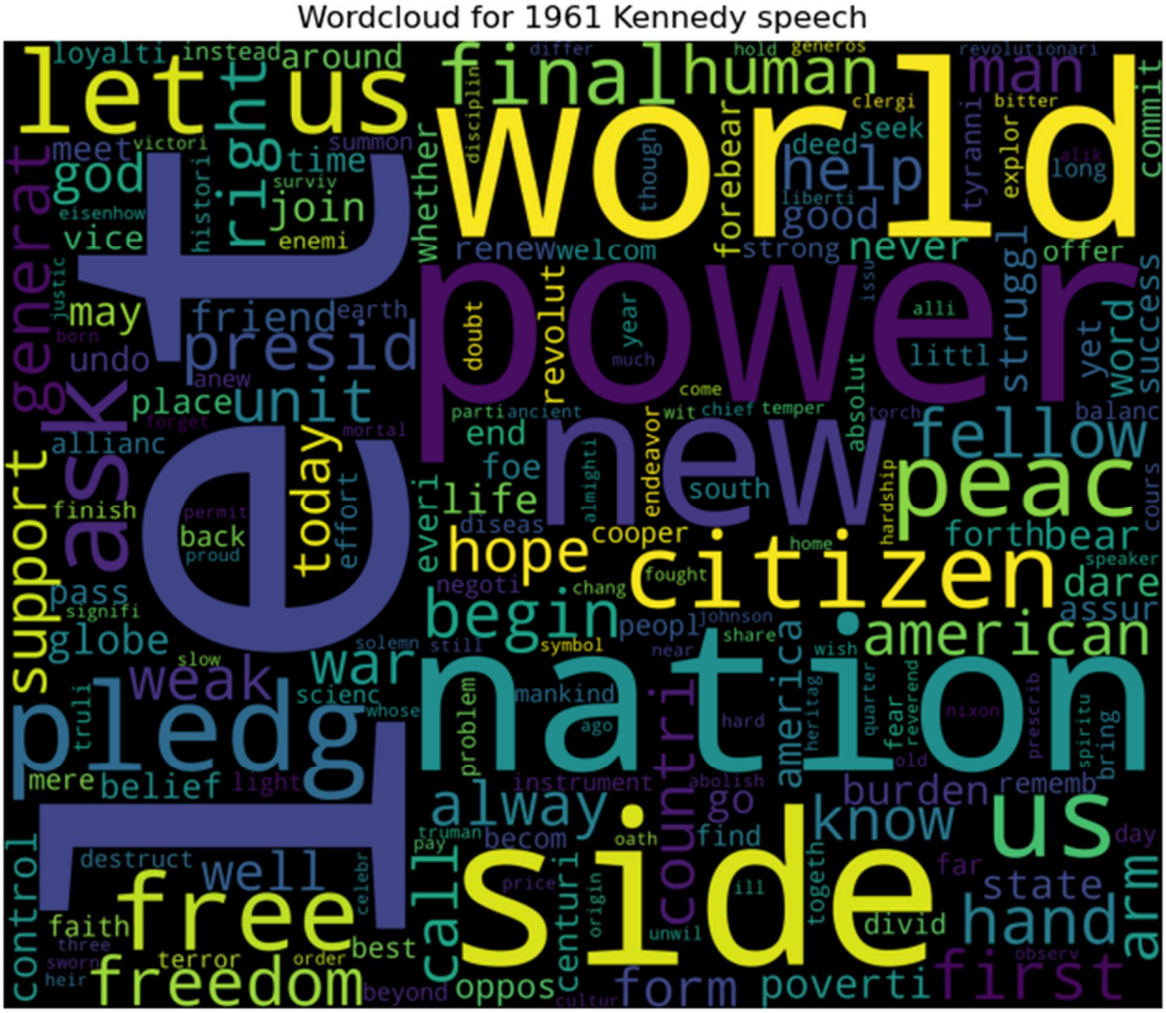


Figure 2: Word cloud

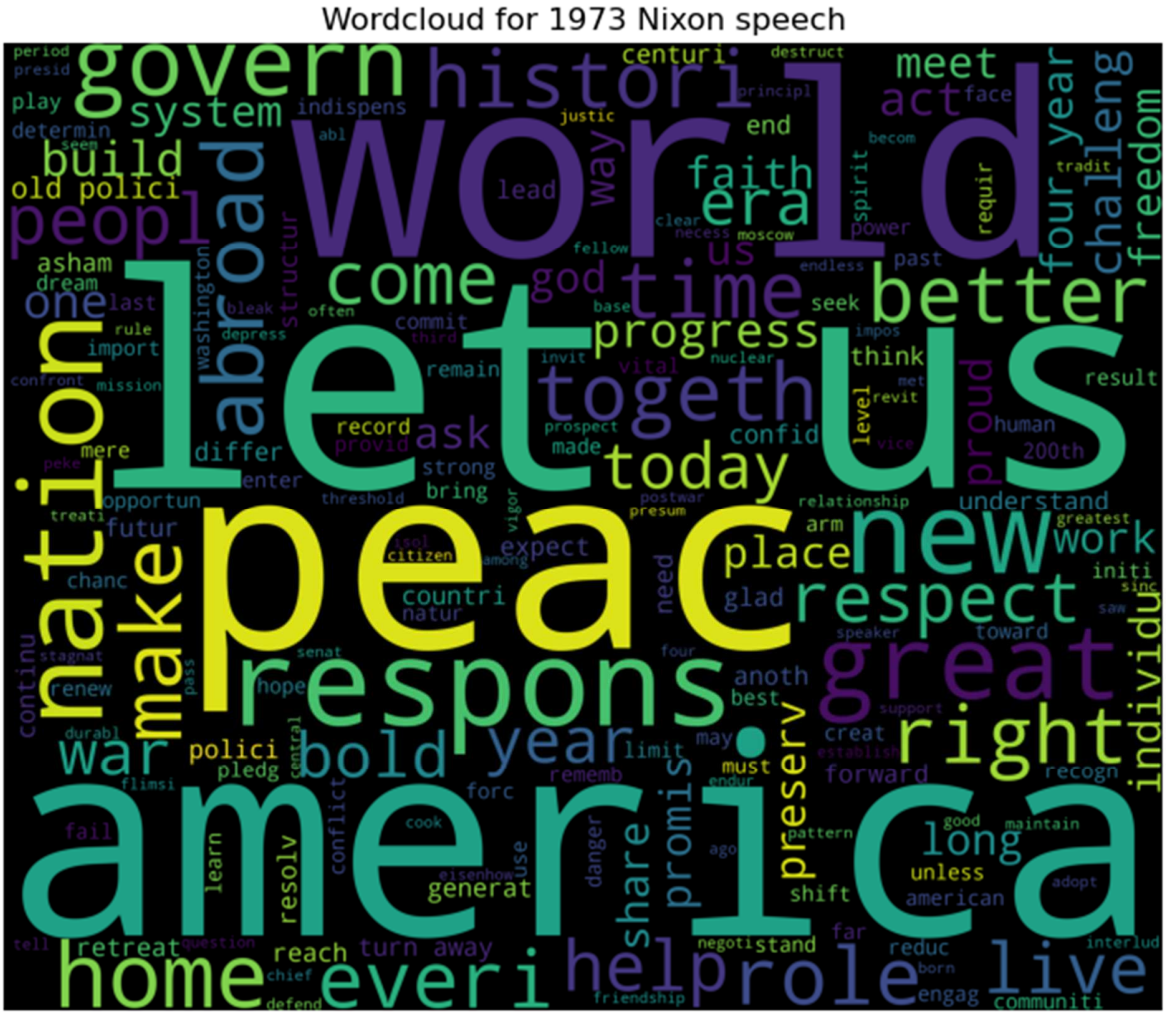


Figure 3: Word cloud

Key Takeaway

The word clouds generated for this text analysis project provide valuable insights into the evolving priorities of U.S. presidents across different decades. In 1941, President Roosevelt's speeches prominently featured words such as 'Nation,' 'Democracy,' and 'Freedom,' reflecting the focus on national unity and democratic values during that era. By 1961, under President Kennedy, the emphasis shifted towards 'World' and 'Power,' indicating a broader international perspective and the geopolitical dynamics of the Cold War. By 1971, President Nixon's speeches highlighted words like 'Peace' and 'Response,' signalling a focus on achieving peace and addressing immediate national and international challenges. These changes in word usage underscore how presidential rhetoric adapts to the prevailing political, social, and global contexts of their respective times. and can be extremely helpful in predicting countries future policies and future course of action.

Problem 1

1.1 Background Information

In the news media industry, providing insightful election coverage is critical for attracting and retaining viewership. To achieve this, data-driven analysis is essential. CNBE News, a prominent news channel, has conducted a comprehensive survey capturing the perspectives of 1,525 voters across various demographic and socio-economic factors. Using this data, they aim to forecast which political party a voter is likely to support. This analysis will serve as the foundation for creating an exit poll that can accurately predict the overall election outcomes.

1.2 Business Objective

Exit polls are integral to election coverage, as accurate predictions can significantly benefit a news channel. With this in mind, CNBE News, a prominent news channel, aims to build a predictive model to forecast which political party a voter is likely to support. They will use survey data that captures voter perspectives across various demographic and socio-economic factors.

1.3 Problem Statement

The objective of this project is to conduct a comprehensive analysis of survey data that captures voter perspectives to identify the key factors influencing voter support for political parties. By utilizing various machine learning techniques, the goal is to build a classification model that can accurately predict election outcomes for exit poll purposes. Additionally, identifying these key factors will enhance the quality of data collected during surveys by focusing on important features, ultimately improving the predictive model's accuracy.

1.4 METHODOLOGY

Import the libraries - Load the data - Check the structure of the data - Check the types of the data – Check for and treat (if needed) missing values - Check the statistical summary – Check for and treat (if needed) Data Irregularities – Univariate Analysis – Bivariate Analysis – Data Encoding – Data Splitting – Apply Classification Models – Predict values – Evaluate model – Compare model – Get Important Features – Conclusion

Key Points

1. **Data Collection**: For data, a survey was carried out by CNBE news which captured the perspectives of 1525 voters across various demographic and socio-economic factors.
2. **Data Cleaning and Pre-processing:** Dataset was checked for duplicates, missing values, bad data and outliers. An irrelevant column named ‘Unnamed: 0’ was found in the dataset which was

dropped and attribute names were found to follow inconsistent nomenclature which were made consistent by renaming relevant columns.

3. **Univariate Analysis:** Individual variables were analyzed using boxplot and histogram to understand distribution, central tendency and variability of variables.
4. **Bivariate Analysis:** All the variables were examined with the aim of gaining deeper insights about voters' perception.
5. **Visualization Techniques:** In the report we have used histograms, boxplot and count plot for univariate analysis, in bivariate analysis, to understand correlation between numeric variables heatmap and pair plot are used, violin plot is used to understand relationship between categorical and numeric variables.
6. **Tools and Software:** We have carried out the analysis using programming language python on Jupyter notebook. For this analysis Python libraries Numpy, Pandas, Matplotlib, Seaborn, Scikit-learn and Math were used.

1.5 Data Overview

1. **Data Description:** Dataset has 1525 rows and 10 columns.

```
shape of the dataset
--------------------------------------------------------------------------------

(1525, 10)
```

Table 3: Dataset Shape

2. **Dataset Information:** Of the 9 columns in the dataset, 2 are object type and 7 are int 64 type.

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

Table 4: Dataset Information

3. **Missing Value Check:** There were no missing values in the dataset.

```
missing values
--------------------------------------------------------------------------------

Unnamed: 0                  0
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 5: Missing values information

4. **Duplicate Values:** Data was checked for duplicate values and no duplicates were found

```
checking for duplicates
--------------------------------------------------------------------------------
number of dupliacte rows: 0
```

Table 6: Data Duplicates

5. **Statistical Summary:**

```
statistical summary
--------------------------------------------------------------------------------
```

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	1525.0	763.000000	440.373894	1.0	382.0	763.0	1144.0	1525.0
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 7: Statistical Summary

6. **Frequency Distribution of Categorical Columns:**

```
value counts for vote
--------------------------------------------------------------------------------
vote
Labour          1063
Conservative     462
Name: count, dtype: int64
```

