Election Data Analysis and Voter Prediction Models

Predictive Analysis of Voter Behavior for Exit Poll Creation

Shoyeb Ansari

March 5, 2025

GitHub Repository:
https://github.com/Shoyeb45/ElectionVotingPrediction

Contents

List of Figures

List of Tables

1 Introduction

This report presents a comprehensive analysis of election data comprising 1525 voters with 9 variables. The main objective was to build machine learning models to predict which party a voter will vote for based on the given information. This predictive analysis will be used to create an exit poll that will help in forecasting the overall win in seats covered by a particular political party.

1.1 Project Scope and Objectives

The primary goals of this project were:

- To perform thorough exploratory data analysis on the voter dataset
- To identify key factors that influence voting patterns
- To build and compare different classification models for predicting voter behavior
- To select the optimal model for exit poll implementation

2 Data Ingestion and Preprocessing

The data is imported using pandas. Using read_excel() function and extracting particular sheet by providing that sheet name.

```
df = pd.read_excel("./Data/Election_Data.xlsx", sheet_name="Election_Dataset_Two Classes")
```

Figure 1: Reading data

2.1 Dataset Overview

The dataset consists of 1525 voter records with 9 variables. These variables include various demographic and behavioral attributes that could potentially influence voting decisions.

Table 1: Description of Variables

Variable	Description	DataType
vote	Party choice: Conservative or Labour	object
age	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

2.2 Null Value Analysis and Unnecessary Columns

A thorough check for missing values was conducted to ensure data quality and completeness before proceeding with the analysis.

Table 2: Null Value Analysis

Variable	Missing Values
vote	0
age	0
economic.cond.national	0
economic.cond.household	0
Blair	0
Hague	0
Europe	0
political.knowledge	0
gender	0

Missing values in the Income and Media Exposure variables were addressed through imputation using the median value for numerical variables and mode for categorical variables.

There is one index column which is name 'Unnamed: 0', so we'll drop them.

```
df.drop(columns = "Unnamed: 0", axis = 1, inplace = True)
```

Figure 2: Dropping column

3 Exploratory Data Analysis

3.1 Descriptive Statistics

The initial inspection of the dataset revealed the following characteristics:

Table 3: Descriptive Statistics of Numerical Variables

Variable	**Count**	**Mean**	**Std. Dev.**	**Min**	**Max**
Age	1525	54.18	15.71	24	93
Economic Condition (National)	1525	3.25	0.88	1	5
Economic Condition (Household)	1525	3.14	0.93	1	5
Blair	1525	3.33	1.17	1	5
Hague	1525	2.75	1.23	1	5
Europe	1525	6.73	3.30	1	11
Political Knowledge	1525	1.54	1.08	0	3

Table 4: Descriptive Statistics of Categorical Variables

Variable	**Count**	**Unique**	**Top**	**Frequency**
Vote	1525	2	Labour	1063
Gender	1525	2	Female	812

3.2 Univariate Analysis

Univariate analysis was performed to understand the distribution of individual variables and identify potential anomalies.

3.2.1 Distribution of Categorical Variables

We need to visualize distribution of the categorical variables. We can see there is a slight imbalance in vote data which is our target variable

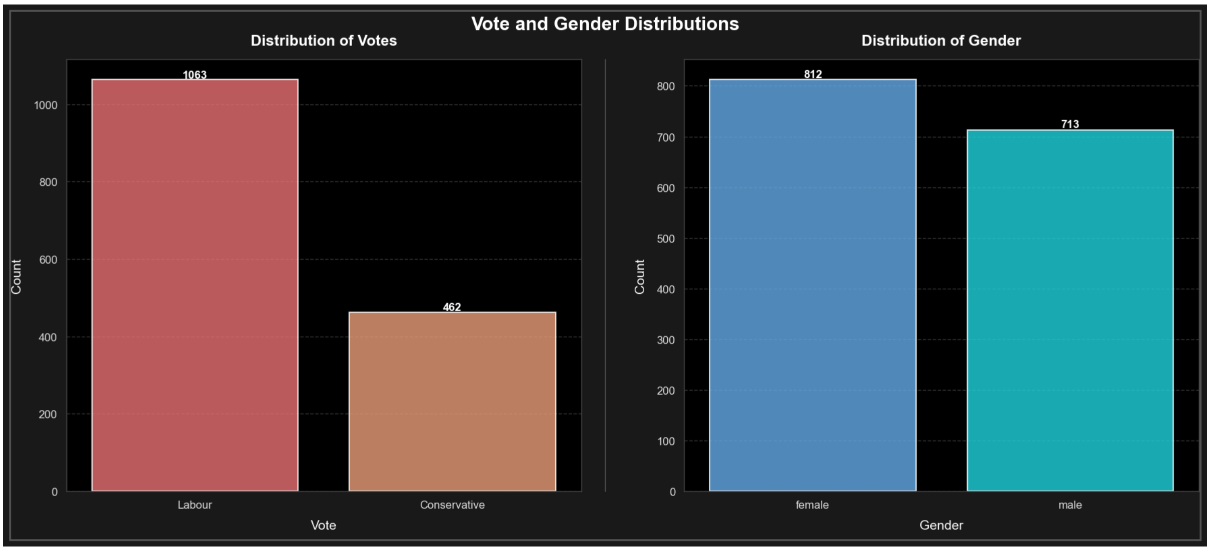

Figure 3: Categorical univariate analysis

3.2.2 Distribution of Numerical Variables

We have 7 numerical variables, so it's important to see the distribution of each variable and draw some important insight.

