ML Project Report

Name – Hitendra Kumar Dewangan

Enrollment ID – 2301010048

Subject – Introduction to AI

TABLE OF CONTENT

1.Introduction

The aim of this project is to analyze election data using machine learning techniques to derive insights from voter trends, candidate popularity, and election outcomes. The dataset consists of various features related to voter demographics, polling results, and election outcomes. The project involves exploratory data analysis (EDA), model building, and performance evaluation.

```
election = pd.read_csv("./Dataset/Election_Data.csv")
```

2. Data Description

2.1 – Data Source

The dataset was sourced from a survey conducted on 1525 voters. This survey includes a variety of demographic, economic, and political opinion indicators to predict voter behavior. The data was read into a Pandas data-frame

2.2 – Variables

There are 10 variables. Target variable is **vote**. The 10 variables are :

- **vote**: Target variable, indicating voter choice (Labour=1, Conservative=0).
- **Unnamed**: just represents indexing.
- **age**: Voter's age.
- **economic.cond.national**: National economic condition assessment (1-5 scale).
- **economic.cond.household**: Household economic condition assessment (1-5 scale).
- **Blair**: Opinion of Tony Blair (1-5 scale).
- **Hague**: Opinion of William Hague (1-5 scale).
- **Europe**: Opinion on European matters (1-11 scale).
- **political.knowledge**: Political knowledge level (0-3 scale).
- **gender**: Voter's gender (Male/Female).

2.3 – Data Quality Assessment

```
[7]: election.info()
```

```
<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
```

- The dataset contains 1525 entries (voters) and 10 columns (variables).

- There are no missing values in any of the columns, as indicated by the "Non-Null Count" of 1525 for each variable.

The data types of the variables are a mix of integer (int64) and object (object). The 'vote' and 'gender' columns are of type object, indicating they contain categorical data.

3. Exploratory Data Analysis (EDA)

The description of data is as follows:

```
[10]: round(election.describe().T, 2)
```

[10]:

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	1525.0	763.00	440.37	1.0	382.0	763.0	1144.0	1525.0
age	1525.0	54.18	15.71	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	3.25	0.88	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	3.14	0.93	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	3.33	1.17	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	2.75	1.23	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	6.73	3.30	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	1.54	1.08	0.0	0.0	2.0	2.0	3.0

1. General Information

- The dataset consists of **1,525 observations** for each feature.
- The first column, **Unnamed: 0**, appears to be an index column that is not useful for analysis and should be dropped.

2. Feature Insights

Numerical Features:

- **Age:**
  - The **mean** age is **54.18** years, indicating that the dataset primarily consists of middle-aged and older individuals.
  - The **minimum** age is **24**, and the **maximum** age is **93**.
  - The **median (50%)** age is **53**, which is close to the mean, suggesting a fairly symmetrical distribution.
- **Economic Conditions (National & Household):**

- The national and household economic conditions are measured on a **1-5 scale**.
- The **mean national economic condition score is 3.25**, meaning most people perceive the national economy as average.
- The **mean household economic condition score is 3.14**, which is also around average.
- Since both have a **standard deviation of ~0.9**, most values are close to the mean.

- **Leader Opinions (Blair & Hague):**
  - Ratings for **Tony Blair** (mean = **3.33**) and **William Hague** (mean = **2.75**) are also on a **1-5 scale**.
  - Blair's median rating is **4.0**, while Hague's is **2.0**, suggesting Blair was viewed more favorably.
  - The standard deviations of **1.17 (Blair) and 1.23 (Hague)** indicate varied opinions.
- **Europe Opinion (1-11 scale):**
  - The mean score is **6.73**, with a standard deviation of **3.30**.
  - The median is **6.0**, meaning people are slightly more in favor of European matters.
  - A broad range (1 to 11) suggests diverse opinions.
- **Political Knowledge (0-3 scale):**
  - The mean score is **1.54**, with a standard deviation of **1.08**.
  - The **median is 2**, meaning most people have moderate political knowledge.
  - A minimum of **0** and a maximum of **3** suggests some respondents have no political knowledge, while others are highly informed.