```
value counts for gender
--------------------------------------------------------------------------------
gender
female    812
male      713
Name: count, dtype: int64
```

Table 8: Frequency Distribution of categorical columns

Key observations

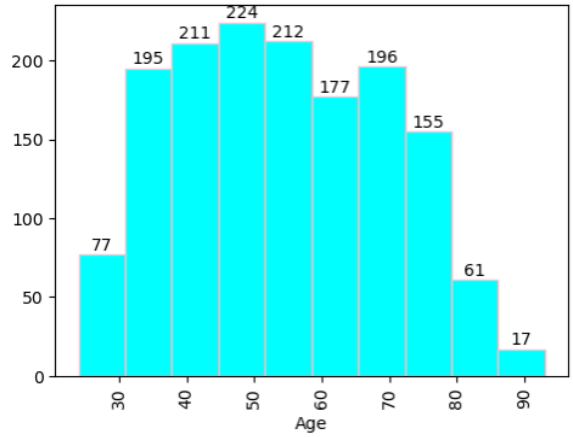
1. Dataset has 1525 rows and 10 columns, out of these columns, first column 'Unnamed: 0' contains index numbers which are irrelevant for us, we will drop this column during pre-processing stage.
2. Dataset has 9 relevant attributes out of which 7 have numeric data and 2 have object type. In the data, attribute 'vote' is the target label, additionally, feature names have inconsistency in the nomenclature as for some features first character is in upper case while for few it is in lower case, we will convert the first character to upper case for all features.
3. From the statistical summary of numeric columns, we can conclude that there are no anomalies in data. Except for age all the other features are ordinal categoric in nature, we will convert them to categoric datatype from numeric during at the time of encoding.
4. Datatype for all features is as per our expectation and there are no missing values or duplicates in the data.
5. In the dataset 1063 people have voted for Labour party while only 462 have voted for conservative party which means that data is skewed, however, since the minority class accounts for over 30% in the data, we will consider it balanced for our project.

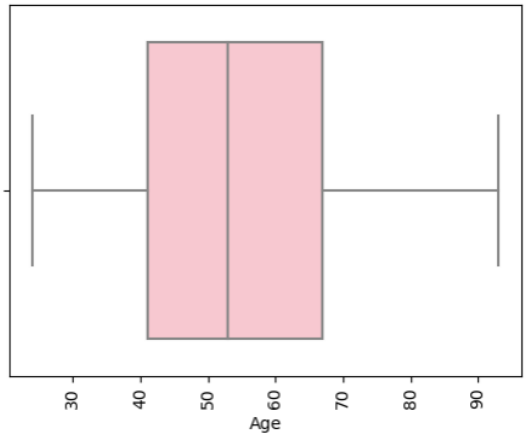
1.6 Exploratory Data Analysis

1.6.1 Univariate Analysis

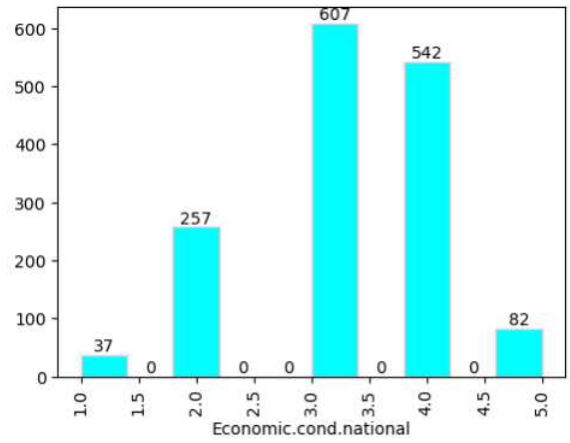
For numeric columns

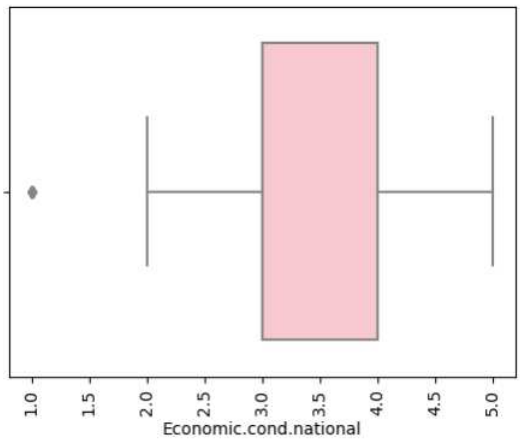
Skewness of Age: 0.14462077228942483
Distribution of Age



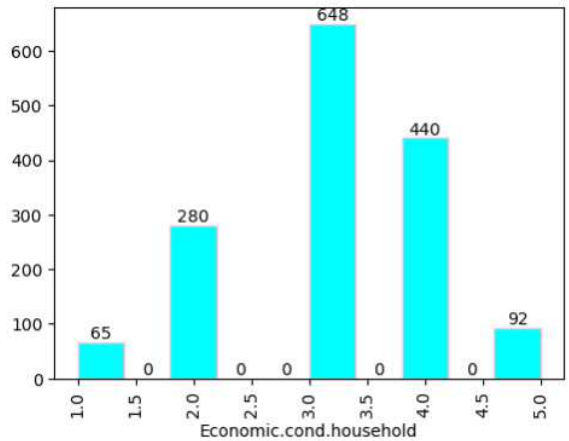


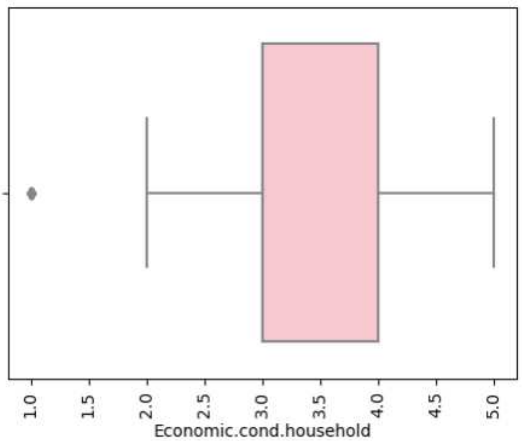
Skewness of Economic.cond.national: -0.2404528899412957
Distribution of Economic.cond.national



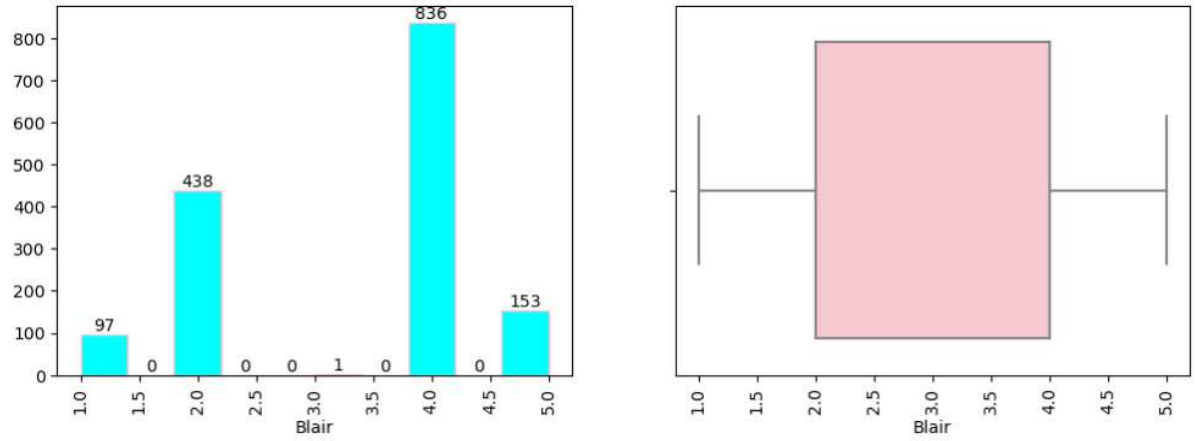


Skewness of Economic.cond.household: -0.14955204997804528
Distribution of Economic.cond.household

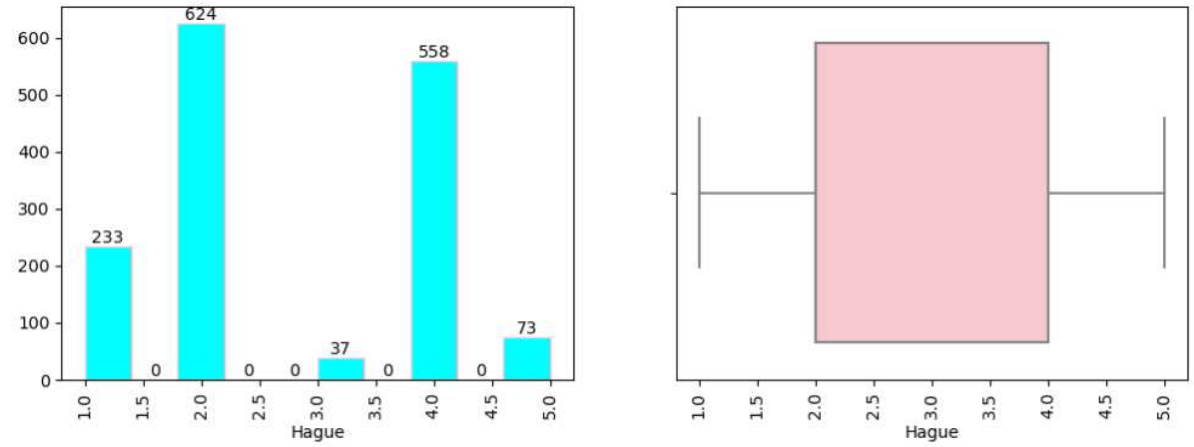




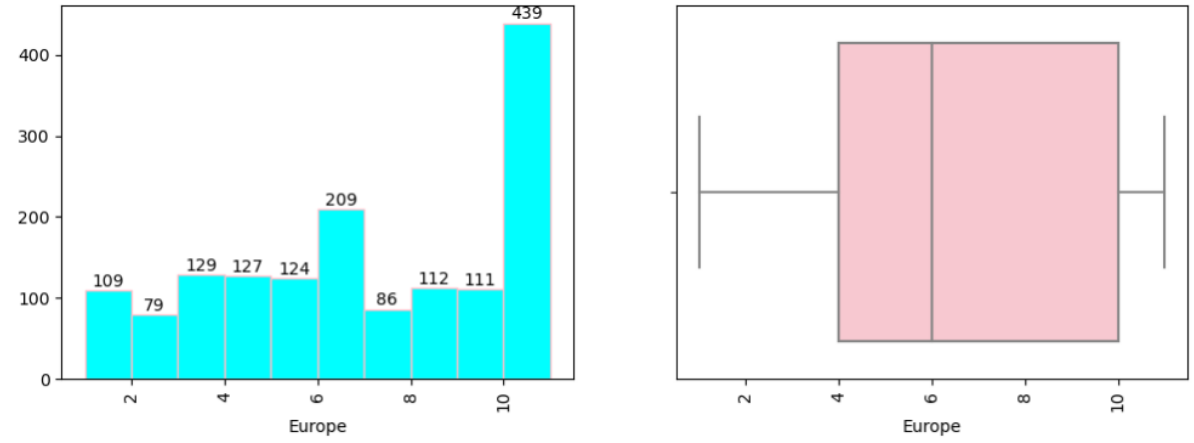
Skewness of Blair: -0.5354186518673825
Distribution of Blair



Skewness of Hague: 0.1520996272526911
Distribution of Hague



Skewness of Europe: -0.13594670991422228
Distribution of Europe



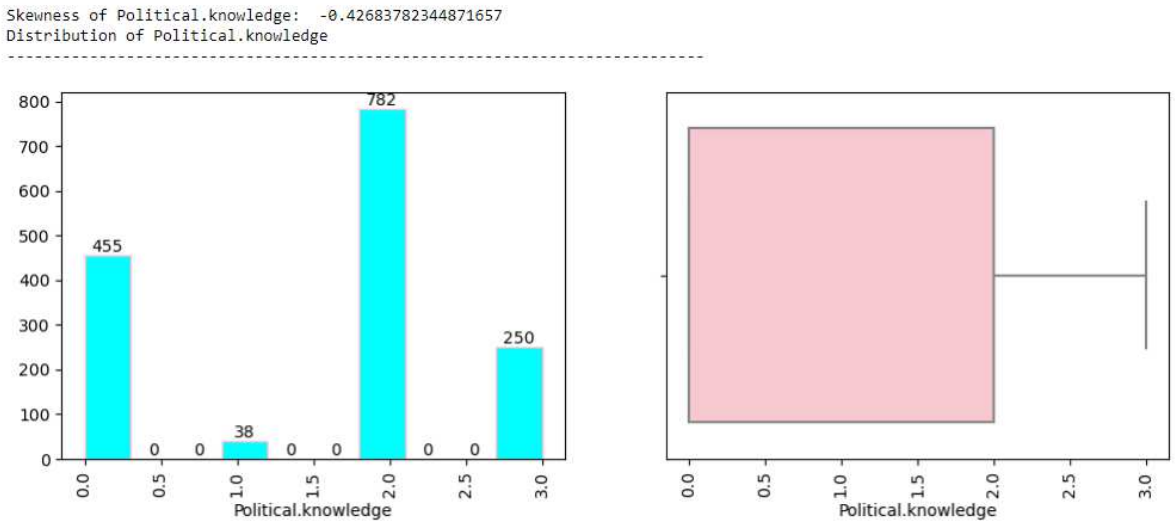
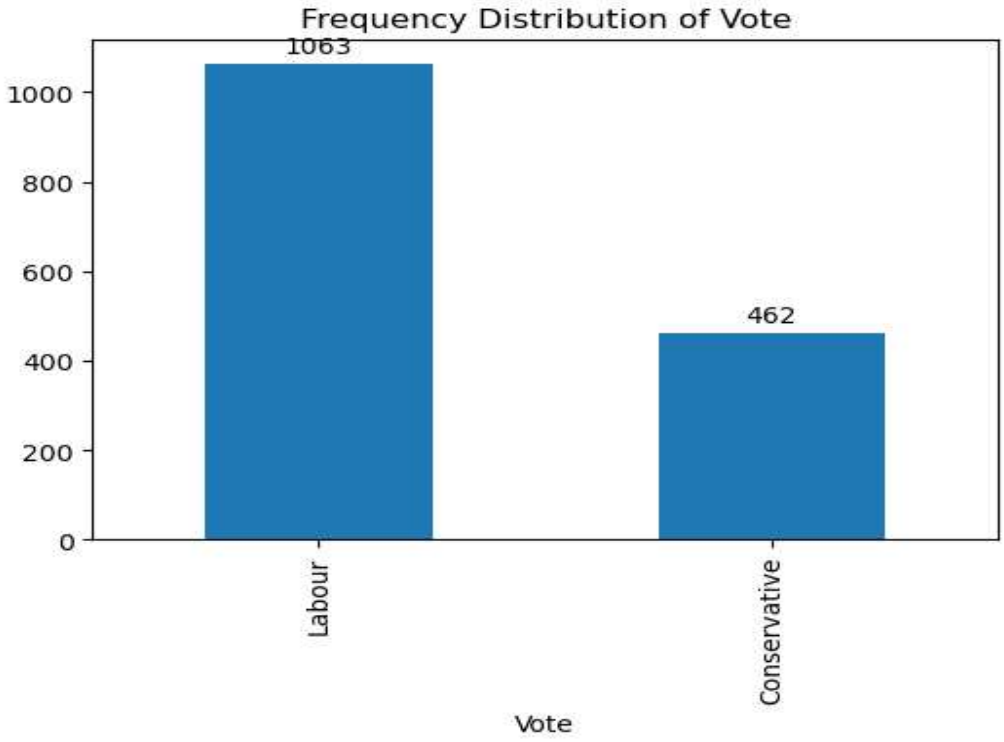


Figure 4: Univariate Analysis numeric columns

Key Observations

1. Data in 'Age' attribute is normally distributed, there are no outliers in it which means that outlier treatment is not required since other numeric attributes are categorical in nature.
2. As per the demographic information, most respondents covered are aged between 30 and 80 years, meaning our data sample covers largely middle aged or senior citizens.
3. Most respondents have given 3 or 4 rating for national and their own household economic condition which means they believe that national as well as their own household economic condition is anywhere between average and good.
4. For attributes 'Blair' and 'Hague' almost all ratings are either 1, 2 or 4, 5, where these ratings represent how each respondent perceive the candidates belonging to both the parties with ratings of 4 and 5 means they have favorable view of that candidate and conversely ratings of 1 and 2 means they have unfavorable view. Here when we check the rating count for both the candidates, we can infer that almost 1000 respondents have favorable view for the Labour party leader Tony Blair and over 600 respondents have favorable view for Conservative party leader William Hague, as per value count of 'Vote' attribute, for 1063 respondents, their party of choice is Labour party and for 462 it is Conservative party. All these attributes are showing similar trends meaning attributes 'Blair' and 'Hague' might be highly correlated to each other as well as target label 'Vote'.
5. Most respondents have rated in excess of 6 for attribute 'Europe' which means that most of them are strongly against UK becoming part of the European Union.
6. While most respondents have moderate or high level of knowledge on stance of political parties on European integration, there are significant number of respondents who do not have any knowledge in this matter.

For categorical columns



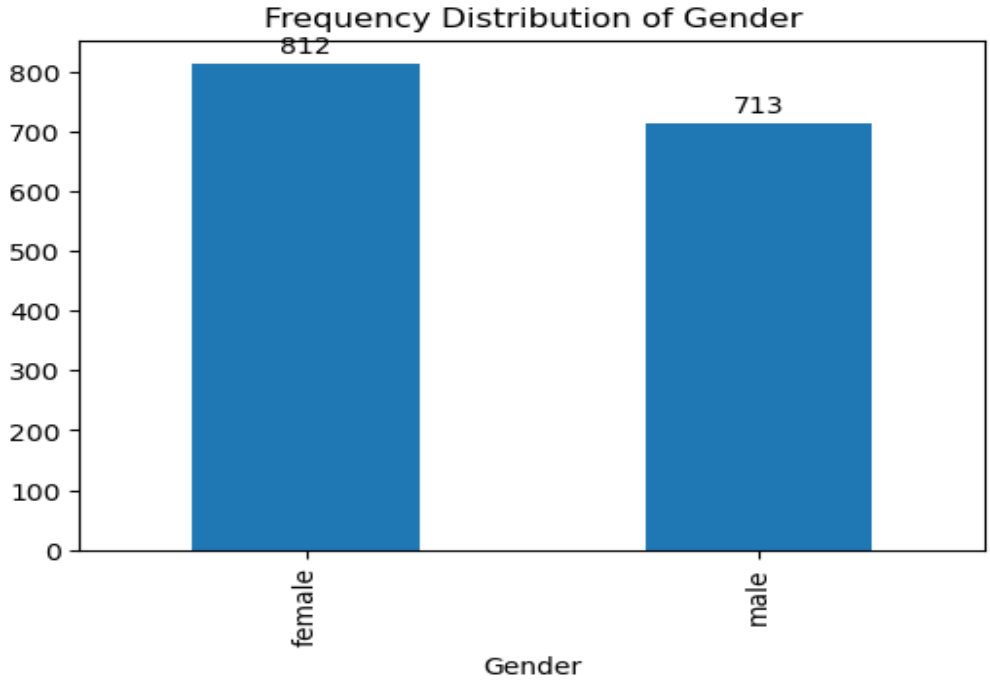


Figure 5: Univariate Analysis categorical columns

1.6.2 Bivariate Analysis

Relation between numeric columns