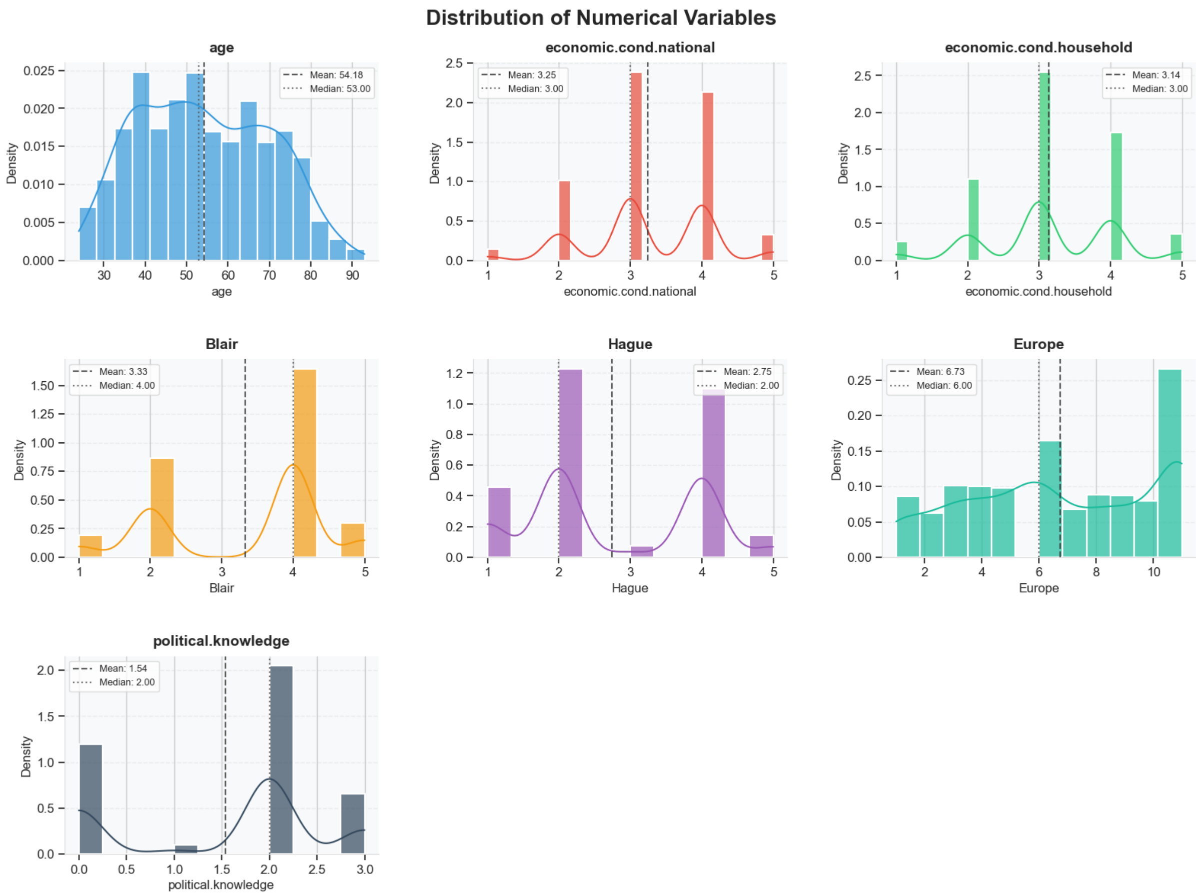

Figure 4: Histogram of all numerical variable

We can see most variable don't follow any specific distribution, but 'age' variable is tending for Gaussian distribution.

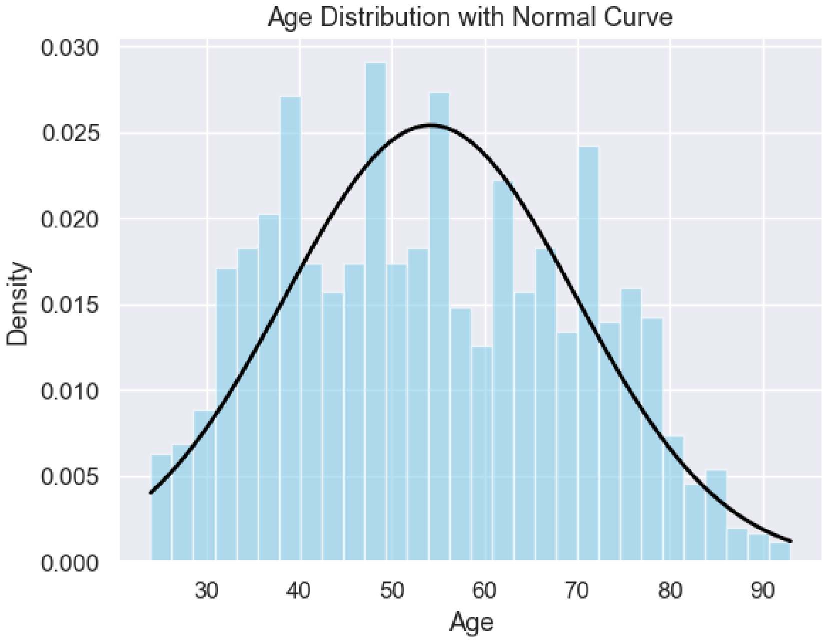

Figure 5: Distribution of age

There's not that much deviation from the Gaussian distribution of age variable. So we can infer that under some condition age variable is following gaussian distribution.

3.3 Bivariate Analysis

Bivariate analysis was conducted to examine relationships between variables and particularly to understand how different factors correlate with the target variable .

We can see that there is a correlation of 0.35 between `economic.cond.household` and `economic.cond.national`.

1. **Is 0.35 Too High for Multicollinearity?**
   - No, a correlation of 0.35 is not high enough to indicate serious multicollinearity.
   - Typically, multicollinearity becomes a problem if $|r| > 0.7$ between independent variables.
2. **Does It Affect Model Performance?**
   - If these correlated features provide unique information, they may still improve the model.
   - If they contain redundant information, one of them might be unnecessary.
   - **Action**: Run feature importance analysis (like SHAP or permutation importance) to check if both features contribute.