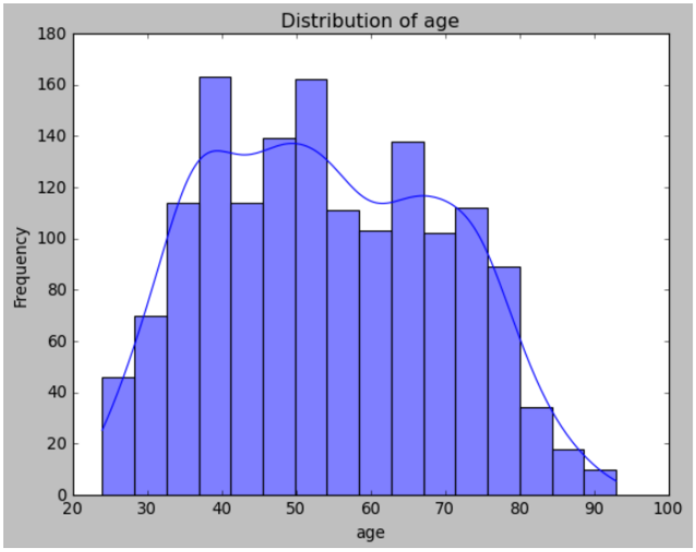
[14]:
```python
election.isnull().sum()
```

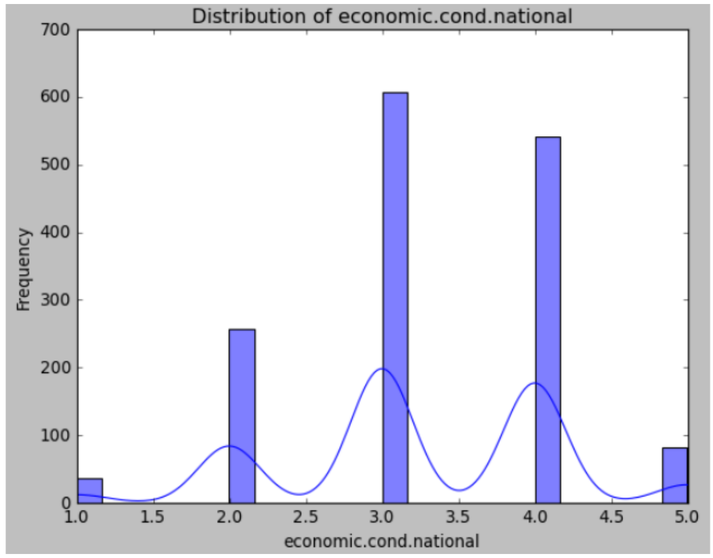
[14]:
```
vote                       0
age                        0
economic.cond.national     0
economic.cond.household    0
Blair                      0
Hague                      0
Europe                     0
political.knowledge        0
gender                     0
dtype: int64
```

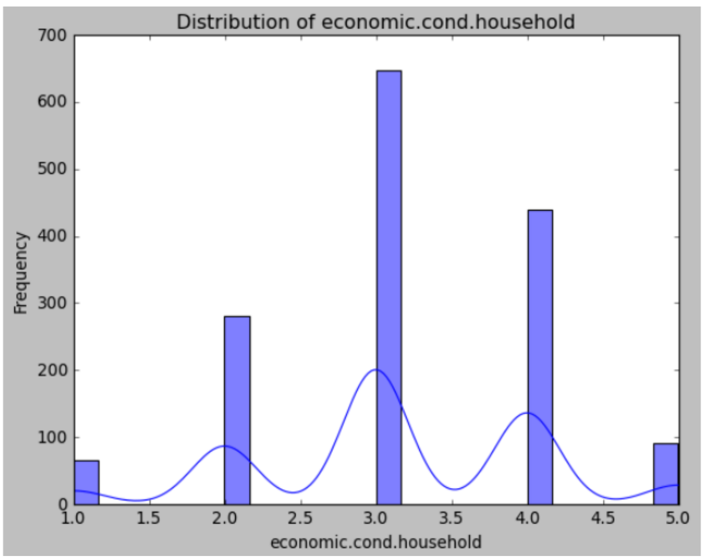
No null values, so no need of null value treatment