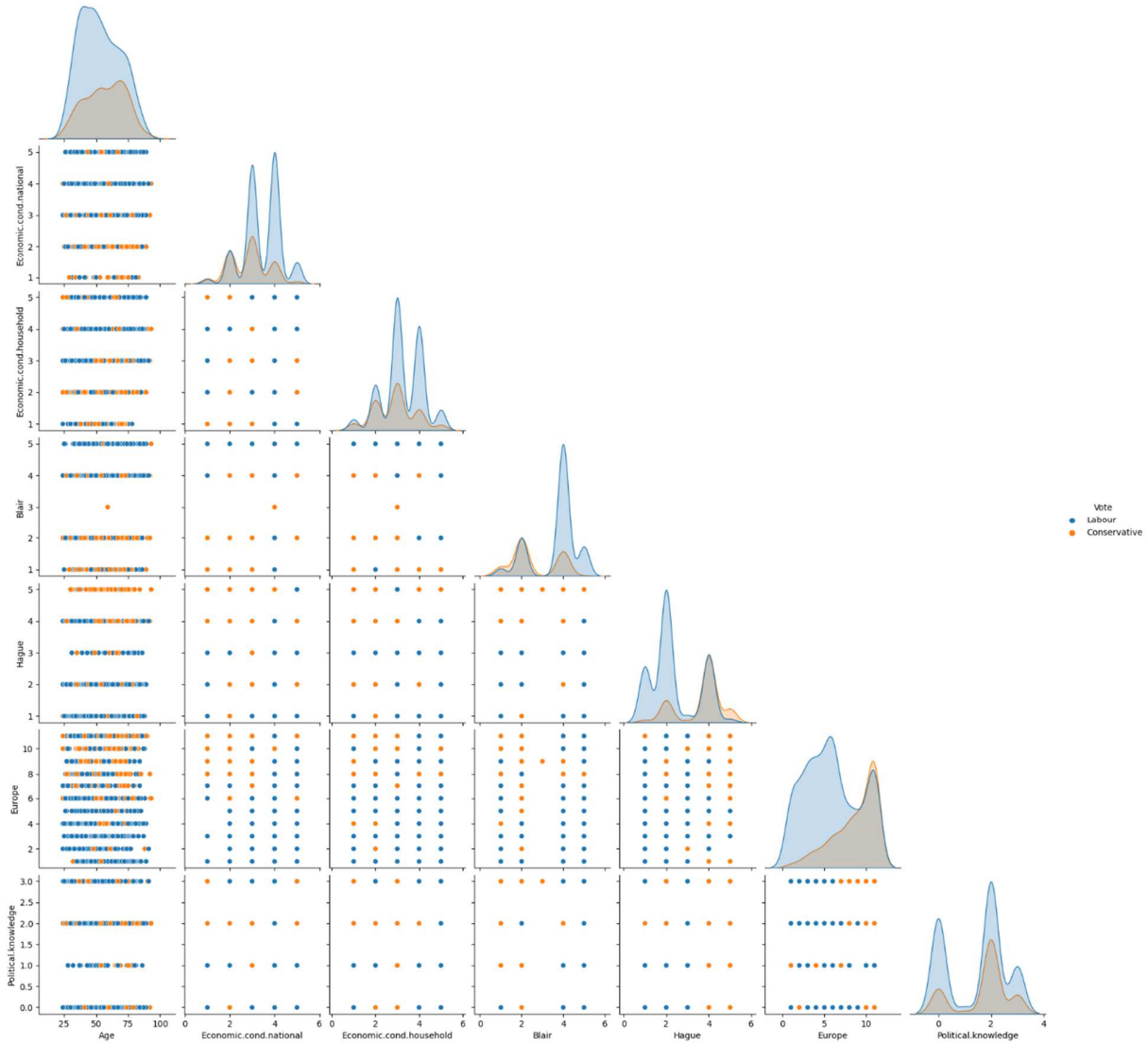


Figure 6: Pair plot

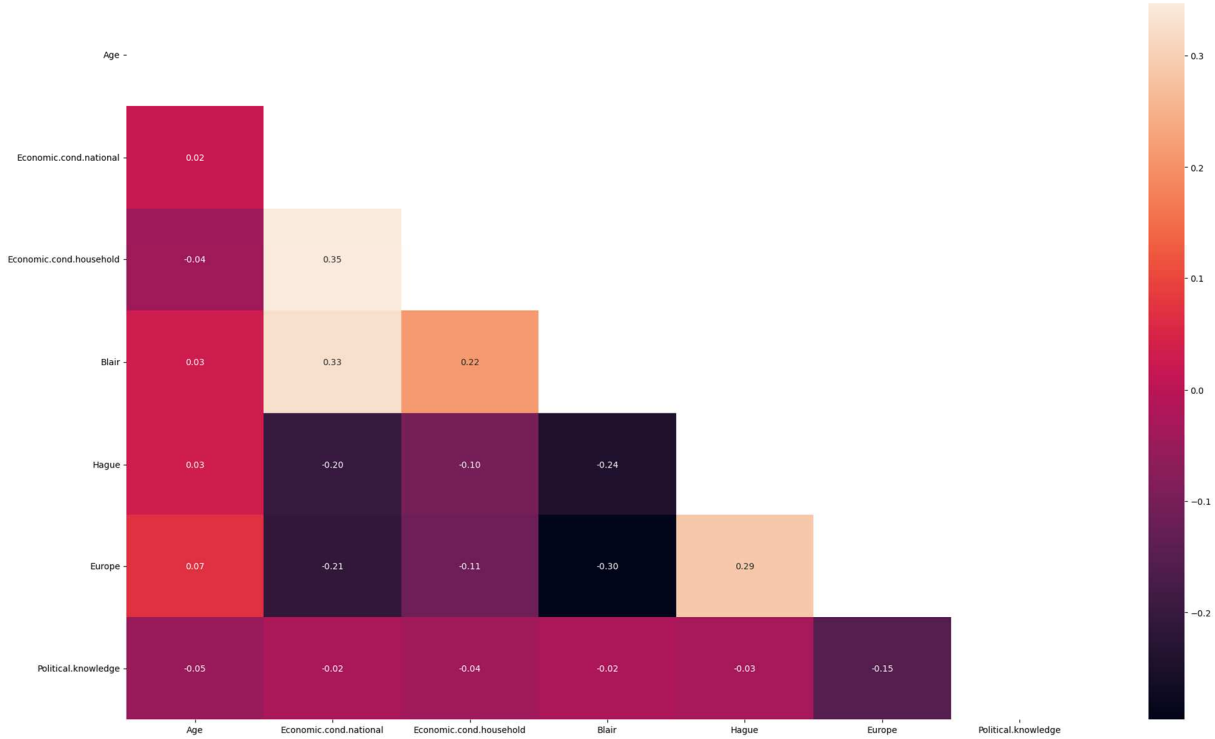


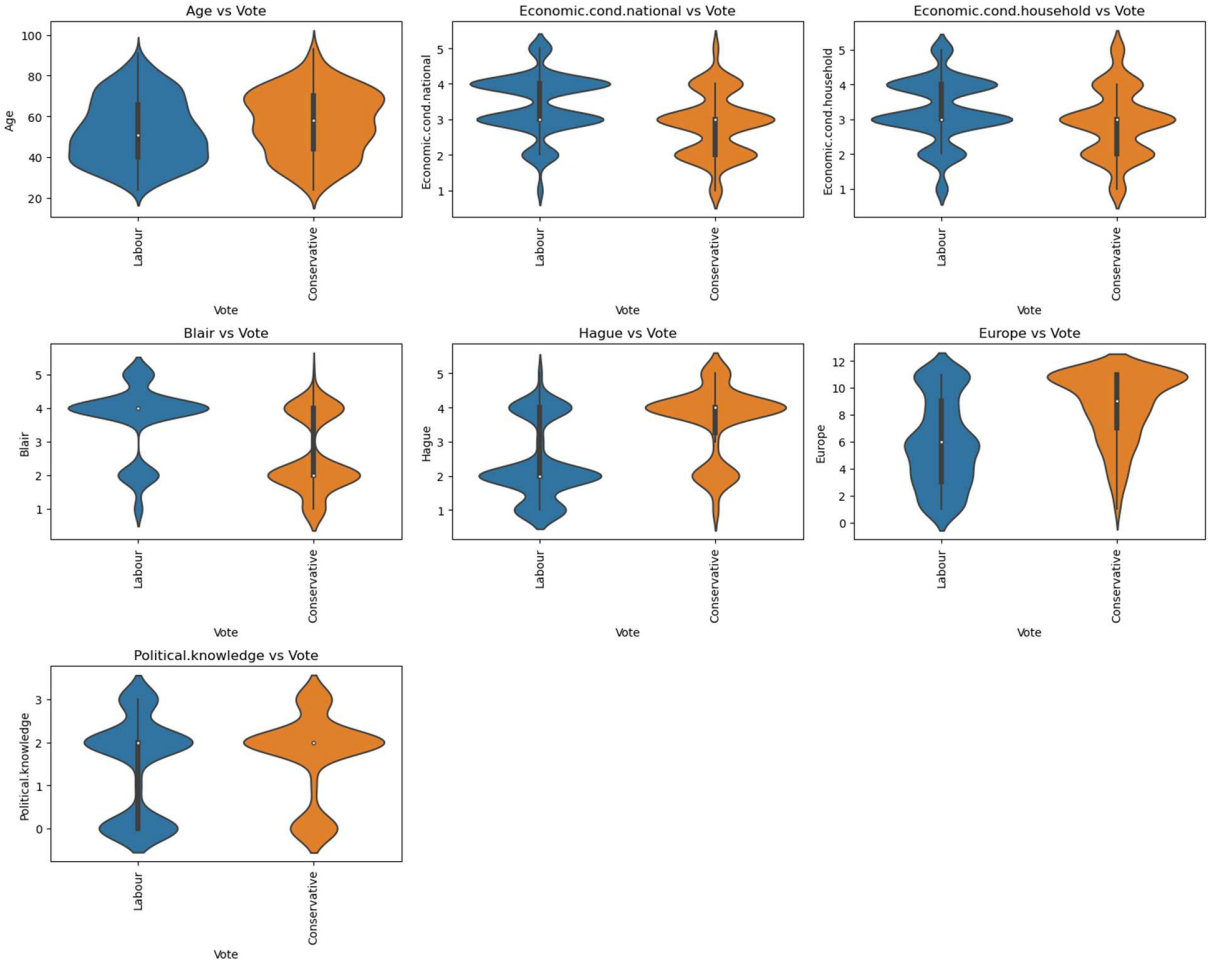
Figure 7: Heatmap

Key Observations

1. From the pair plot we can conclude that distribution for both the class labels overlap for all the features which means that these features are weak predictors.
2. There is no significant correlation between the numeric attributes.

Relation between numeric and categorical columns

bivariate analysis for Vote



bivariate analysis for Gender

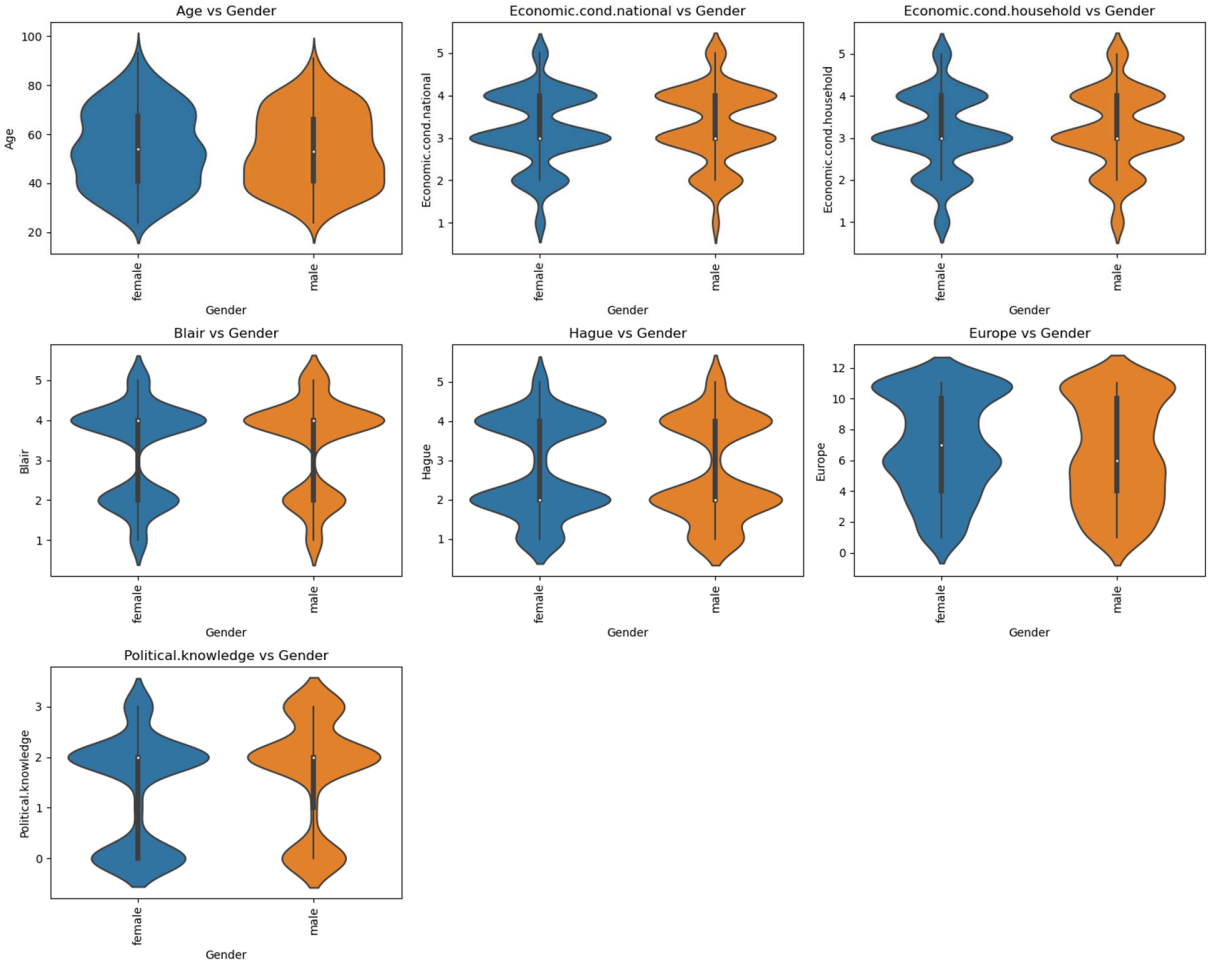


Figure 8: Bivariate analysis

Key Observations

1. From the bivariate analysis of target label 'Vote' with all the numeric attributes we can infer:

- Younger respondents prefer Labour party more while older respondents prefer Conservative party.
- As we had anticipated earlier that attributes 'Blair' and 'Hague' might have correlation with 'Vote', from the plot we can comprehend that our hypothesis was correct as respondents who have given higher ratings to Tony Blair are more likely to Labour party and vice versa for conservative party.
- Another key differentiator is 'Europe' where people who have stronger Eurosceptic sentiments are more likely to vote for Conservative party and those less averse to Europe integration, Labour party.

2. In bivariate analysis of 'Gender' with numeric features, while the preference for both male and female are similar for all the attributes there is variance in case of 'Europe' where from the plot we can conclude that male are comparatively less averse to Europe integration.

1.7 Data Encoding

For data encoding we have converted the numeric ordinal features to category type while keeping their order intact and converted features with object data into numeric using Label Encoder.

Glimpse of encoded data

	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 9: Data Overview

Based on first 5 rows of the data we can clearly see that the entire data is in numeric form.

1.8 Splitting Data

Here data is divided into X and Y where X contains all the independent attributes and Y has response variable. This X and Y are further split into train and test data where for this problem we have taken train to test split ratio of 70:30.

Train data

	Age	Economic.cond.national	Economic.cond.household	Blair	Hague	Europe	Political.knowledge	Gender
1372	74	4	4	4	2	2	2	1
126	46	4	3	4	4	2	2	1
327	77	3	3	2	4	9	2	1
292	51	3	3	4	2	6	1	0
1058	37	3	4	4	2	8	0	0

Table 10: Data Overview

Test data

	Age	Economic.cond.national	Economic.cond.household	Blair	Hague	Europe	Political.knowledge	Gender
782	35	4	4	5	2	6	2	1
76	42	4	3	4	2	4	2	0
1009	32	4	3	4	2	4	1	0
1403	48	3	3	2	4	2	2	0
846	35	3	4	2	1	11	2	0

Table 11: Data Overview

1.9 Classification Modelling

We will build models using different classification techniques namely Naive Bayes and KNN and then we will try to improve the model performance using different ensemble techniques. We will compare different model performances using their AUC (Area Under the ROC Curve) score. The AUC score is a robust measure for skewed data as it accounts for both the true positive rate and the false positive rate. It provides a single metric that summarizes the model's ability to distinguish between classes across all thresholds. Additionally, plotting the ROC curve helps visually evaluate the model's discriminative power.

For evaluation of each model, we will additionally be using classification table and confusion matrix as a classification report provides a detailed summary of key metrics like precision, recall, F1 score, and support for each class, helping to evaluate the performance of a model comprehensively. A confusion matrix offers a visual and numerical breakdown of true positives, false positives, true negatives, and false negatives, allowing for an in-depth understanding of the model's accuracy and error types.

Naïve Bayes Model

A classification model was created using GaussianNB from naïve Bayes in scikit-learn library whose accuracy on train and test data were:

Model accuracy for train data
0.8379

Model accuracy for test data
0.8144

Accuracy score for both test and train data are almost identical which means we have a stable model.

Model Evaluation

Using AUC-ROC Method

for training data
AUC Score: 0.892

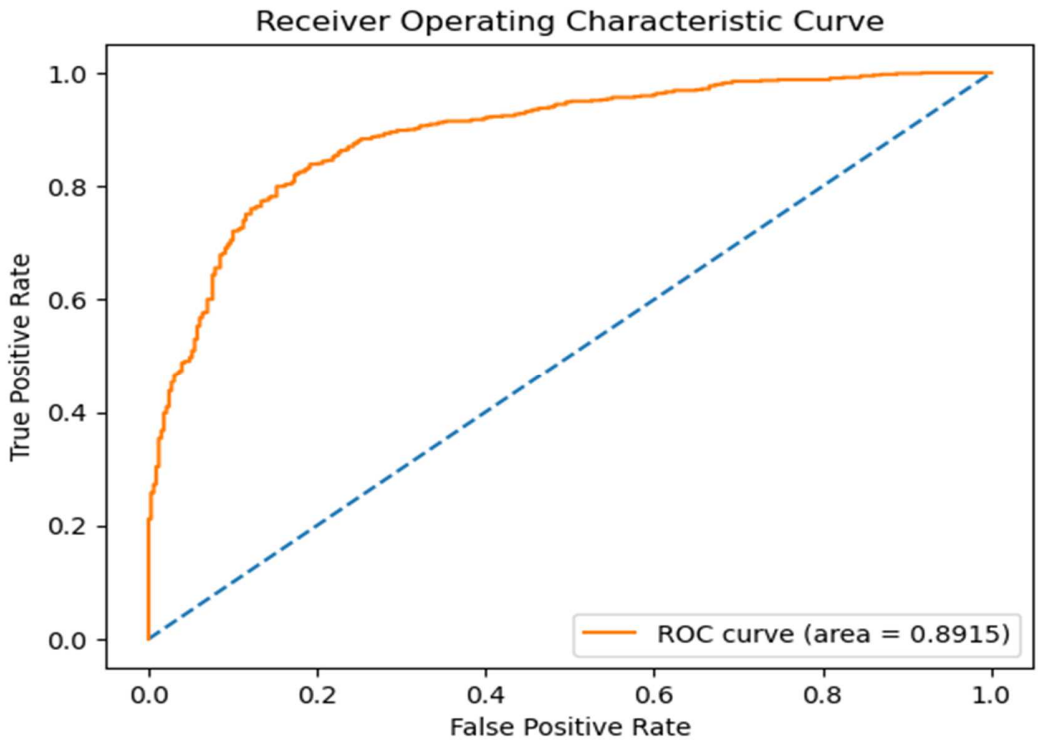


Figure 9: AUC-ROC curve

for test data

AUC Score: 0.867

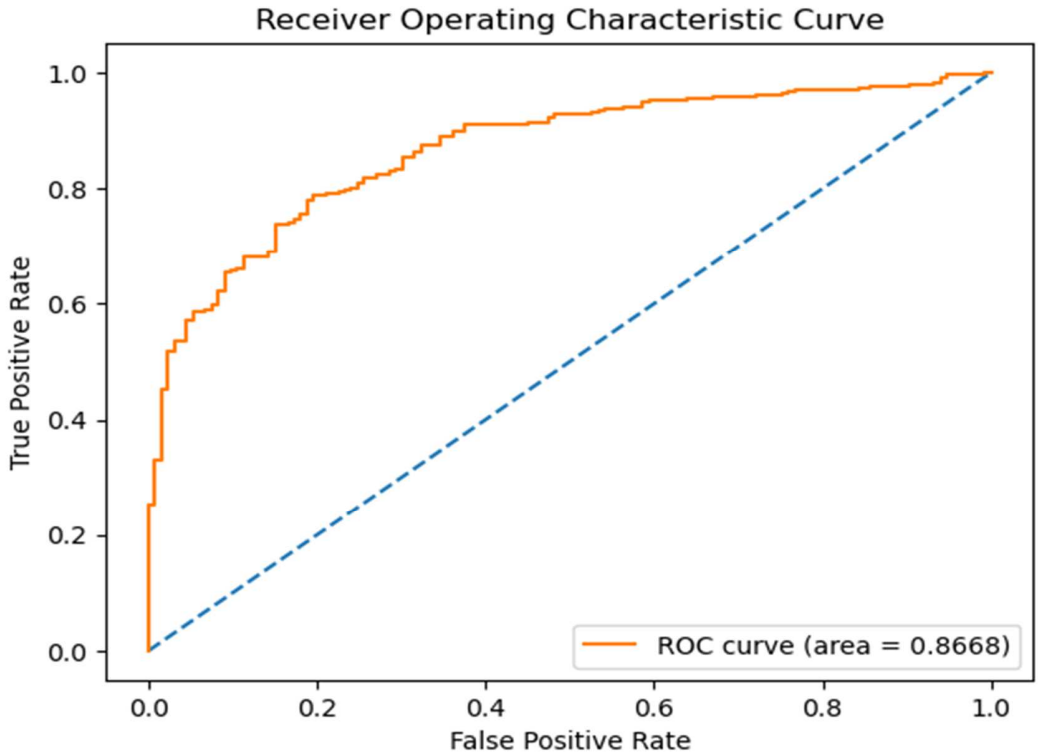


Figure 10: AUC-ROC curve

Confusion Matrix