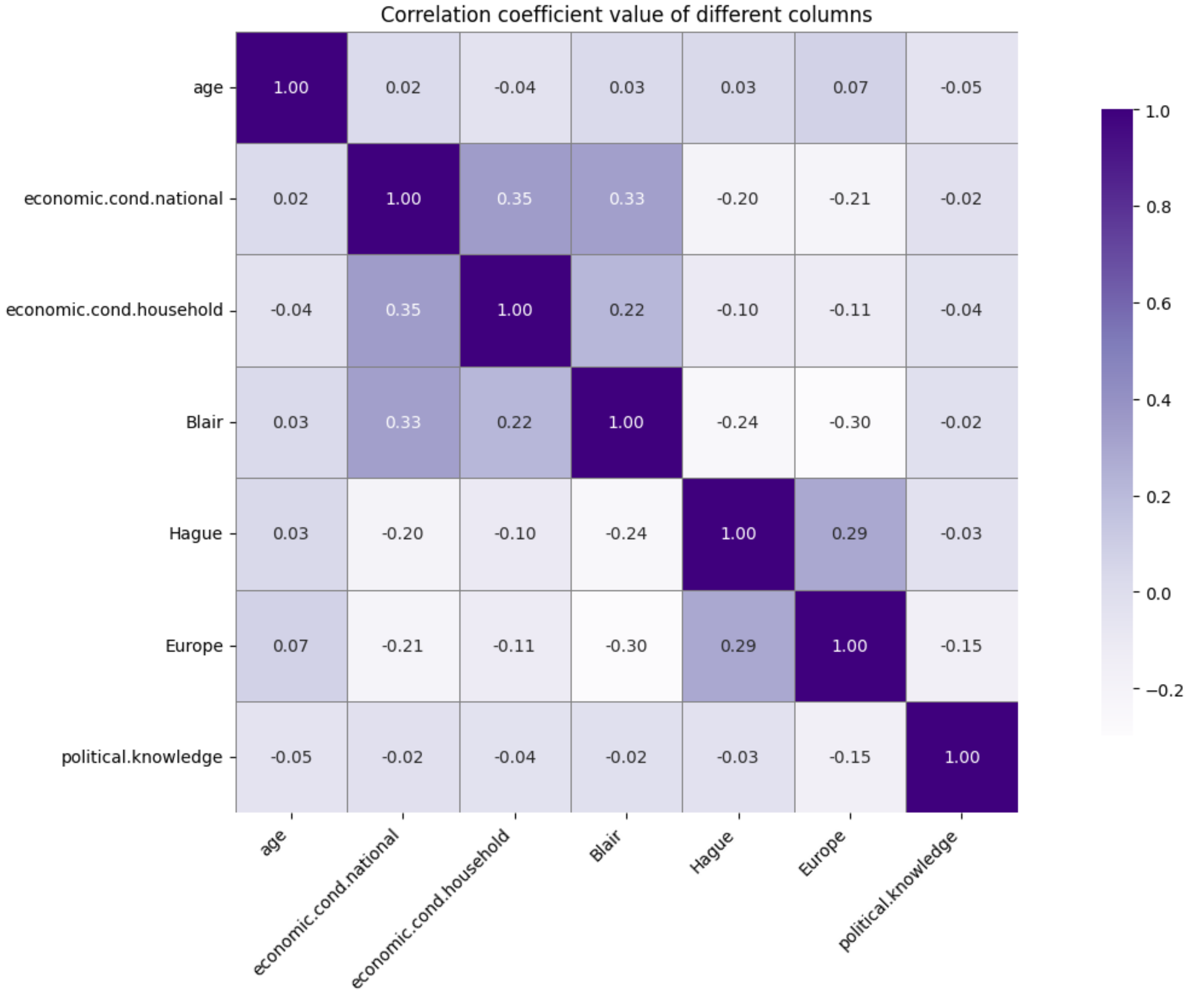

Figure 6: Correlation heatmap between variables

3. **Shall we Drop One of the Features?**

   - If we will use tree-based models (e.g., Decision Trees, Random Forest, XG-Boost, etc.), they handle correlated features well, so we don't need to drop anything.
   - If we will use linear models (e.g., Logistic Regression, SVM), mild correlation usually isn't a big issue, but feature scaling & regularization (L1/L2) can help.
   - **Action**: If using Logistic Regression, check Variance Inflation Factor (VIF) to detect multicollinearity.
   - VIF > 5 or 10 → Feature is highly collinear and might need removal.

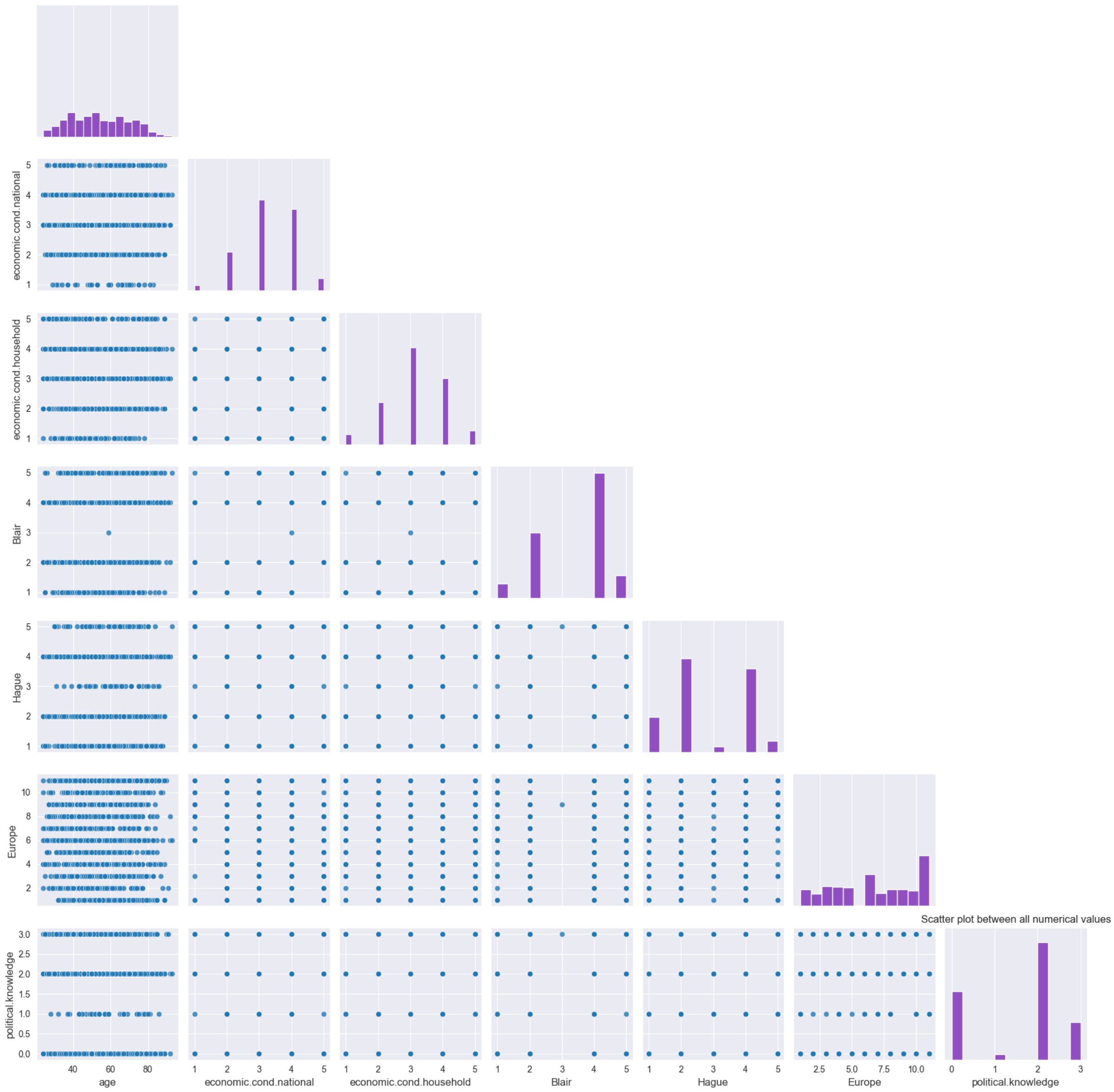

Figure 7: Scatter plot between all numerical values

We also need to visualize categorical variables. We only have 2 categorical variables. So let's see by gender.

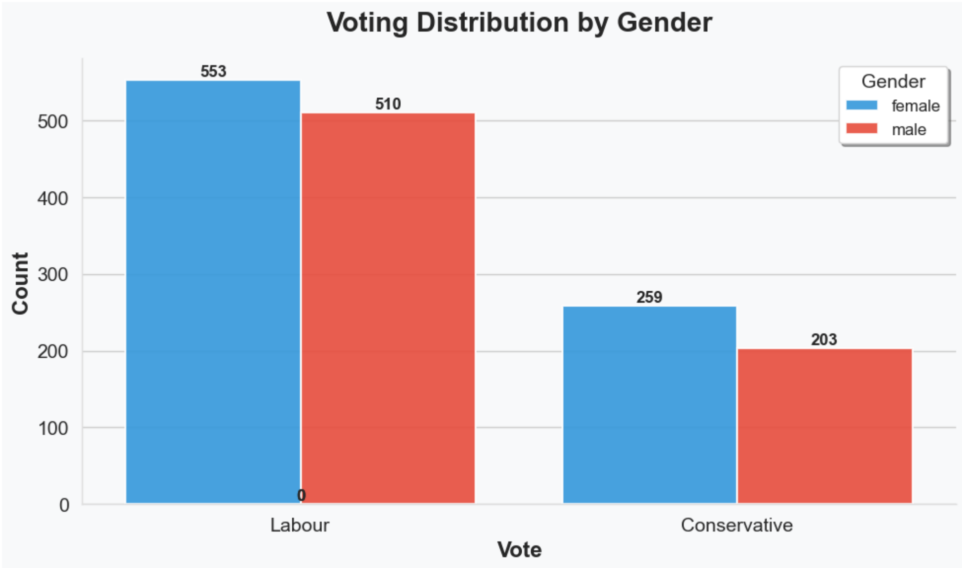

Figure 8: Categorical v/s Categorical Analysis

3.4 Outlier Analysis

The dataset was examined for potential outliers that could affect model performance by visualizing box-plot of variables.