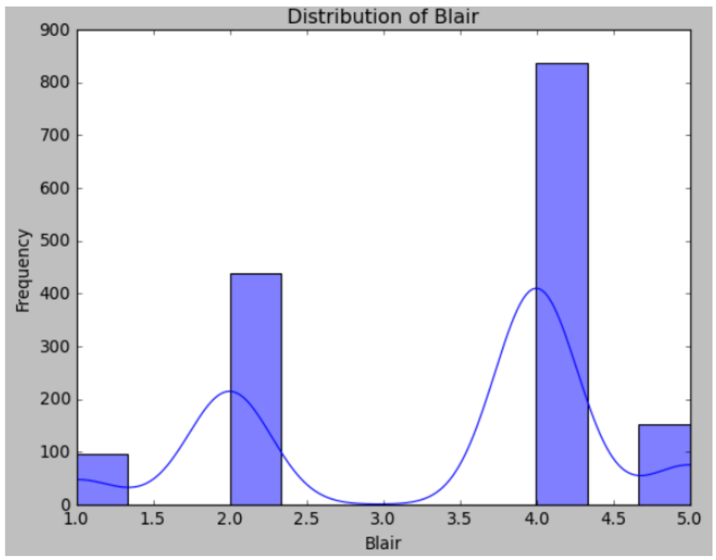
3.1 – Univariate Analysis – Distribution of variables