for training data

array ([[237, 92],
       [81, 657]], dtype=int64)

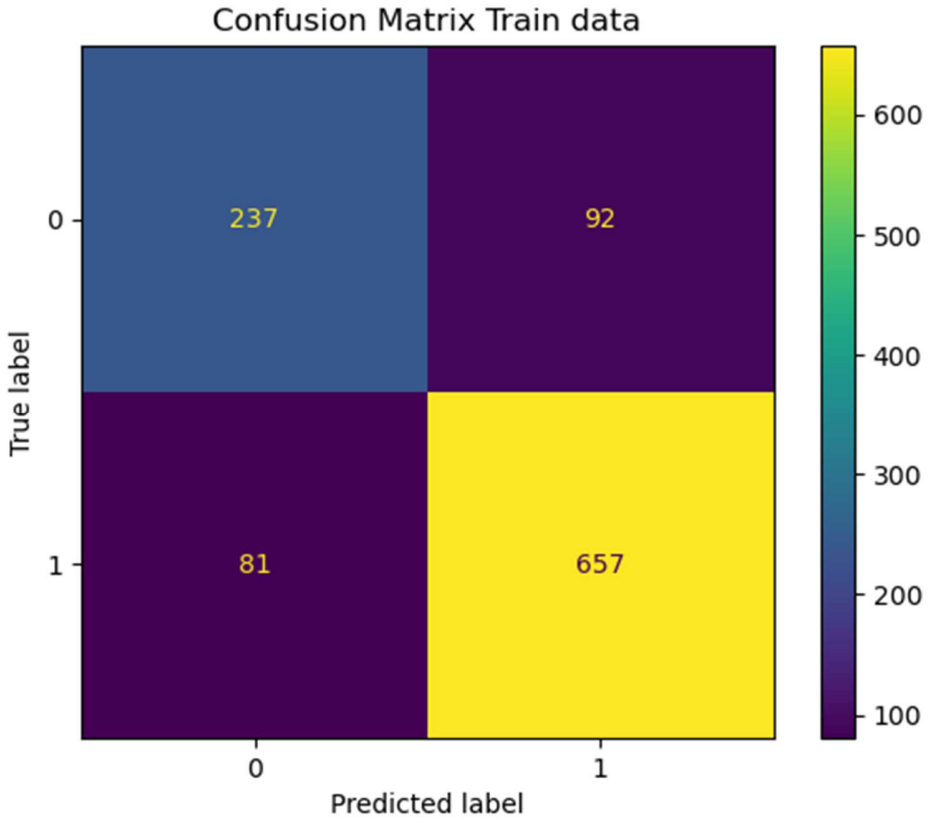


Figure 11: Confusion matrix

Classification report Train data

	precision	recall	f1-score	support
0	0.75	0.72	0.73	329
1	0.88	0.89	0.88	738
accuracy			0.84	1067
macro avg	0.81	0.81	0.81	1067
weighted avg	0.84	0.84	0.84	1067

Table 12: Classification report

for test data

array ([[87, 46],
       [39, 286]], dtype=int64)

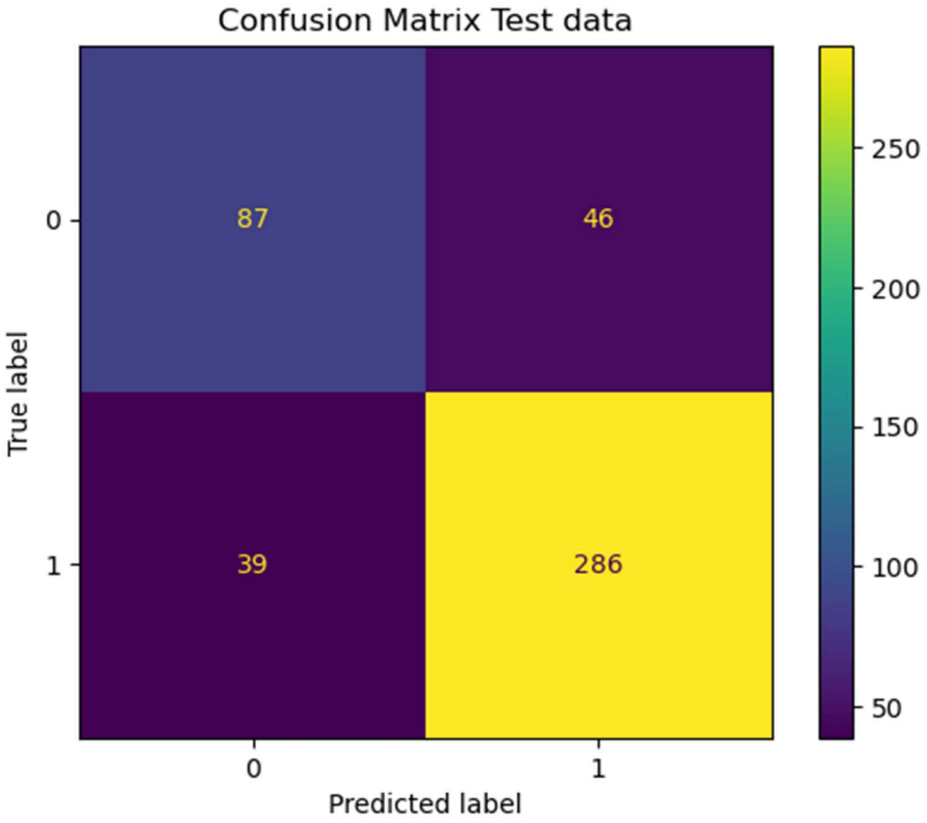


Figure 12: Confusion Matrix

Classification report Test data

	precision	recall	f1-score	support
0	0.69	0.65	0.67	133
1	0.86	0.88	0.87	325
accuracy			0.81	458
macro avg	0.78	0.77	0.77	458
weighted avg	0.81	0.81	0.81	458

Table 13: Classification report

The naive bayes model has on test data an AUC score of 0.8668 meaning it has high performance and there is 86.68% chance that the model will correctly distinguish a randomly chosen instance, however, f1-score for class 0 (Conservative party) is only 0.67 which means model is performing weakly when classifying for minority class.

KNN Model

We built a classification model using KNeighborsClassifier from neighbours module in scikit-learn library whose accuracy on train and test data are:

Model accuracy for train data
0.8594

Model accuracy for test data
0.7729

Accuracy score for both train and test are with range of 10% which means we have a stable model.

Model Evaluation

Using AUC-ROC Method

for training data
AUC Score: 0.929

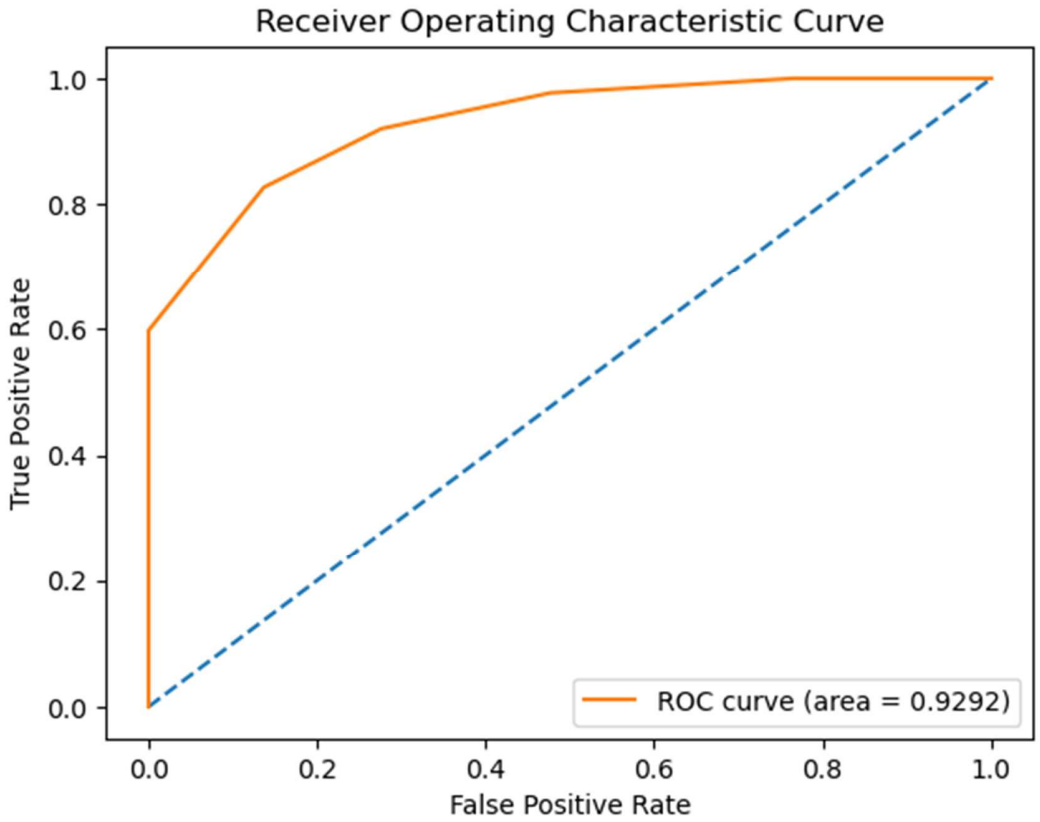


Figure 13: AUC-ROC curve

for test data
AUC Score: 0.832

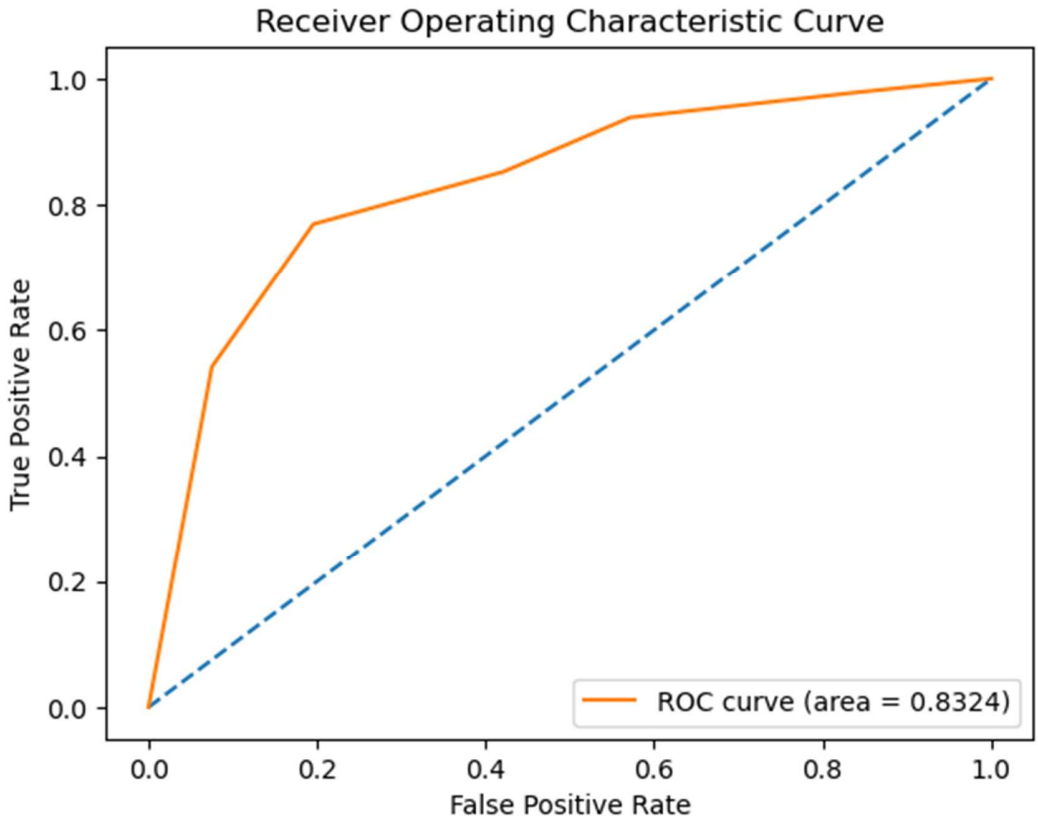


Figure 14: AUC-ROC curve

Confusion Matrix

for training data

array ([[238, 91],
       [59, 679]], dtype=int64)

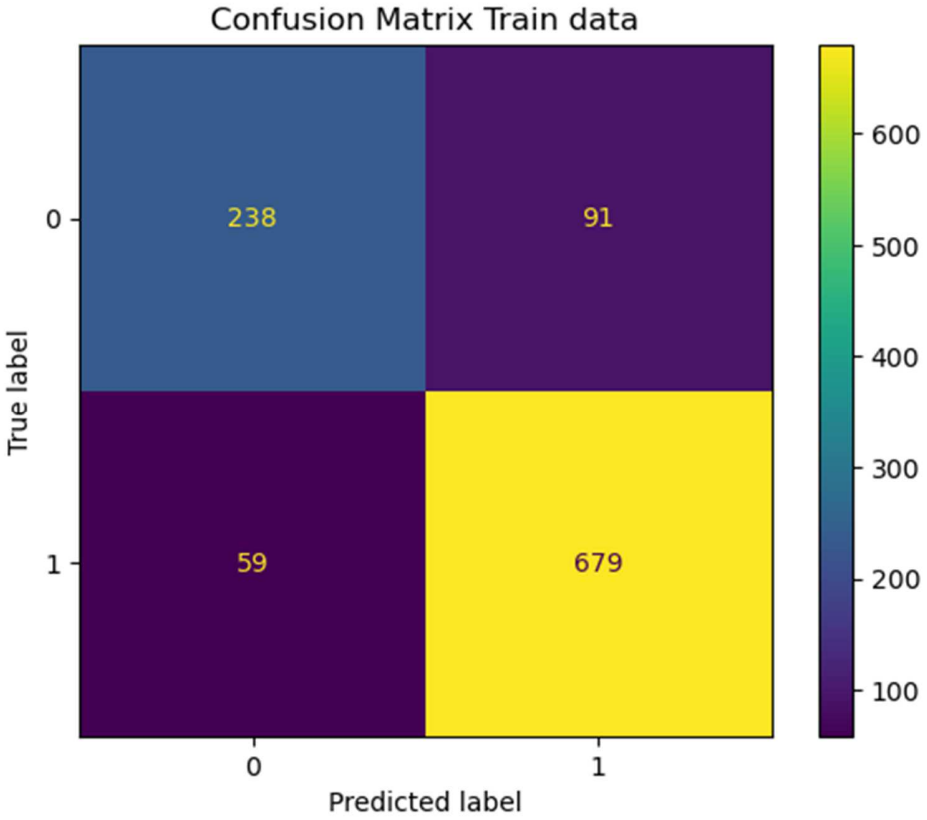


Figure 15: Confusion matrix

Classification report Train data

	precision	recall	f1-score	support
0	0.80	0.72	0.76	329
1	0.88	0.92	0.90	738
accuracy			0.86	1067
macro avg	0.84	0.82	0.83	1067
weighted avg	0.86	0.86	0.86	1067

Table14: Classification report

for test data

array ([[87, 46],
       [39, 286]], dtype=int64)

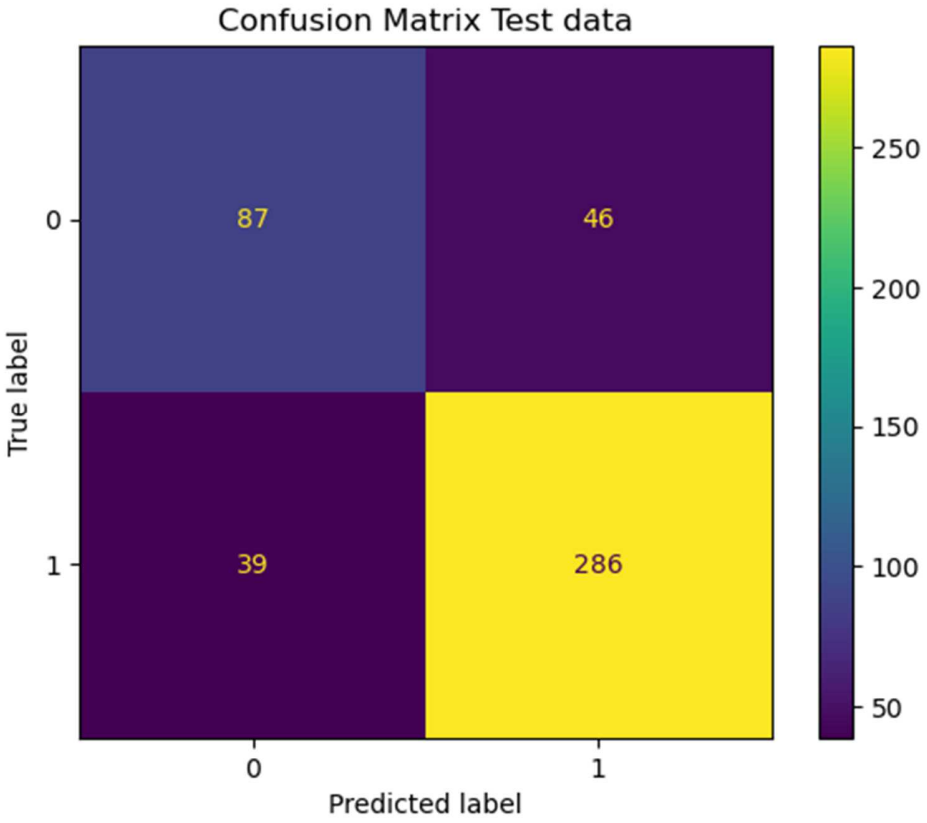


Figure 16: Confusion matrix

Classification report Test data

	precision	recall	f1-score	support
0	0.69	0.65	0.67	133
1	0.86	0.88	0.87	325
accuracy			0.81	458
macro avg	0.78	0.77	0.77	458
weighted avg	0.81	0.81	0.81	458

Table 15: Classification report

We have made models using Naive bayes and KNN method where model accuracy for Naive bayes on test data is 0.814 and AUC score is 0.867, for KNN these metrics values are 0.773 and 0.832 respectively. Based on these scores we can conclude that for now Naive bayes model is the best performing model, however, we will try to improve the model performance by using different ensemble techniques.

Random Forest Model

We built a model using RandomForestClassifier from ensemble module of scikit-learn library whose accuracy for train and test data were:

Model accuracy for train data
0.9991

Model accuracy for test data
0.8144

Since, model accuracy score for train and test set were significantly different we can conclude that there is overfitting in the model which we optimized by finding best parameters using gridsearch cv which were:
{'max_depth': 5, 'max_features': 0.5, 'n_estimators': 400}

Using these parameters, we rebuilt the model using random forest whose model scores were:

Model accuracy for train data
0.8809746954076851

Model accuracy for test data
0.8209606986899564

By pruning the trees in random forest and finding best parameters we have been able to overcome the overfitting by bringing the model scores for train and test data within the acceptable range while improving the model performance.

Model Evaluation

AUC-ROC Curve

for training data
AUC Score: 0.941

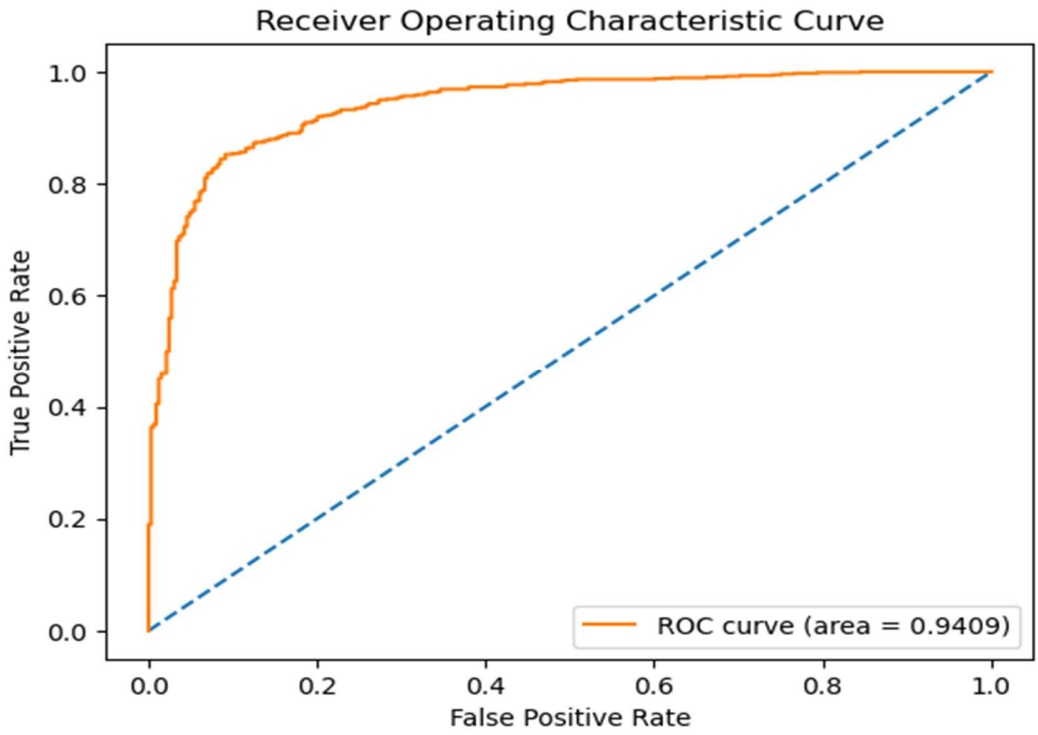


Figure 17: AUC-ROC curve

for test data

AUC Score: 0.884

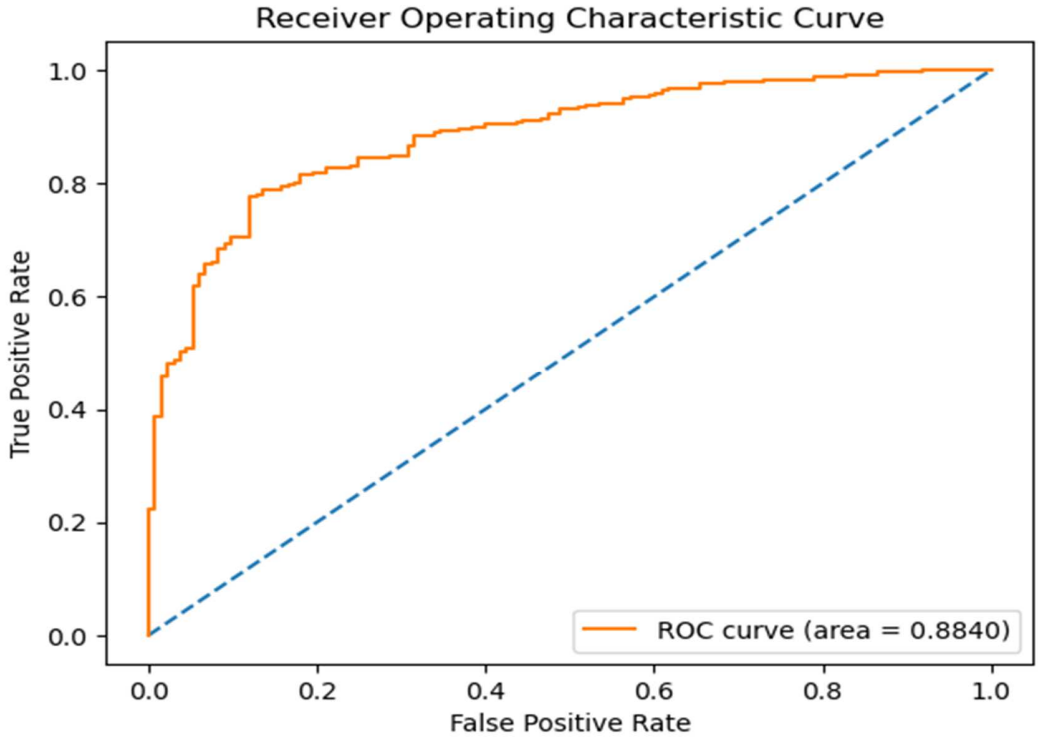


Figure 18: AUC-ROC curve

Confusion Matrix

for train data

array ([[250, 79],
 [48, 690]], dtype=int64)

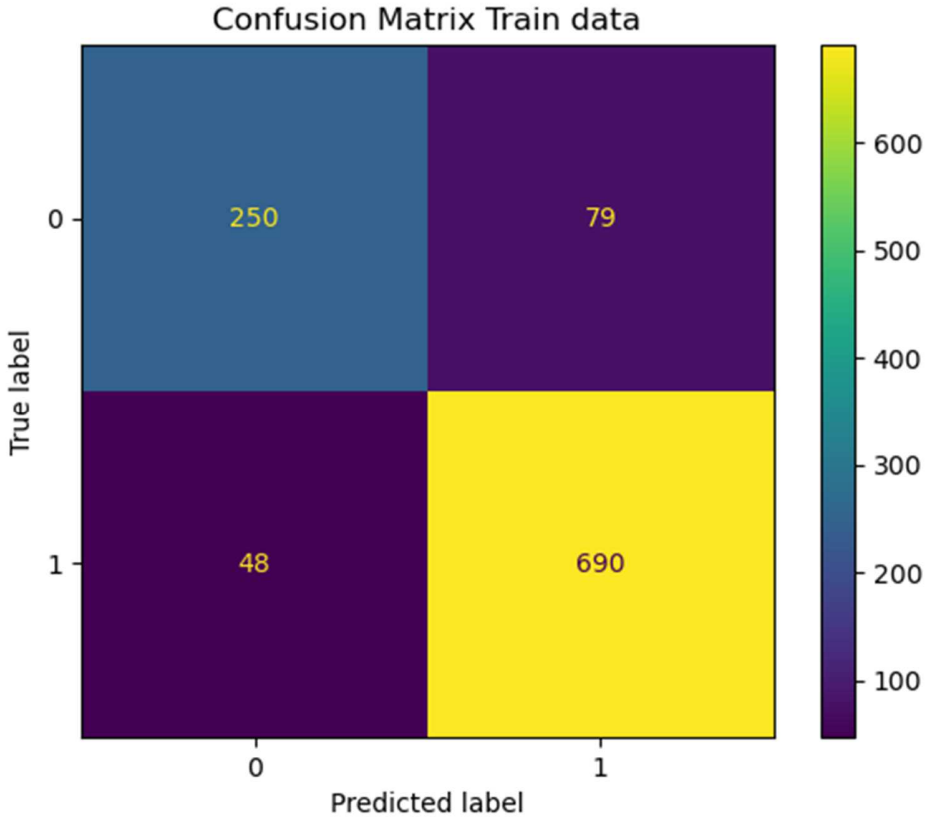


Figure 19: Confusion matrix

Classification report Train data

	precision	recall	f1-score	support
0	0.84	0.76	0.80	329
1	0.90	0.93	0.92	738
accuracy			0.88	1067
macro avg	0.87	0.85	0.86	1067
weighted avg	0.88	0.88	0.88	1067

Table 16: Classification table

for test data

array([[88, 45],
       [37, 288]], dtype=int64)

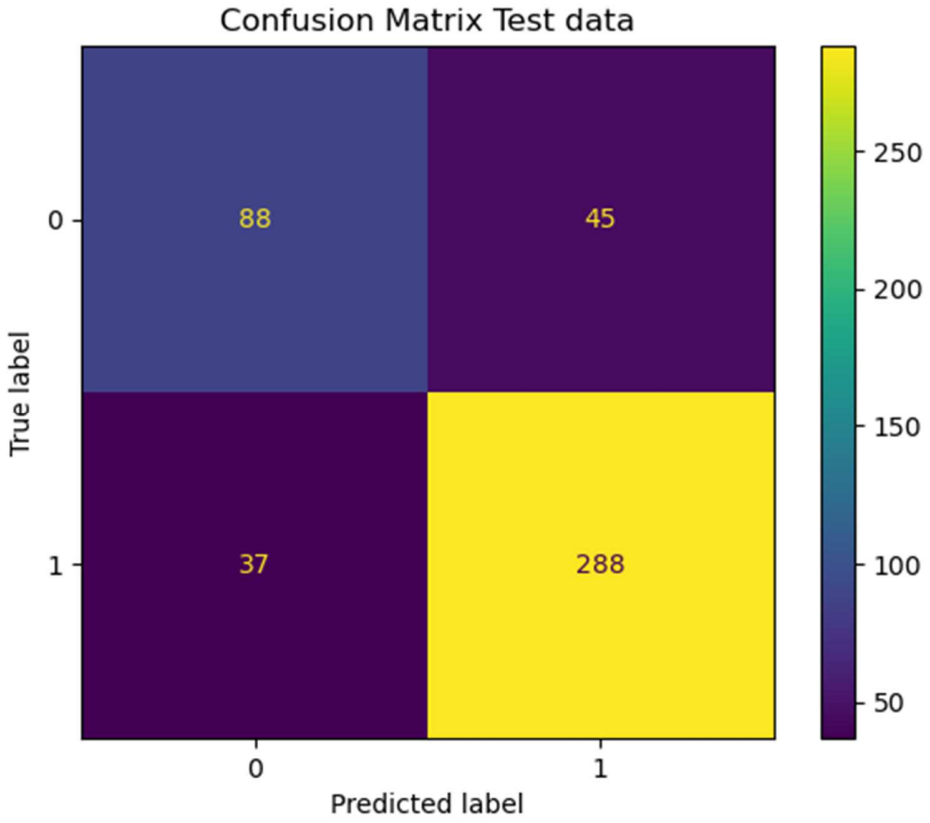


Figure 20: Confusion matrix

Classification report Test data