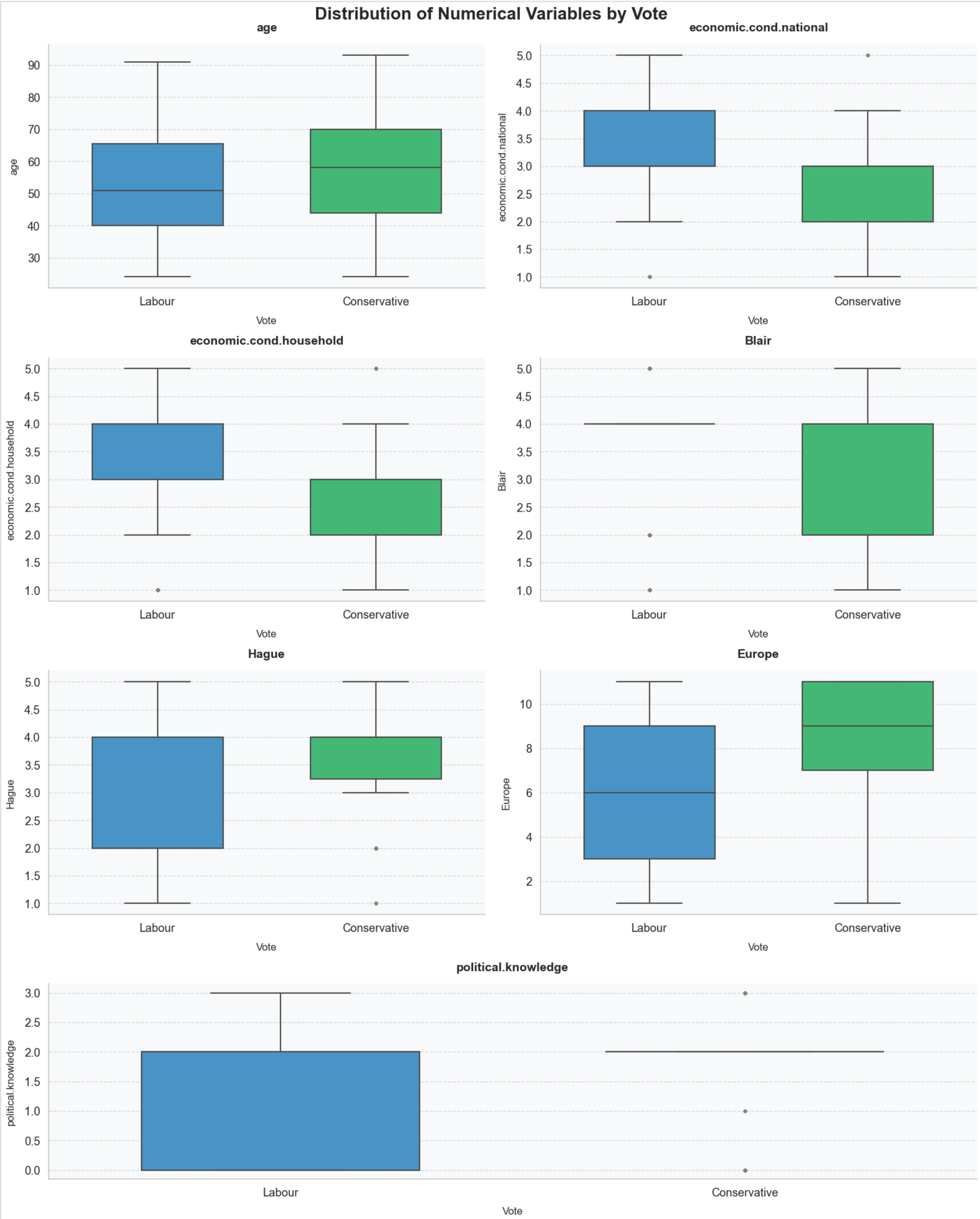

Figure 9: Function for treating outlier

Outliers were treated using the Winsorization method, where we defined the lower and upper bound and value less or greater than them will be replaced by lower and upper bound value respectively.

```python
def treat_outliers(df, col):
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1

    # Define outlier thresholds
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Removing outlier only if there are error
    # df = df[(df[col] >= lower_bound) & (df[col] <= upper_bound)]

    # Outlier treatment
    df[col] = np.where(df[col] > upper_bound, upper_bound, df[col])
    df[col] = np.where(df[col] < lower_bound, lower_bound, df[col])
```

Figure 10: Function for treating outlier

Figure 11: Box plots after outlier treatmant

3.5 Summary of EDA Insights

3.5.1 Removed Unnecessary Columns

During the data preprocessing stage, certain features were identified as redundant or irrelevant based on domain knowledge and statistical analysis. These features were removed to improve model efficiency and reduce noise.

3.6 No Null Values, So No Null Value Treatment

A check for missing values using the `df.isnull().sum()` function revealed that no null values were present in the dataset. As a result, no imputation or missing value treatment was required, ensuring that all data remained intact for analysis.

3.6.1 No Multicollinearity

Multicollinearity was assessed using:

- A correlation heatmap to visualize relationships between features.

Since no strong correlations were found between independent variables, all retained features contributed uniquely to the model.

3.6.2 Gaussian Distribution of Age Variable

A histogram and Kernel Density Estimate (KDE) plot confirmed that the `age` variable followed a near-normal distribution, as shown in Figure 5. This normality is beneficial for models like Logistic Regression that assume a Gaussian distribution of features.

3.6.3 Outlier Handling

Outliers were detected using boxplots and the Interquartile Range (IQR) method. Handling outliers was performed as follows:

- Extreme values were either **capped** using Winsorization or **removed** if significantly skewed.

The final dataset was well-prepared for model training, ensuring robust and unbiased predictions.

4 Data Pre-Processing

4.1 Data Encoding

Categorical variables were encoded using:

- Label encoding for ordinal variables (Education Level, Media Exposure)

Used `LabelEncoder` class from `sklearn.preprocessing` to replace categorical values in vote and gender column.

```python
# We'll need something called LableEncoder
from sklearn.preprocessing import LabelEncoder

encoder_gender = LabelEncoder()
df["gender"] = encoder_gender.fit_transform(df["gender"])
encoder_gender.classes_, df["gender"].unique()
```

✓ 0.5s

```
(array(['female', 'male'], dtype=object), array([0, 1]))
```

```python
encoder_votes = LabelEncoder()
df["vote"] = encoder_votes.fit_transform(df["vote"])
encoder_votes.classes_, df["vote"].unique()
```

✓ 0.0s

```
(array(['Conservative', 'Labour'], dtype=object), array([1, 0]))
```

Figure 12: Box plots before outlier treatmant

4.2 Train-Test Split

The dataset was split into training and testing sets to evaluate model performance:

Table 5: Data Split Parameters

Parameter	Value
Training set size	80% (1214 samples)
Testing set size	20% (304 samples)
Random state	42 (for reproducibility)

5 Model Development and Evaluation

As there are two classes to predict i.e., `Labour` and `Conservative`, so this model does not predict negative or positive classes. We will focus on improving the accuracy of the model.