	precision	recall	f1-score	support
0	0.70	0.66	0.68	133
1	0.86	0.89	0.88	325
accuracy			0.82	458
macro avg	0.78	0.77	0.78	458
weighted avg	0.82	0.82	0.82	458

Table 17: Classification report

Final random forest model has on test data an AUC score of 0.884 meaning it has high performance and there is 88.4% chance that the model will correctly distinguish a randomly chosen instance and f1-score for class 0 (Conservative party) is 0.68 which though still is poor when classifying for minority class there is a slight improvement when compared to that of Naive Bayes.

Bagging Model

We have built a bagging model using BaggingClassifier which is a part of ensemble module in scikit-learn library, accuracy scores for train and test set were:

Model accuracy for train data
0.9878163074039362

Model accuracy for test data
0.8100436681222707

Since, there was a significant difference between accuracy scores for test and train data we concluded that there is overfitting in the model and to overcome this we tried to find the best pruning parameters that can reduce this overfitting using gridsearch cv which came as:
{'max_features': 0.5, 'max_samples': 0.7, 'n_estimators': 3000}

However, when we made the model using these parameters, the overfitting in the model still existed.

Model accuracy for train data
0.9578256794751641

Model accuracy for test data
0.8056768558951966

Since model performance did not improve, we decided use Naive bayes model and KNN model as base in estimator in bagging classifier, model performance was:

Accuracy score for GaussianNB()
Model accuracy for train data
0.8350515463917526

Model accuracy for test data
0.8165938864628821

Accuracy score for KNeighborsClassifier()
Model accuracy for train data
0.8641049671977507

Model accuracy for test data
0.777292576419214

Based on the accuracy scores we can concluded that Naive Bayes model is performing better in bagging, we will continue with it and tried finding best parameters which could improve the model performance using gridsearch cv.
{'max_features': 0.5, 'max_samples': 0.6, 'n_estimators': 3000}

Using these parameters we again built a bagging model whose accuracy scores were:

Model accuracy for train data
0.8209934395501406

Model accuracy for test data
0.8034934497816594

By changing the base estimator and finding best parameters we have been able to overcome the overfitting in the tree, however, building the model based on tuning parameters is actually bringing down the model performance, thus we will continue with without specifying n_estimators by taking base estimator as naive bayes model.

Model accuracy for train data
0.8219306466729147

Model accuracy for test data
0.8187772925764192

The bagging model made on based on updated parameter is showing some improvement in its performance, we will take this as a final bagging model and will evaluate this model.

Model Evaluation

AUC-ROC Curve

for training data
AUC Score: 0.890

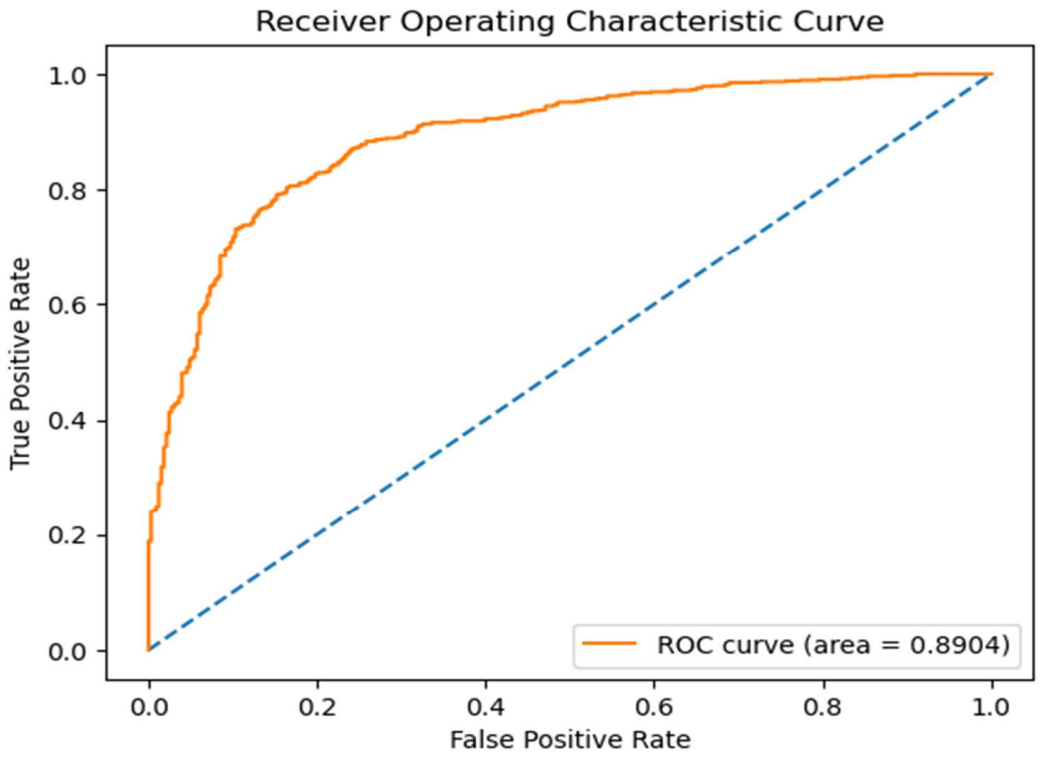


Figure 21: AUC-ROC curve

for test data

AUC Score: 0.871

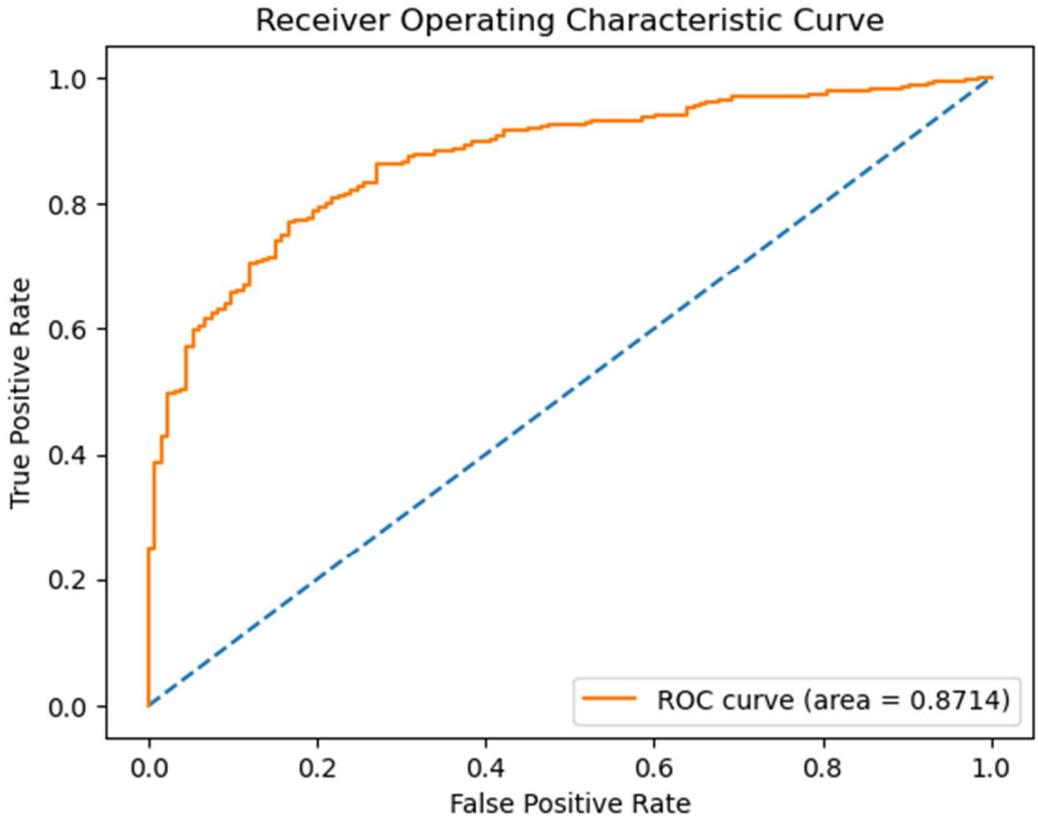


Figure 22: AUC-ROC curve

Confusion Matrix

for training data

array ([[198, 131],
[59, 679]], dtype=int64)

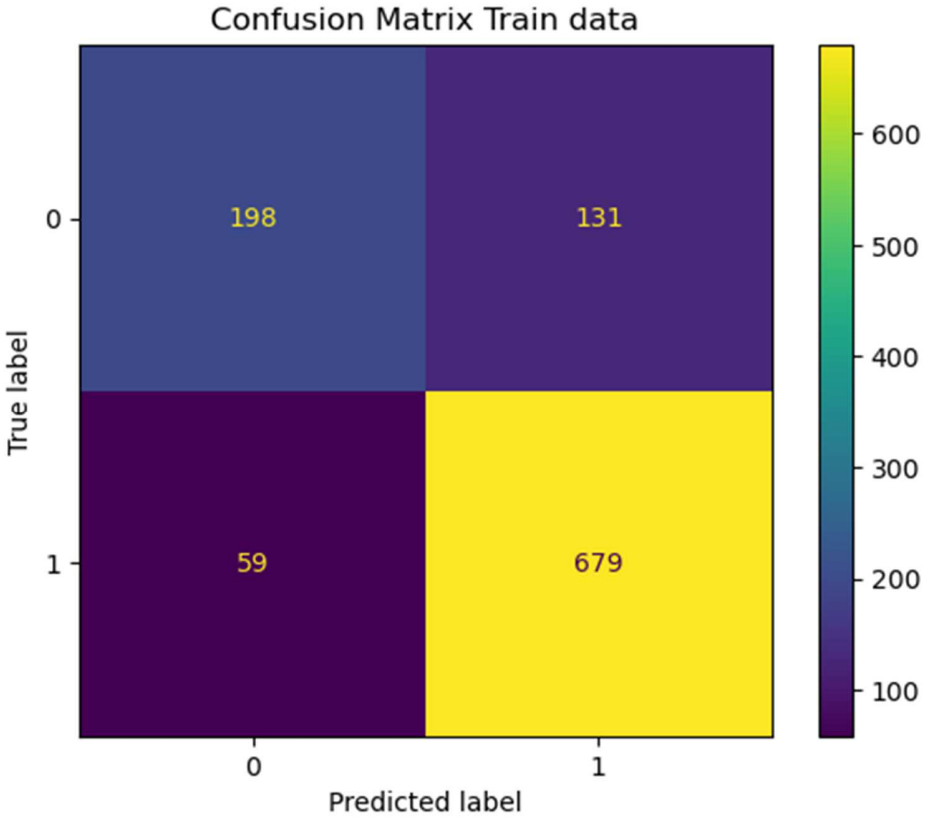


Figure 23: Confusion matrix

Classification report Train data

	precision	recall	f1-score	support
0	0.77	0.60	0.68	329
1	0.84	0.92	0.88	738
accuracy			0.82	1067
macro avg	0.80	0.76	0.78	1067
weighted avg	0.82	0.82	0.82	1067

Table 18: Classification report

for test data

array ([[77, 56],
 [27, 298]], dtype=int64)

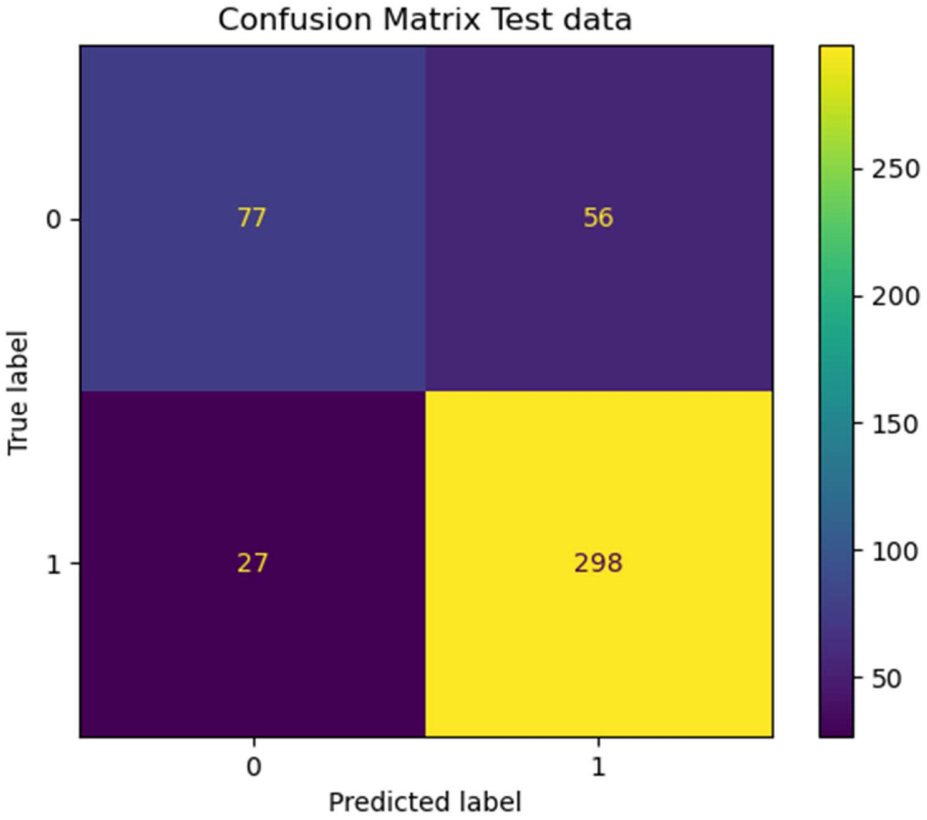


Figure 24: Confusion matrix

Classification report Test data

	precision	recall	f1-score	support
0	0.74	0.58	0.65	133
1	0.84	0.92	0.88	325
accuracy			0.82	458
macro avg	0.79	0.75	0.76	458
weighted avg	0.81	0.82	0.81	458

Table 19: Classification report

Final bagging model has on test data an AUC score of 0.87.14 meaning it has high performance and there is 87.14% chance that the model will correctly distinguish a randomly chosen instance and f1-score for class 0 (Conservative party) is 0.65 which is not very good when classifying for minority class especially for recall the score is 0.58 which is only slightly higher than a random chance.

Ada Boosting

We have built a model using AdaBoostClassifier from ensemble module of scikit-learn, accuracy score for this model is:

Model accuracy for train data
0.85941893158388

Model accuracy for test data
0.8034934497816594