5.1 Logistic Regression Model

A logistic regression model was implemented as a baseline classifier.

Table 6: Logistic Regression Performance

Metric	Value
Accuracy	0.8066
Precision	0.8369
Recall	0.9028
F1-Score	0.8686
AUC-ROC	0.8642

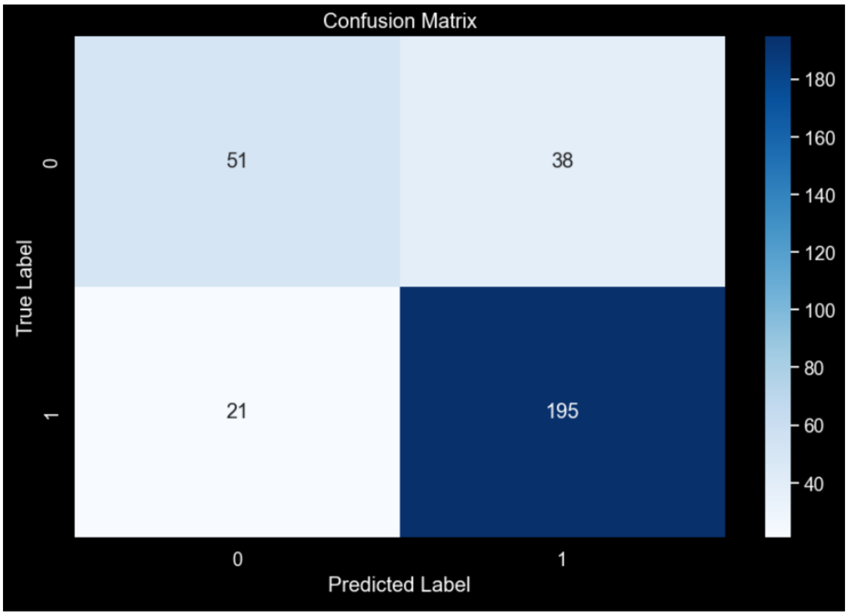

Figure 13: Confusion Matrix of Logistic Regression

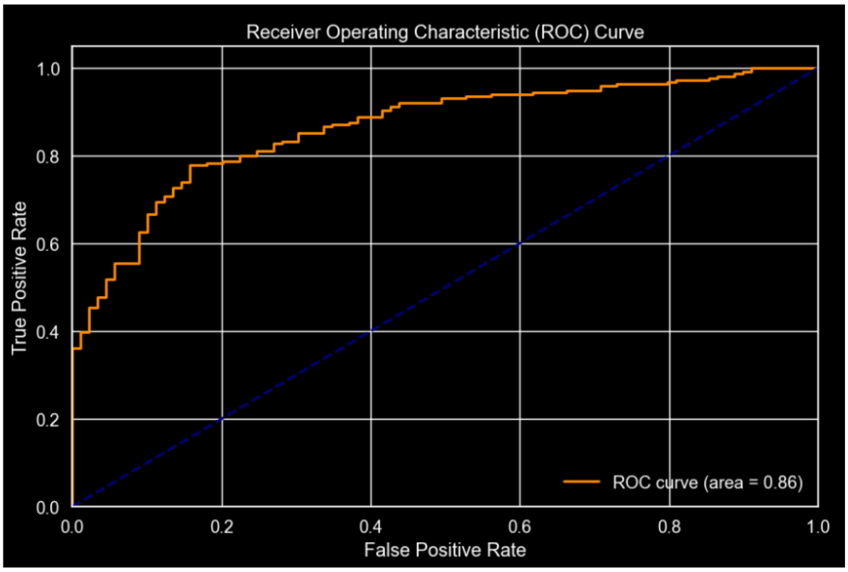

Figure 14: ROC curve of Logistic Regression

5.2 Decision Tree Model

A decision tree classifier was implemented and optimized using grid search.

Table 7: Decision Tree Parameters

Parameter	Value
max_depth	5
min_samples_split	10
min_samples_leaf	2
criterion	gini

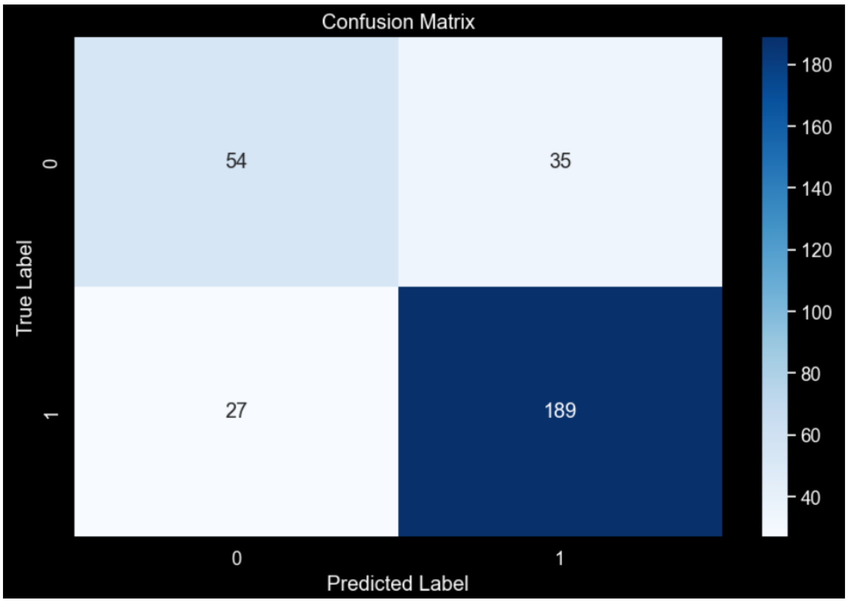

Figure 15: Confusion Matrix of Decision Tree

Table 8: Decision Tree Performance

Metric	**Value**
Accuracy	0.7967
Precision	0.8438
Recall	0.8750
F1-Score	0.8591
AUC-ROC	0.8405

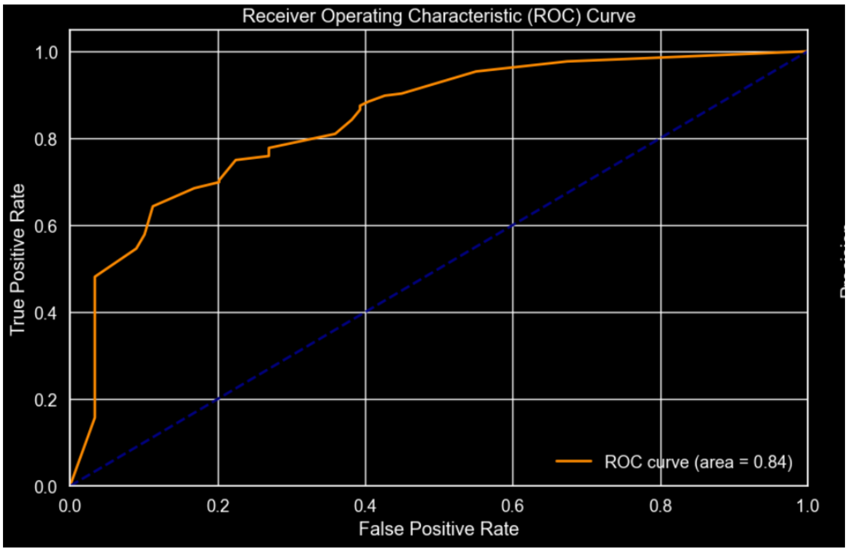

Figure 16: ROC Curve of Decision Tree

5.3 Random Forest Model

A random forest classifier was implemented to improve upon the decision tree model.