Accuracy score for train and test data are with the acceptable range, we improved the model performance by tuning the parameters for which we explored for the best performing parameter combination using gridsearch cv, which came as:
{'learning_rate': 0.1, 'n_estimators': 1000}

Using these parameters we again built the model whose score was:

Model accuracy for train data
0.8584817244611059

Model accuracy for test data
0.8122270742358079

We were able to bring some improvement to the model performance. We take this as final model and evaluate its performance.

Model Evaluation

AUC-ROC Curve

for training data
AUC Score: 0.917

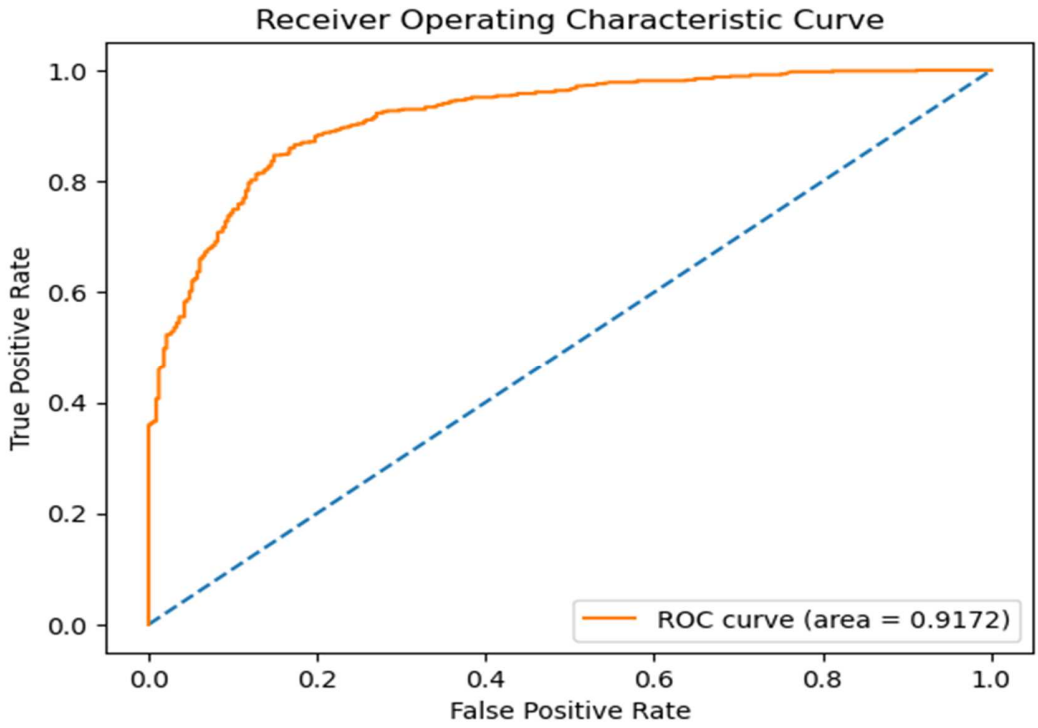


Figure 25: AUC-ROC curve

for test data

AUC Score: 0.875

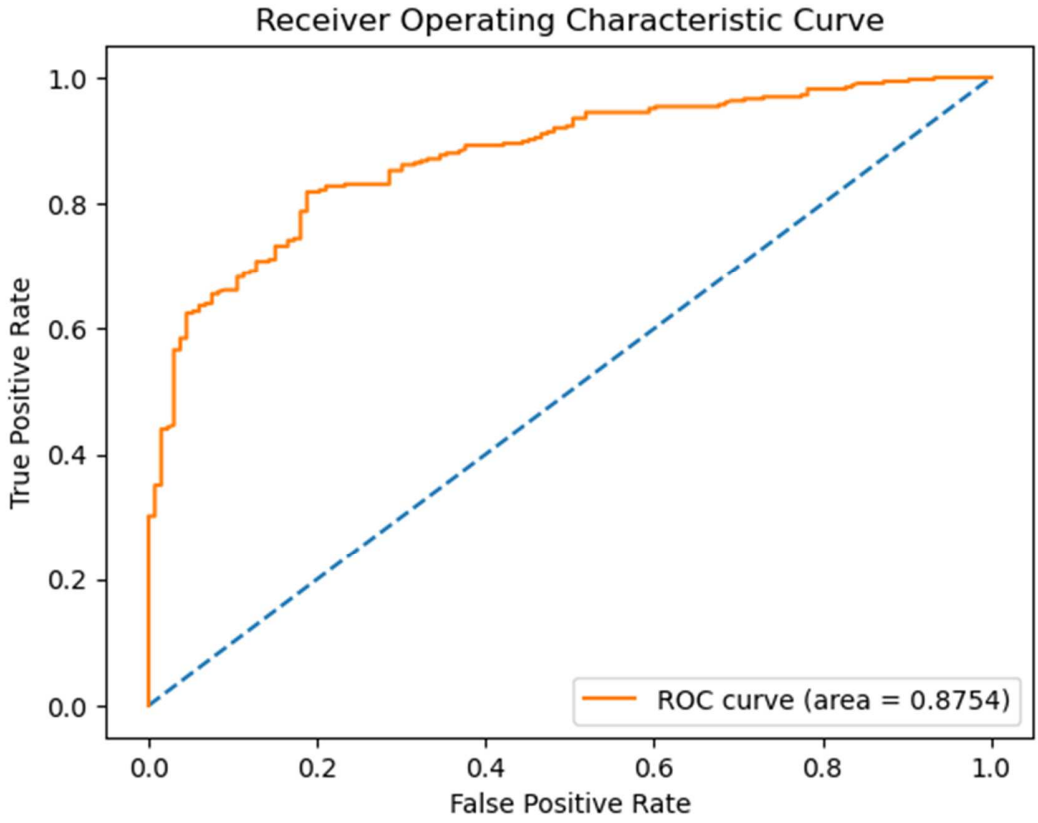


Figure 26: AUC-ROC curve

Confusion Matrix

for training data

array ([[240, 89],
       [62, 676]], dtype=int64)

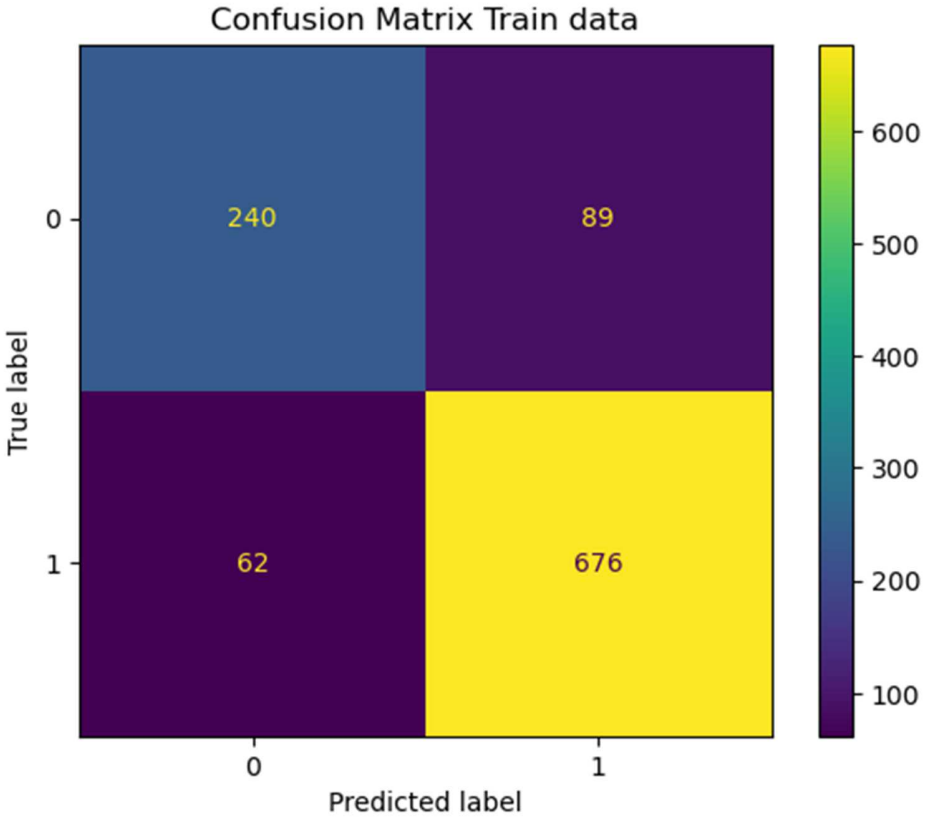


Figure 27: Confusion matrix

Classification report Train data

	precision	recall	f1-score	support
0	0.79	0.73	0.76	329
1	0.88	0.92	0.90	738
accuracy			0.86	1067
macro avg	0.84	0.82	0.83	1067
weighted avg	0.86	0.86	0.86	1067

Table 20: Classification report

for test data

array ([[87, 46],
 [40, 285]], dtype=int64)

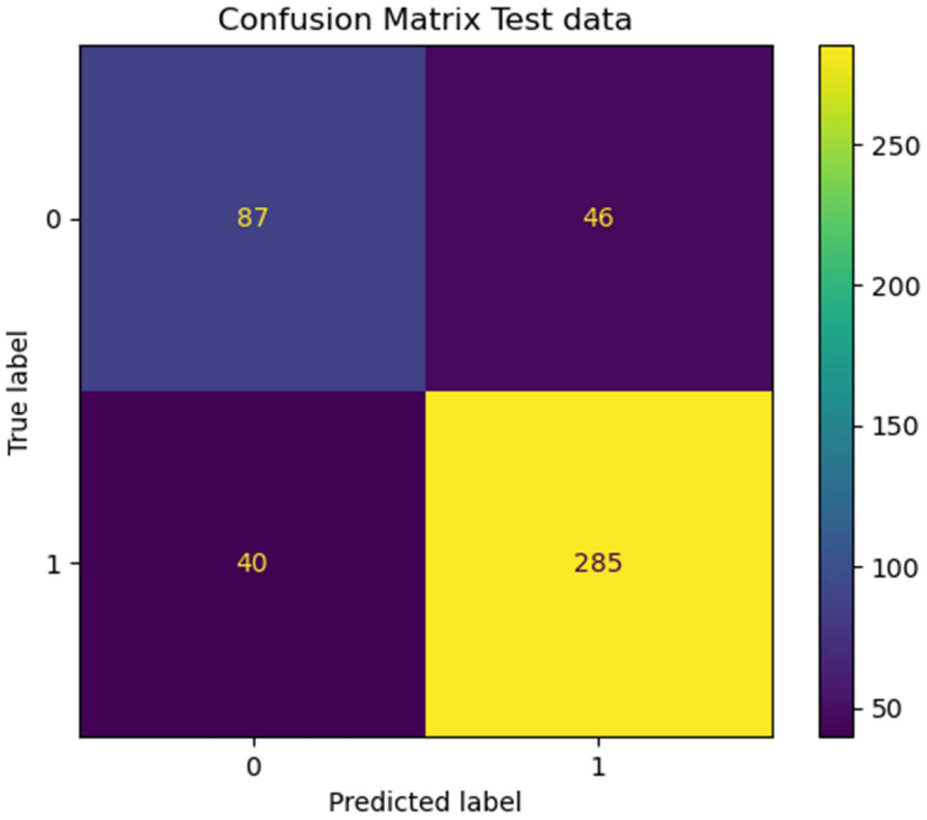


Table 28: Confusion matrix

Classification report Test data

	precision	recall	f1-score	support
0	0.69	0.65	0.67	133
1	0.86	0.88	0.87	325
accuracy			0.81	458
macro avg	0.77	0.77	0.77	458
weighted avg	0.81	0.81	0.81	458

Table 21: Classification report

Tuned Ada boosting model has on test data an AUC score of 0.8752 meaning it has high performance and there is 87.52% chance that the model will correctly distinguish a randomly chosen instance and f1-score for class 0 (Conservative party) is 0.67 which means model is also performing weakly when classifying for minority class.

Gradient Boosting

We have built a model using GradientBoostClassifier from ensemble module of scikit-learn, accuracy score for this model is:

Model accuracy for train data
0.8987816307403936

Model accuracy for test data
0.8034934497816594

Model accuracy score for train and test data are within the acceptable range of 10%, so will consider this a stable model. However, we improved the model's performance by finding best parameters which helped reduce the difference between scores of test and train data while improving model performance on test data, for this purpose we used the following parameters which we got using gridsearch cv.

{'learning_rate': 0.1,
 'max_depth': 2,
 'max_features': 0.2,
 'n_estimators': 150}

Using these parameters we again built the model whose score was:

Model accuracy for train data
0.8697

Model accuracy for test data
0.8187

We were able to bring some improvement to the model performance. We take this as final model and evaluate its performance.

Model Evaluation

AUC-ROC Curve

for training data
AUC Score: 0.926

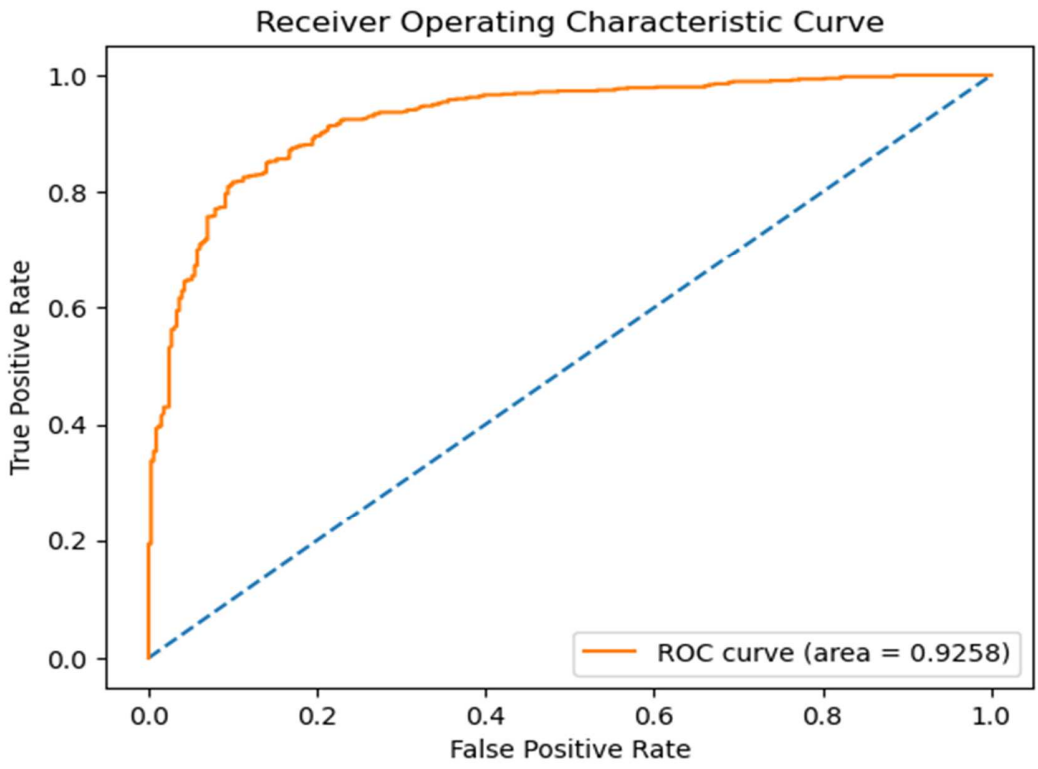


Figure 29: AUC-ROC curve

for test data

AUC Score: 0.881

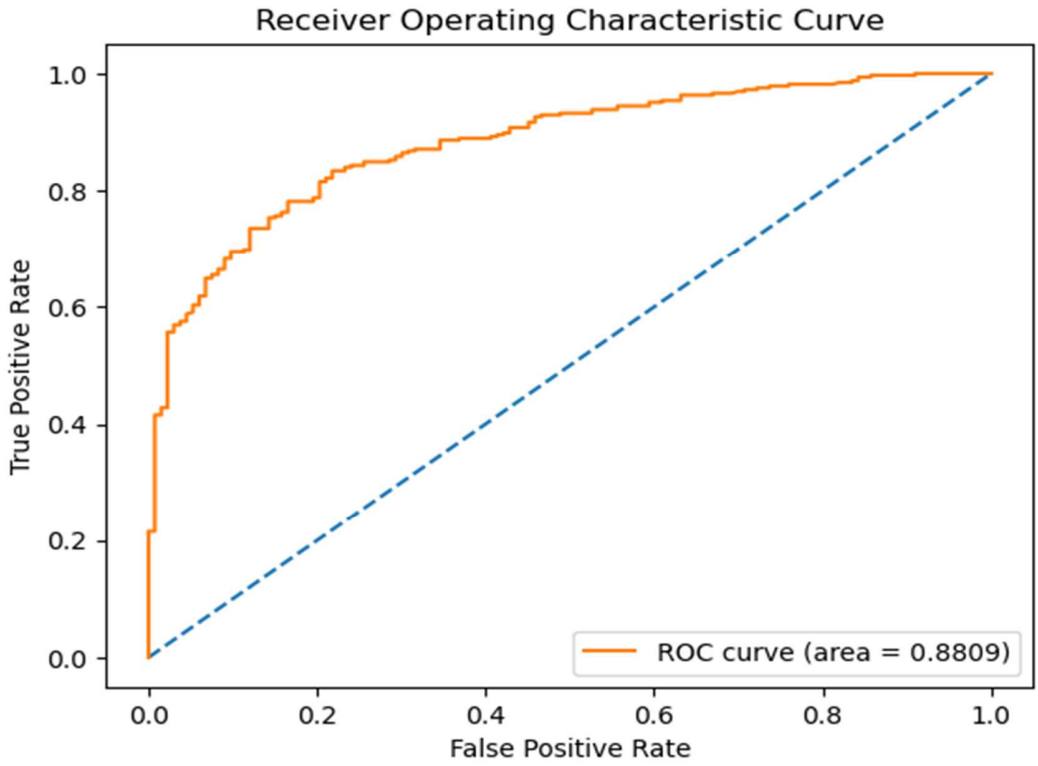


Figure 30: AUC-ROC curve

Confusion Matrix

for training data

array ([[246, 83],
 [56, 682]], dtype=int64)

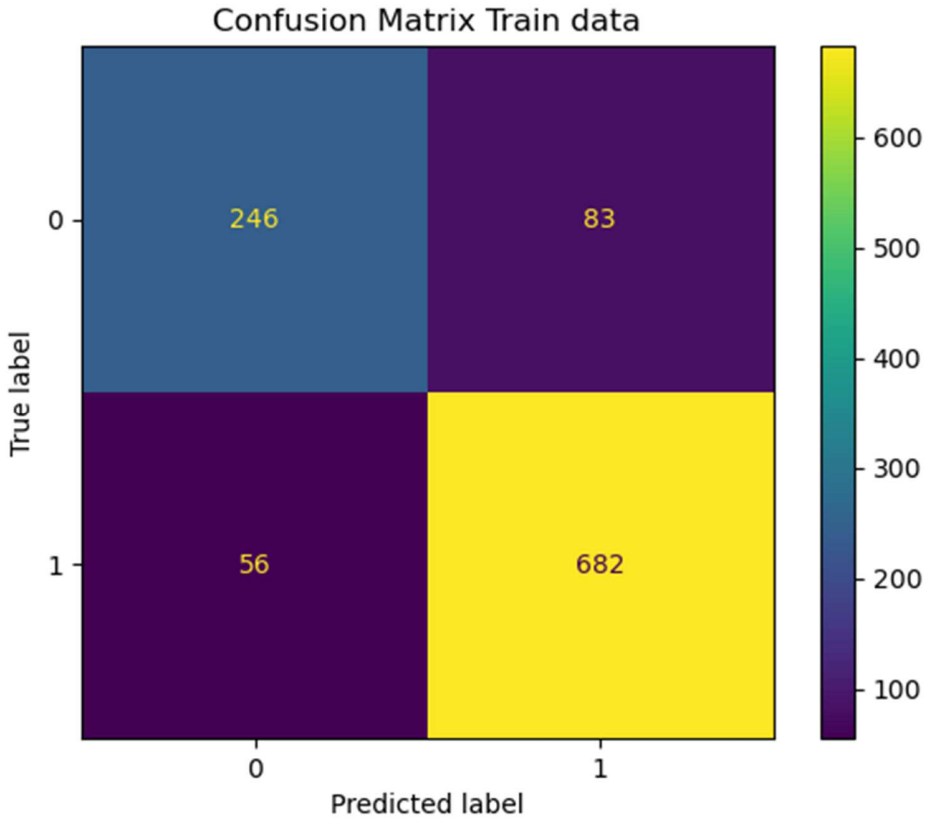


Figure 31: Confusion matrix

Classification report Train data

	precision	recall	f1-score	support
0	0.81	0.75	0.78	329
1	0.89	0.92	0.91	738
accuracy			0.87	1067
macro avg	0.85	0.84	0.84	1067
weighted avg	0.87	0.87	0.87	1067

Table 22: Classification report

for test data

array ([[87, 46],
 [37, 288]], dtype=int64)

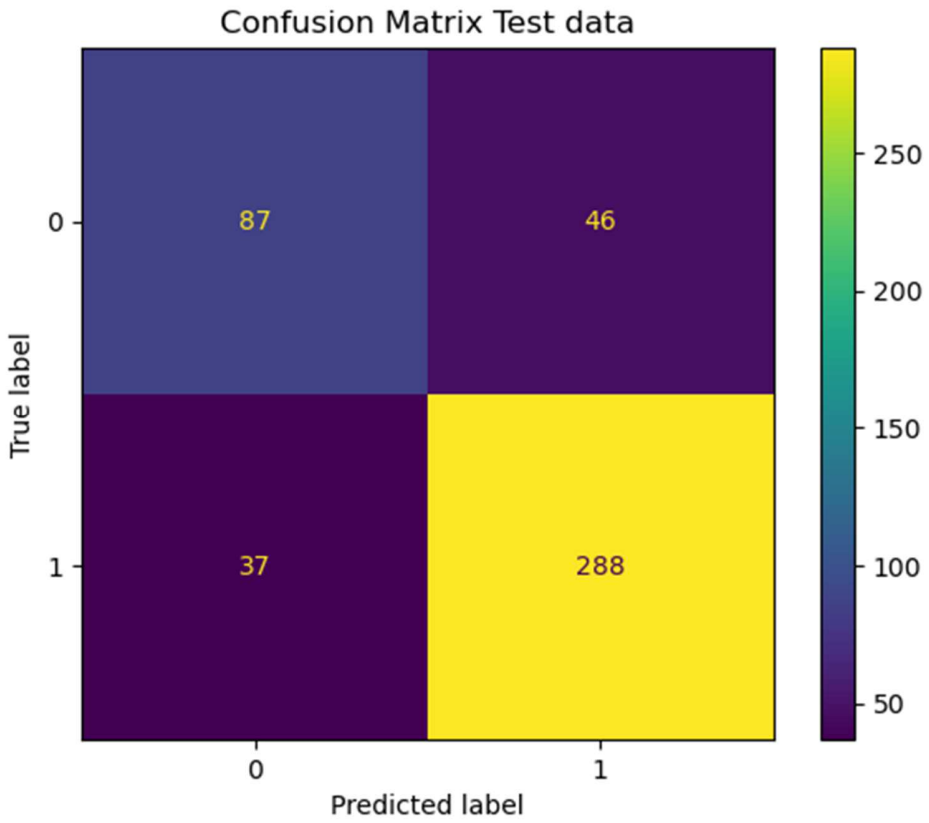


Figure 32: Confusion matrix

Classification report Test data

	precision	recall	f1-score	support
0	0.70	0.65	0.68	133
1	0.86	0.89	0.87	325
accuracy			0.82	458
macro avg	0.78	0.77	0.78	458
weighted avg	0.82	0.82	0.82	458

Table 23: Classification report

Tuned gradient boosting model has on test data an AUC score of 0.8894 meaning it has high performance and there is 88.94% chance that the model will correctly distinguish a randomly chosen instance and f1-score for class 0 (Conservative party) is 0.66 which means model is also performing weakly when classifying for minority class.

1.10 Model Comparison

We have created 6 models using different techniques compared each model's performance for test and train data using key metrics and have found that all the models are stable now we will compare these models with each other to find the best model based on their AUC score for test data.