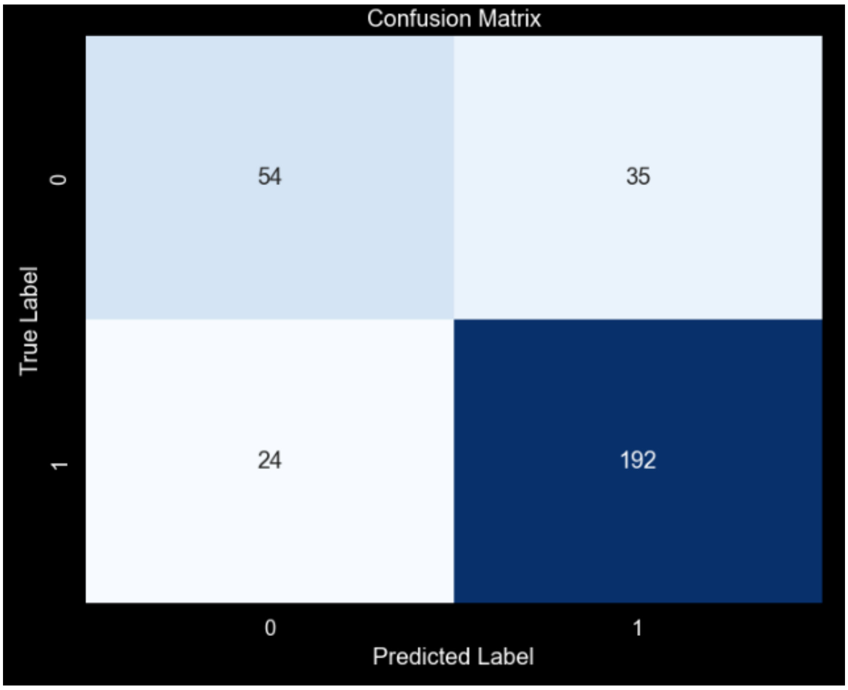

Figure 17: Confusion Matrix of Random Forest Classifier

Table 9: Random Forest Performance

Metric	Value
Accuracy	0.8066
Precision	0.8458
Recall	0.8889
F1-Score	0.8668
AUC-ROC	0.8585

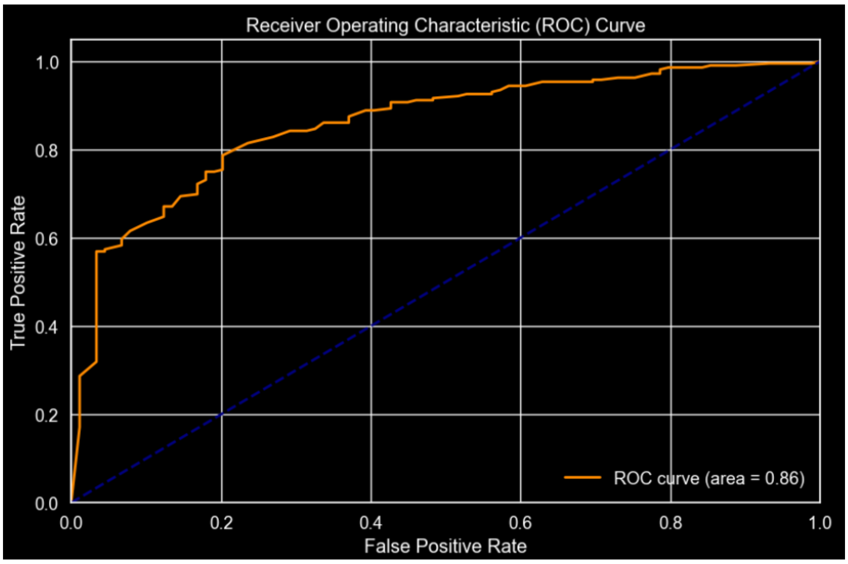

Figure 18: ROC Curve of Random Forest Classifier

5.4 XGBoost Model

An XGBoost classifier was implemented for its strong performance in various classification tasks.

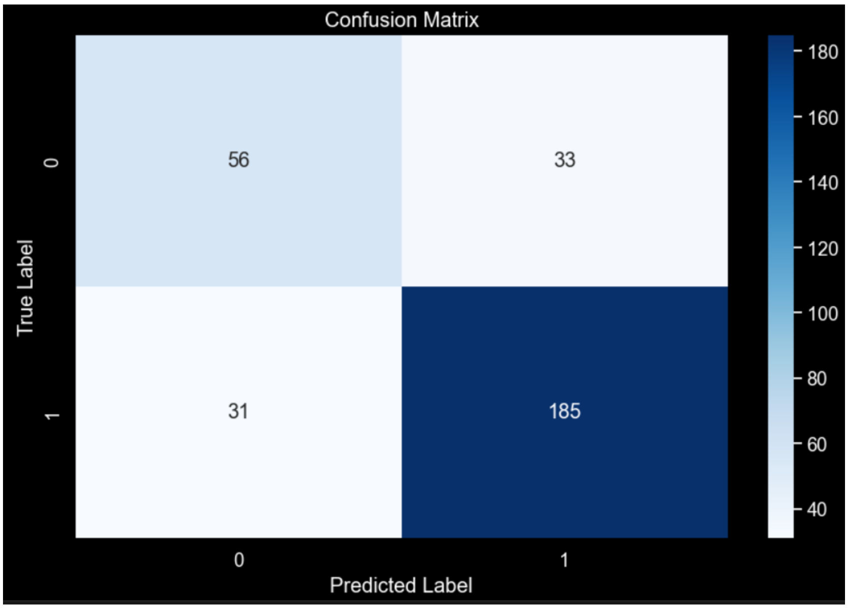

Figure 19: Confusion Matrix of XGBoost Classifier

Table 10: XGBoost Performance

Metric	**Value**
Accuracy	0.7902
Precision	0.8486
Recall	0.8565
F1-Score	0.8525
AUC-ROC	0.8368

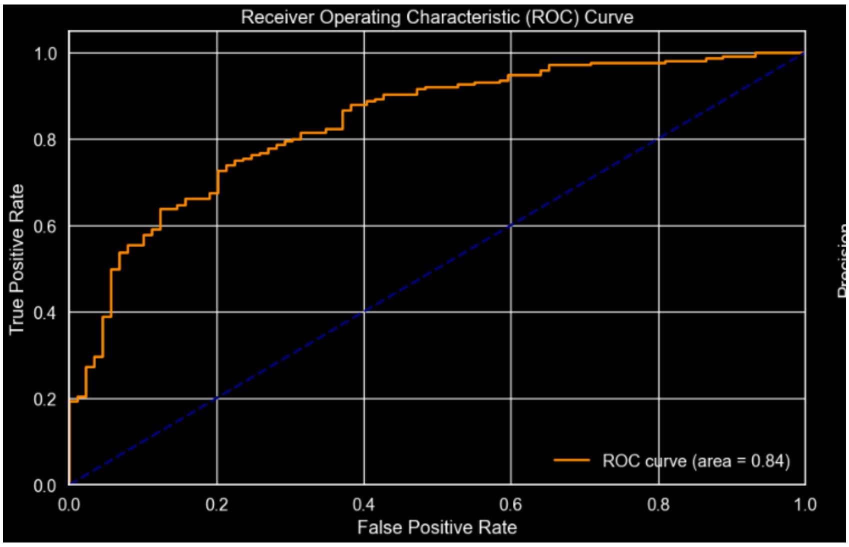

Figure 20: ROC Curve of XGBoost Classifier

5.5 AdaBoost Model

An AdaBoost classifier was implemented to evaluate its performance on the election dataset.

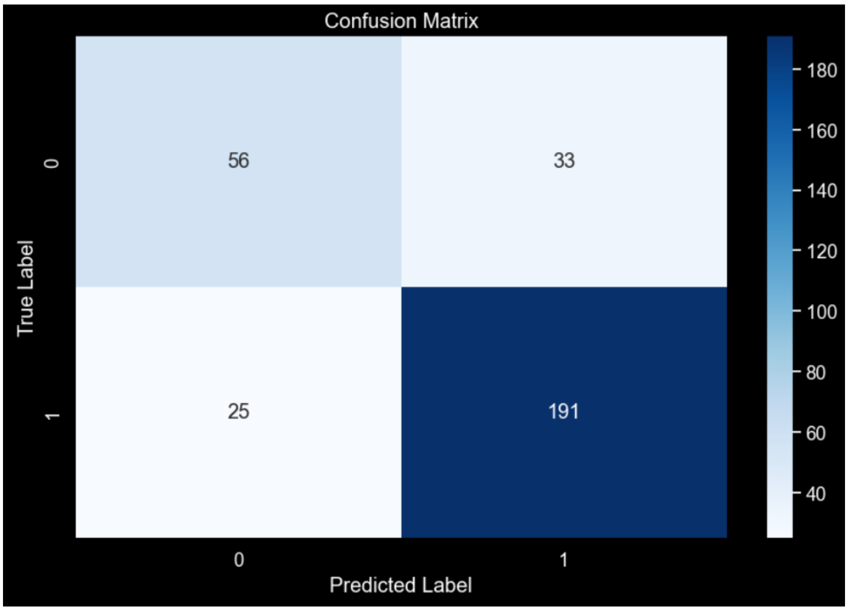

Figure 21: Confusion Matrix of AdaBoost Classifier

Table 11: AdaBoost Performance

Metric	Value
Accuracy	0.8098
Precision	0.8527
Recall	0.8843
F1-Score	0.8682
AUC-ROC	0.8611

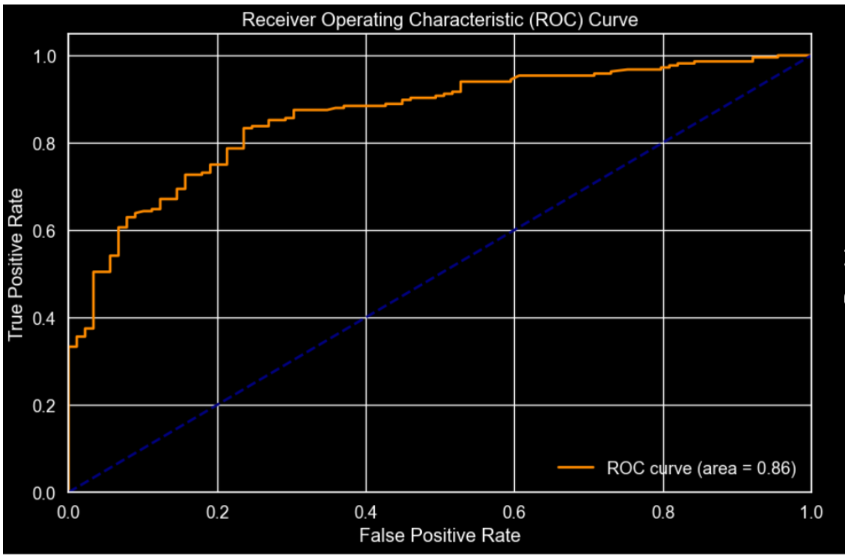

Figure 22: ROC Curve of AdaBoost Classifier

5.6 Gradient Boosting Model

A Gradient Boosting classifier was implemented to further evaluate boosting algorithms.

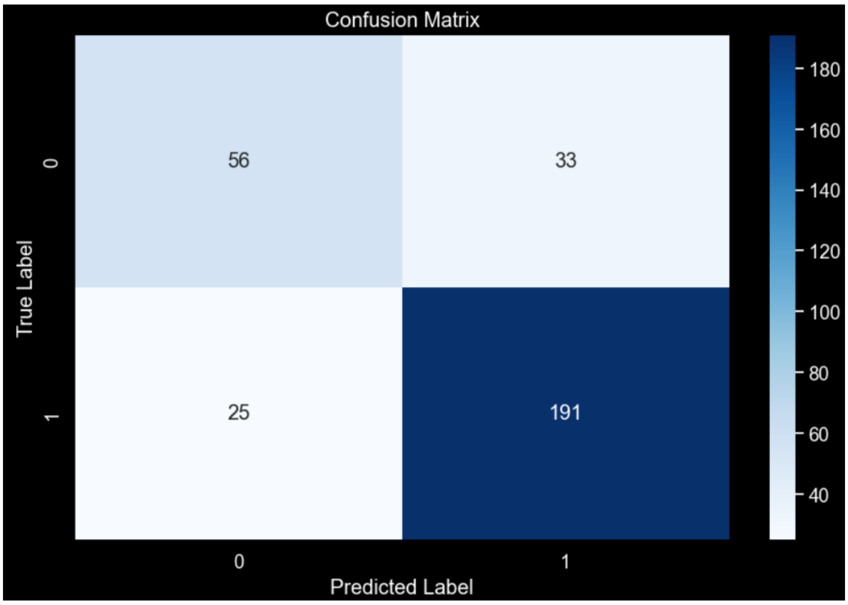

Figure 23: Confusion Matrix of GradientBoost Classifier

Table 12: Gradient Boosting Performance

Metric	Value
Accuracy	0.8098
Precision	0.8527
Recall	0.8843
F1-Score	0.8682
AUC-ROC	0.8611