	Train data score	Test data score	AUC score train data	AUC score test data
Random Forest Model	0.880975	0.820961	0.940641	0.884118
Gradient Boosting Model	0.869728	0.818777	0.925820	0.880937
ADA Boosting Model	0.858482	0.812227	0.917216	0.875361
Bagging Model	0.821931	0.818777	0.890419	0.871394
Naive Bayes Model	0.837863	0.814410	0.891535	0.866767
KNN Model	0.859419	0.772926	0.929247	0.832354

Table 24: Model comparison

Based on the above table, we can conclude that when comparing models by AUC score for the test set, random forest model is performing the best with an AUC score of 0.8841. This indicates that the random forest model has a superior ability to distinguish between classes, making it the most effective model for our classification task. We will check for the most important features which play crucial role in distinguishing between classes.

1.11 Important Features

	imp
Hague	0.291610
Blair	0.235054
Europe	0.191686
Political.knowledge	0.105977
Age	0.081862
Economic.cond.national	0.061651
Economic.cond.household	0.024827
Gender	0.007334

Table 25: Important Features

From the above table we can conclude that as we had anticipated features 'Hague' and 'Blair' are the most important features in classification model with an impact of 0.291610 and 0.235054 followed by 'Europe' which records how Eurosceptic a person is, meaning 'Hague' account for about 29.16% of the model while 'Blair' account for 23.50% of the model on the contrary impact of 'Gender is only 0.007334 meaning its impact on model is only 0.73%

1.12 Conclusion

1. For the current election, the most important factor is the candidate where a voter is more likely to vote for the candidate for whom they have a favorable view. In fact, just by taking columns 'Hague' and 'Blair' we can build a model with an accuracy of 0.7947 and AUC score of 0.836 which is only couple of points lower than the overall model scores.
2. On comparing different models by their AUC scores, we find that Random Forest model is performing the best with a score of 0.8841 on test data meaning there is 88.41% chance that the model will correctly distinguish a randomly chosen instance. We have used AUC score as a key metric to select the best performing model because AUC score is a robust measure for skewed data as it accounts for both the true positive rate and the false positive rate. It provides a single metric that summarizes the model's ability to distinguish between classes across all thresholds.
3. Key takeaways from EDA:

- Based on the EDA, we found that voters had a clear preference for candidates, as reflected in the ratings they gave to both candidates. This clarity significantly aided in prediction, with 'Hague' and 'Blair' emerging as the most influential features impacting the outcome.
- Another important factor was Europe where from EDA we found that voters who are more Eurosceptic are more likely to vote for the Conservative party.
- The median age of voters for Conservative party is higher than that for Labour party meaning older people are more likely to vote for the Conservative party.

4. While we successfully built and selected the best-performing predictive model with approximately 82% accuracy and an AUC score of 88%, there is potential for further improvement. By expanding the sample to include more features that clearly capture a voter's preference, similar to the features 'Hague' and 'Blair', we can enhance the model's performance. Additionally, recording which factors are most important to voters—such as candidate preference, Europe, or economic factors—would allow us to assign weights to each factor, thereby refining our model further.

Problem 2

2.1 Business Context

To analyse the speeches of three different Presidents of the United States and find the most frequent words used which might give us the understanding of their priorities and provide some insight into their policy making.

2.2 Problem Statement

The objective of this analysis is to study speeches for three different Presidents of the United States belonging from three different decades and find the most common words used by each of them in these speeches.

2.3 METHODOLOGY

Import the libraries - Load the text - Check the text – Descriptive statistical summary – Text cleaning – Frequency count – Word cloud – Key takeaway.

Key Points

1. **Data Collection**: Data was taken from inaugural module of nltk library.
2. **Text Cleaning:** Text was tokenized, non-alphabetic characters and stop words were removed and words were stemmed.