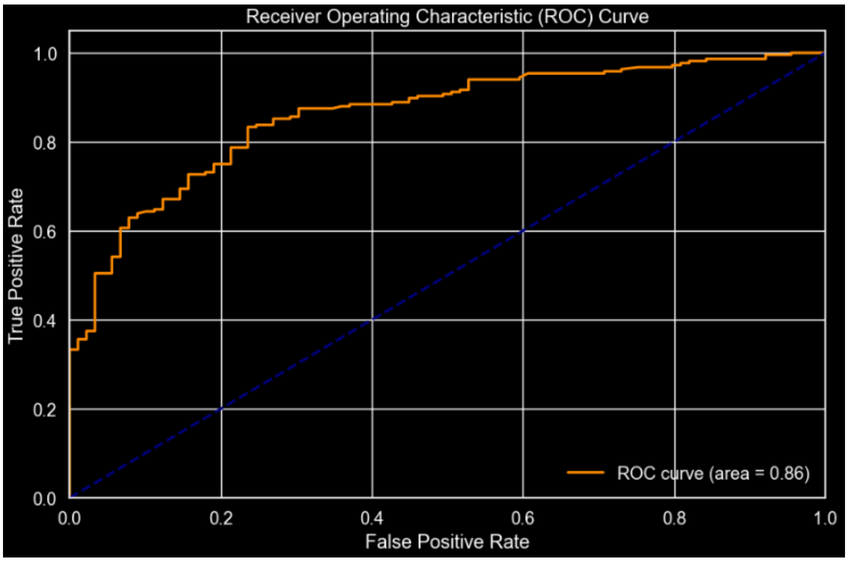

Figure 24: ROC Curve of GradientBoost Classifier

5.7 K-Nearest Neighbors Model

A K-Nearest Neighbors classifier was implemented to evaluate its performance on the election dataset.

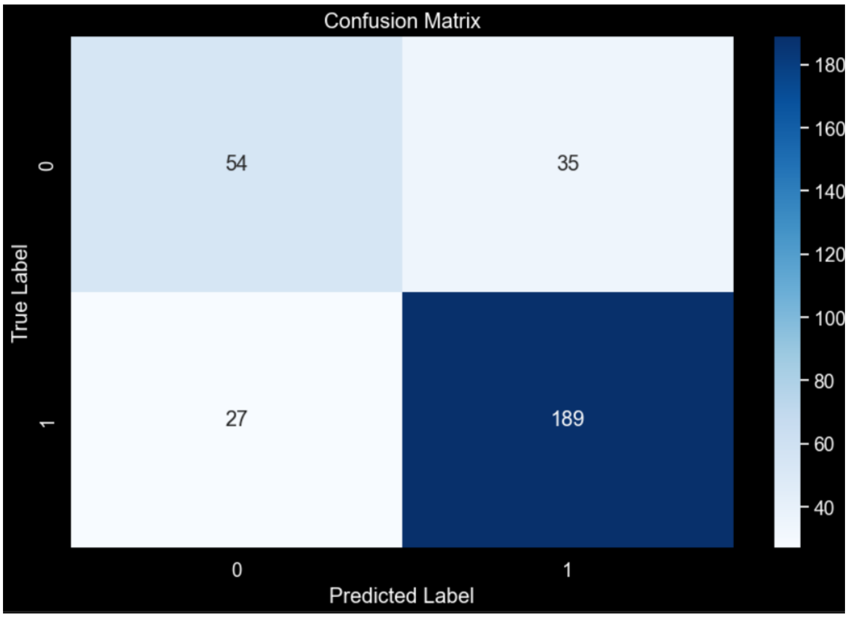

Figure 25: Confusion Matrix of kNN Classifier

Table 13: KNN Performance

Metric	**Value**
Accuracy	0.7967
Precision	0.8438
Recall	0.8750
F1-Score	0.8591
AUC-ROC	0.8405

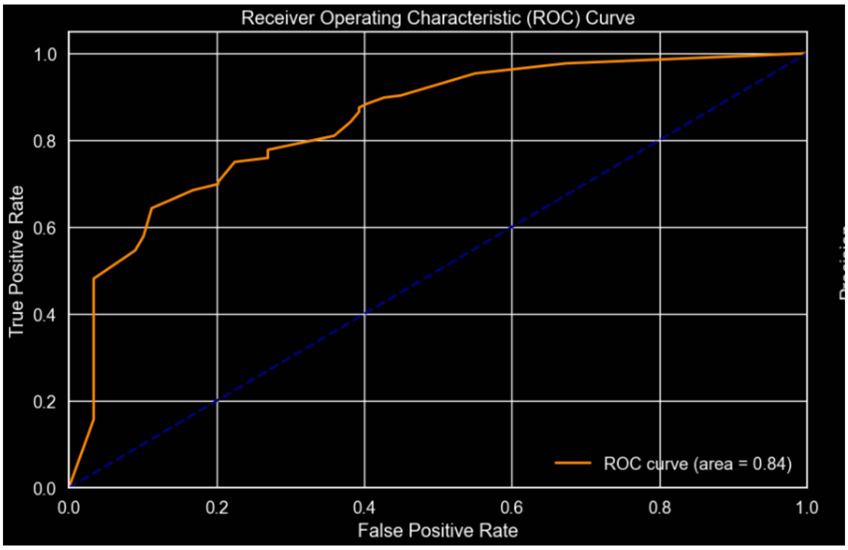

Figure 26: ROC Curve of kNN Classifier

6 Model Comparison and Selection

6.1 Performance Comparison of All Models

The performance of all implemented models was compared to determine the most suitable model for voter prediction.

Table 14: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.8066	0.8369	0.9028	0.8686
Decision Tree	0.7967	0.8438	0.8750	0.8591
Random Forest	0.8066	0.8458	0.8889	0.8668
XGBoost	0.7902	0.8486	0.8565	0.8525
AdaBoost	0.8098	0.8527	0.8843	0.8682
Gradient Boosting	0.8098	0.8527	0.8843	0.8682
KNN	0.7967	0.8438	0.8750	0.8591

Based on the model performance, we can derieve following conclusion for model selection:

- Selected **AdaBoost** or **Gradient Boosting** for best overall results.
- Chose **Logistic Regression** for focusing on recall (capturing all Labour voters) is most important.

7 Insights

7.1 Vote Distribution

- **Labour**: 69.7%
- **Conservative**: 30.3%
- Labour has significantly more supporters in this dataset.

7.2 Demographics

- **Age**: Ranges from 24 to 93, with a mean of 54 years.
- **Gender**: Almost equal distribution (46.8% Female, 53.2% Male).

7.3 Economic Conditions

- **National Economic Condition Average**: 3.26 (on a scale of 1-5).
- **Household Economic Condition Average**: 3.16 (similar scale).
- Most people rate economic conditions as neutral/slightly positive.

7.4 Leadership Ratings

- **Blair (Labour Leader)**: Average rating of 3.33 (higher).
- **Hague (Conservative Leader)**: Average rating of 2.75.

7.5 Europe Sentiment

- **Mean Score**: 6.73 (scale 1-11).
- Higher scores indicate Eurosceptic views.

7.6 Political Knowledge

- **Mean Score**: 1.54 (scale 0-3).
- Many respondents have moderate political knowledge.