3. **Visualization Techniques:** In the report we have used bar chart for descriptive statistical summary and word cloud to show the most common words used.
4. **Tools and Software:** We have carried out the analysis using programming language python on Jupyter notebook. For this analysis Python libraries Matplotlib, Seaborn, Nltk, Re, String and Wordcloud were used.

2.4 Data Overview

1. **Number of characters:** Number of characters in each speech.

- Number of characters in 1941 speech of Roosevelt: 7571
- Number of characters in 1961 speech of Kennedy: 7618
- Number of characters in 1973 speech of Nixon: 9991

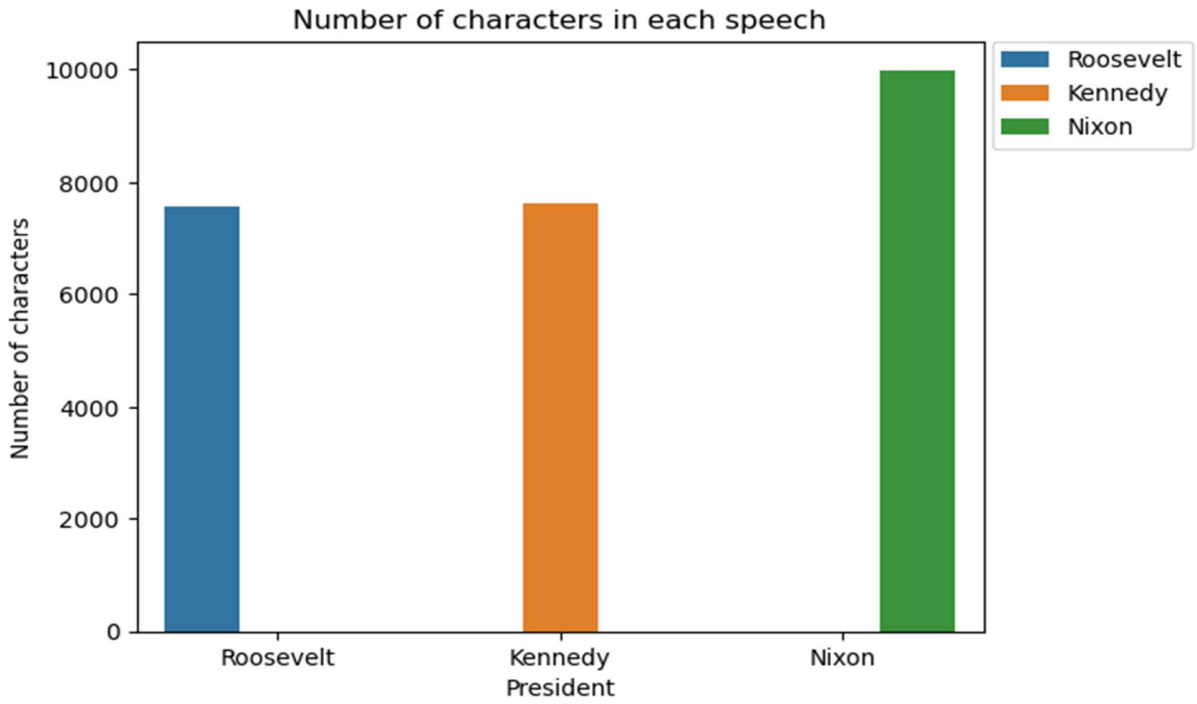


Figure 33: Character count

2. **Number of words:** Number of words in each speech.

- Number of words in 1941 speech of Roosevelt: 1536
- Number of words in 1961 speech of Kennedy: 1546
- Number of words in 1973 speech of Nixon: 2028

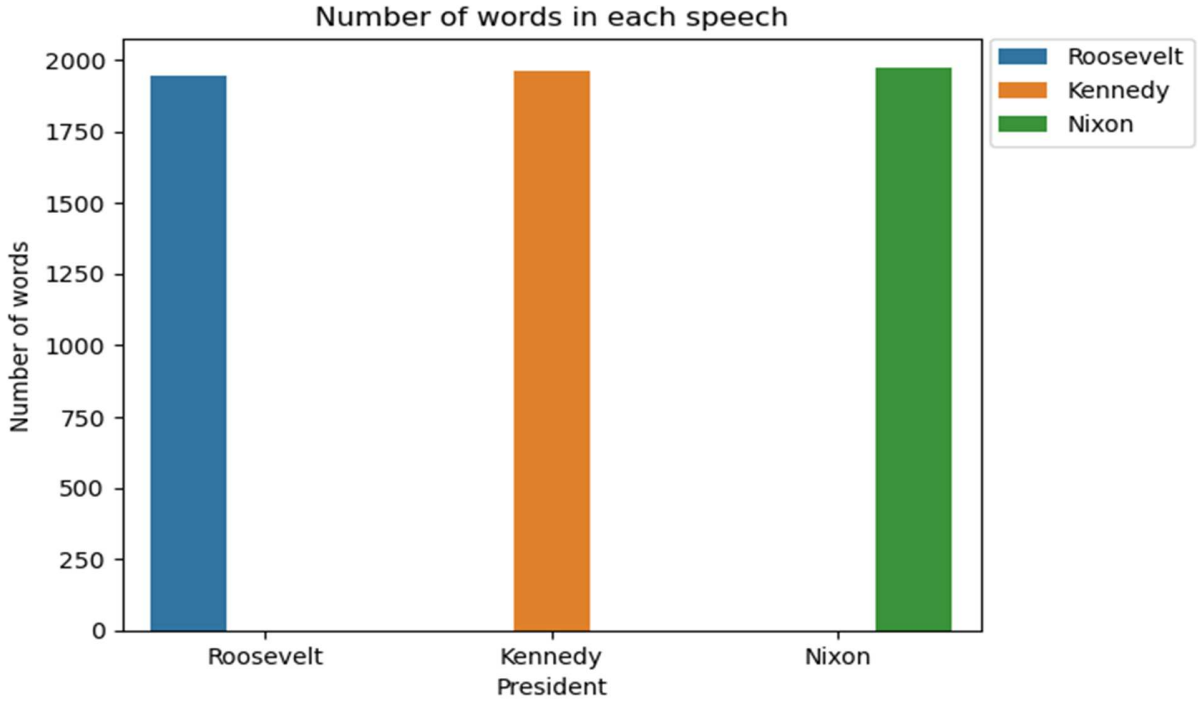


Figure 34: Word count

3. **Number of sentences:** Number of sentences in each speech.

- Number of sentences in 1941 speech of Roosevelt: 68
- Number of sentences in 1961 speech of Kennedy: 52
- Number of sentences in 1973 speech of Nixon: 69

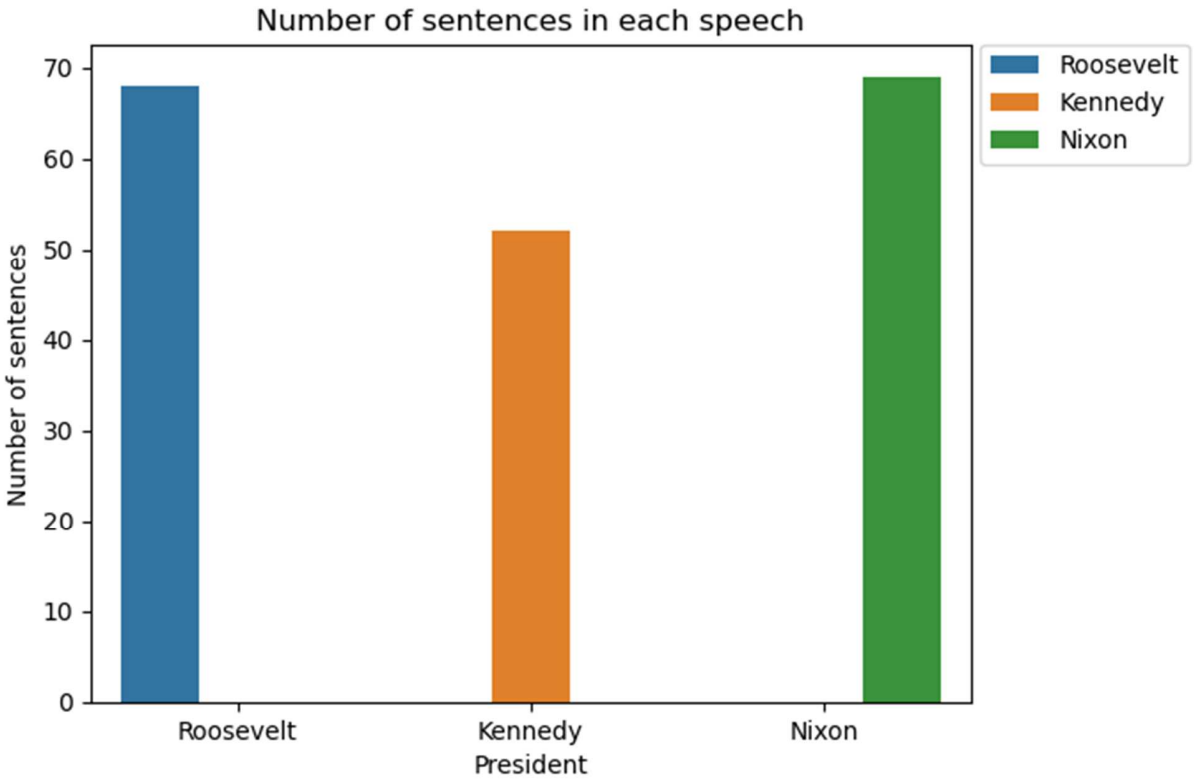


Figure 35: Sentence count

Key Observation

While as per character and word count President Nixon has used significantly more words and characters, number of sentences used by both President Roosevelt and President Nixon are almost same which has been better represented in a visualized manner.

2.5 Frequency Count

To understand which president has emphasised more on which words we did a frequency count to find how many times each word was used in their speech and on basis of this we shortlisted 3 most common words used by each president.

- Top 3 words used by President Roosevelt in speech of 1941
  [('nation', 17), ('know', 10), ('peopl', 9)]

- Top 3 words used by President Kennedy in speech of 1961
  [('let', 16), ('us', 12), ('power', 9)]

- Top 3 words used by President Nixon in speech of 1973
  [('us', 26), ('let', 22), ('america', 21)]

2.6 Word cloud

To identify the most commonly used words in each speech, we created a word cloud for each one. This allowed us to see which words were used most frequently by each president and how these words changed with different presidents and across decades.

Roosevelt speech 1941

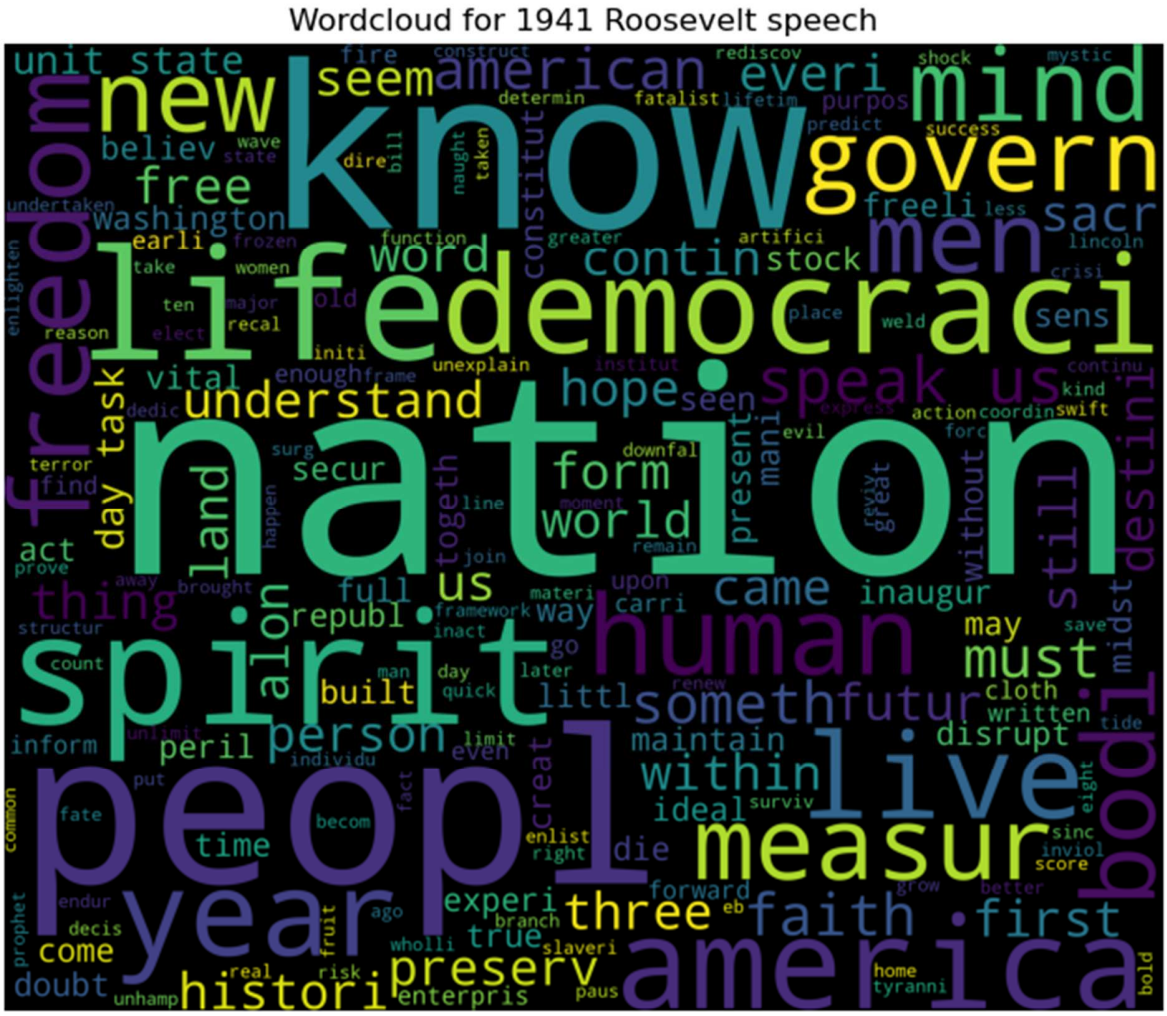


Figure 36: Word cloud

Key Observation

In his speech, President Roosevelt used most frequently words like ‘Nation’, ‘Know’, ‘Life’, ‘Democracy’, ‘People’, ‘Spirit’, ‘Freedom’ based we can infer that the emphasis was more towards the nation, its values and its people.

Kennedy speech 1961

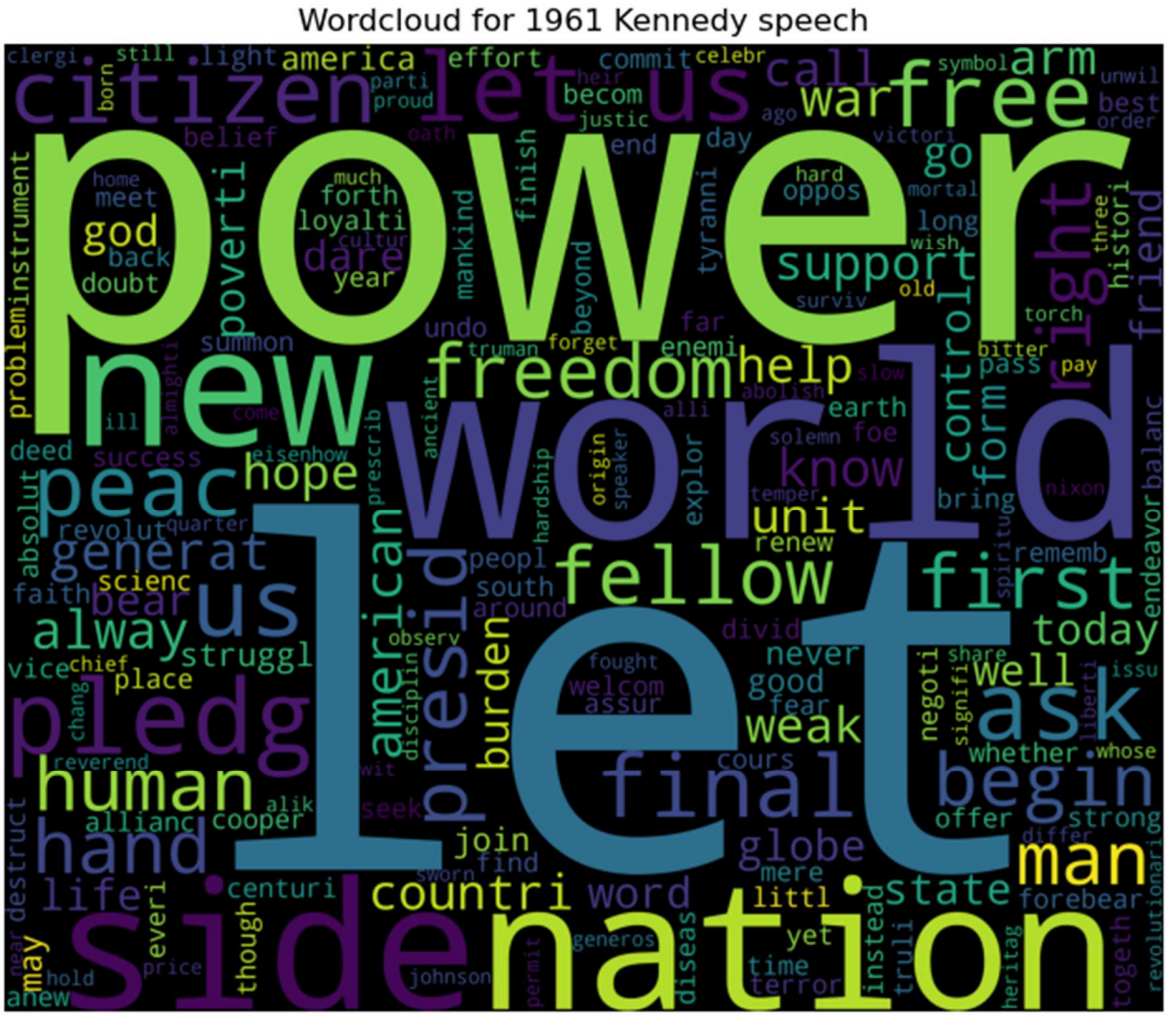


Figure 37: Word cloud

Key Observation

"In his speech, President Kennedy frequently used words like ‘Let’, ‘Power’, ‘World’, ‘New’, ‘Nation’, and ‘Side’. While some words are common in both speeches, the inclusion of new terms is notable. The larger size of these words in the word cloud indicates a shift in emphasis towards a global role, highlighted by the use of words like ‘World’ and ‘Power’.

Nixon speech 1973

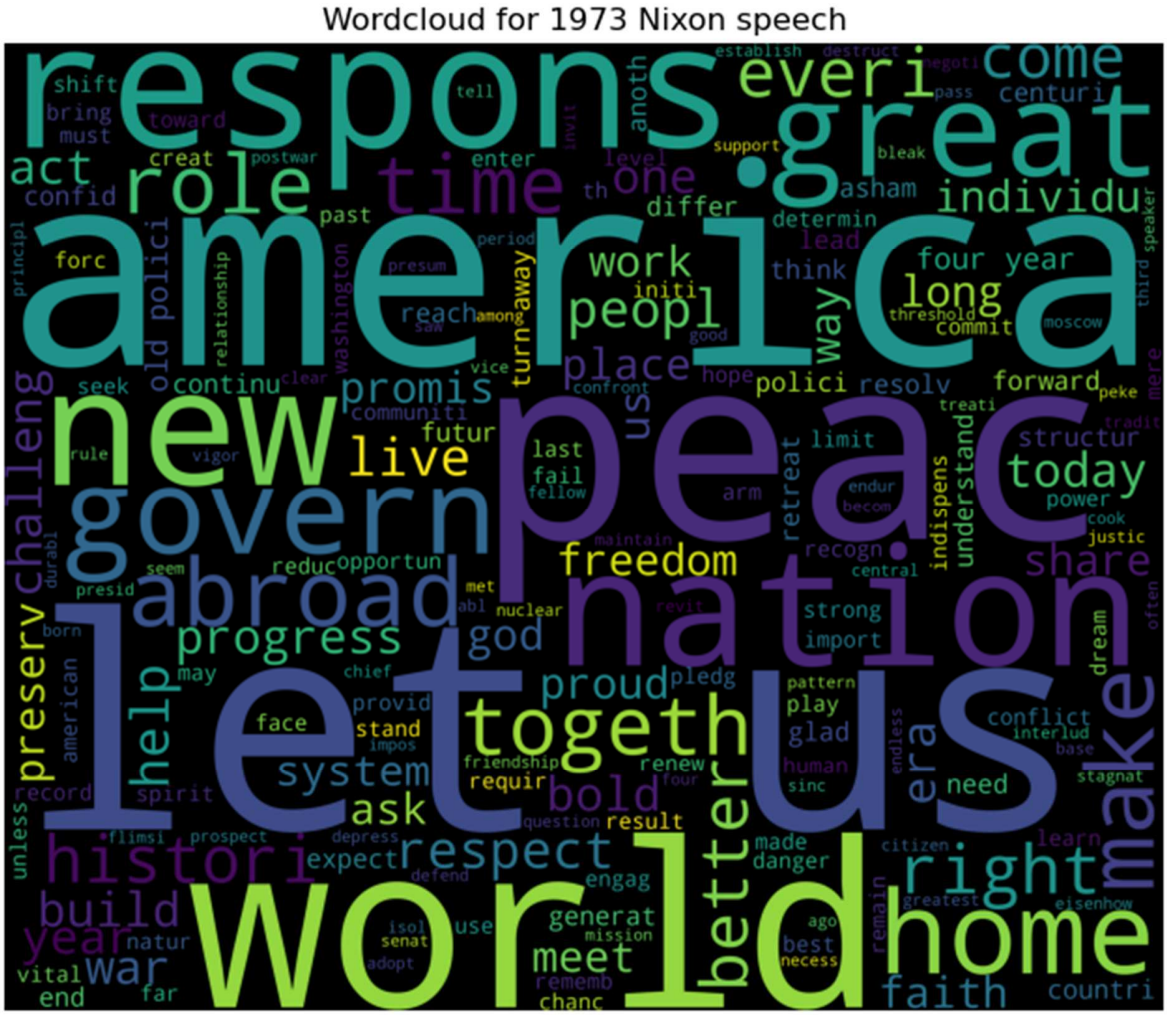


Figure 38: Word cloud

Key Observation

In President Nixon's speech most frequently, words used included 'America', 'Peace', 'Let', 'Us', 'World', 'Great', 'Nation', 'Home', 'Response' based on which we can infer that emphasis is now on how make the nation better while maintaining peace.

2.7 Kay Takeaway

The word clouds generated for this text analysis project provide valuable insights into the evolving priorities of U.S. presidents across different decades. In 1941, President Roosevelt's speeches prominently featured words such as 'Nation,' 'Democracy,' and 'Freedom,' reflecting the focus on national unity and democratic values during that era. By 1961, under President Kennedy, the emphasis shifted towards 'World' and 'Power,' indicating a broader international perspective and the geopolitical dynamics of the Cold War. By 1971, President Nixon's speeches highlighted words like 'Peace' and 'Response,' signalling a focus on achieving peace and addressing immediate national and international

challenges. These changes in word usage underscore how presidential rhetoric adapts to the prevailing political, social, and global contexts of their respective times. and can be extremely helpful in predicting countries future policies and future course of action.