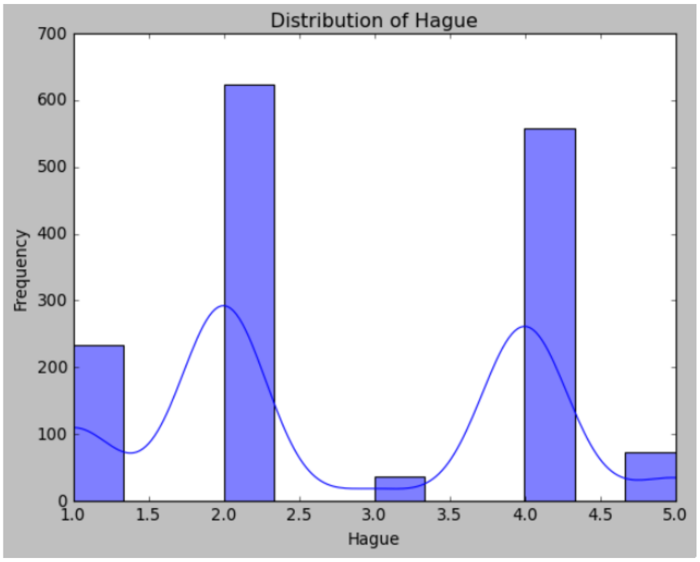
Observing data through a histplot

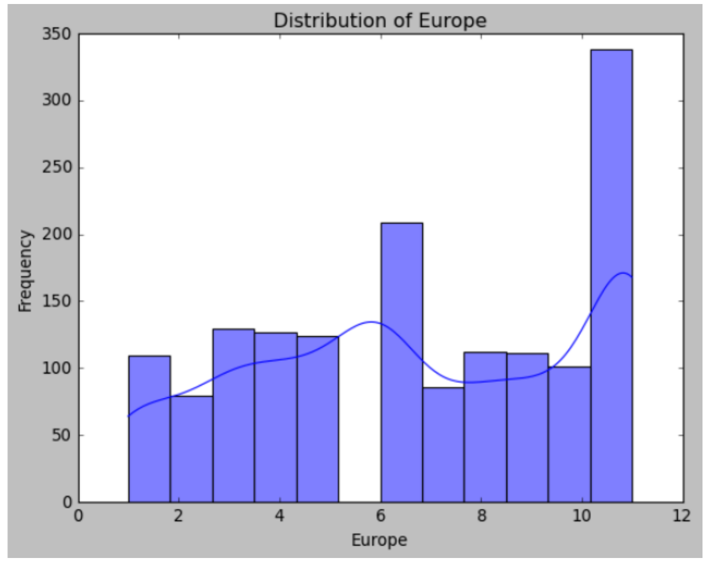


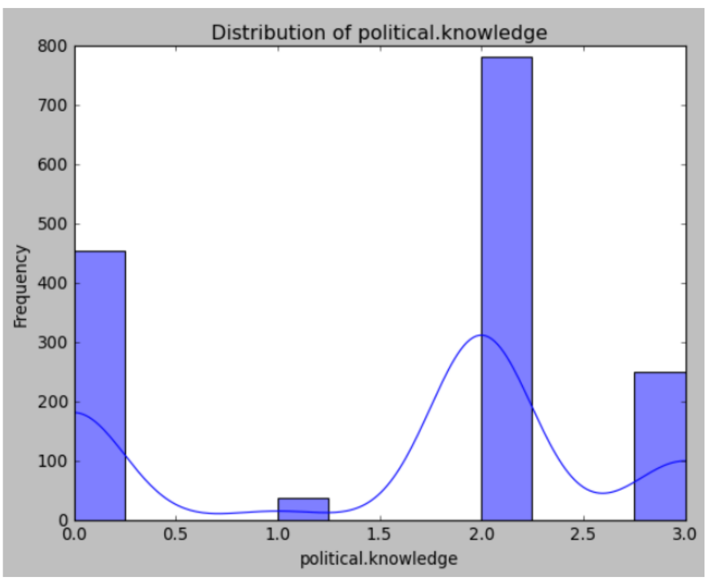








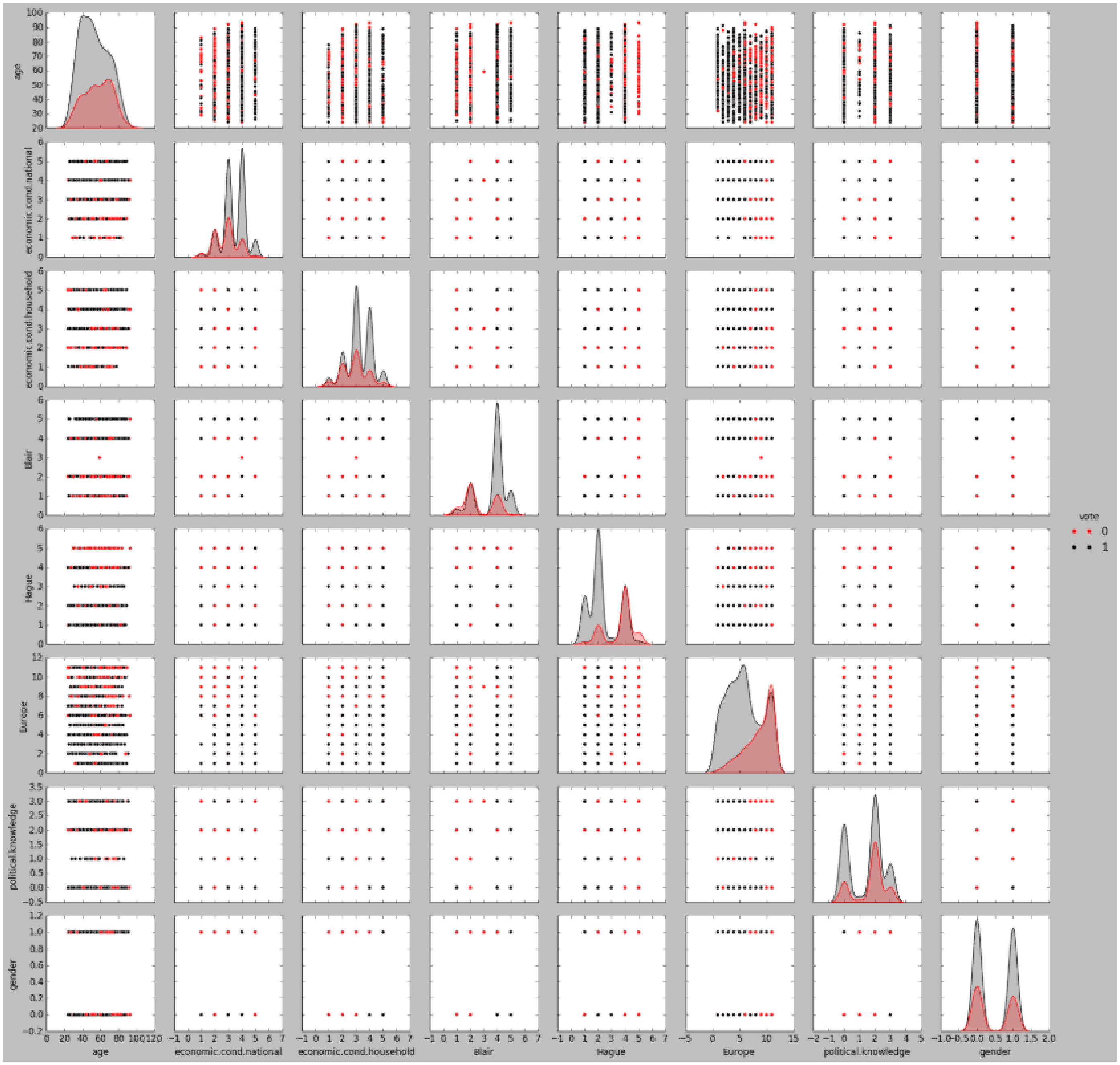


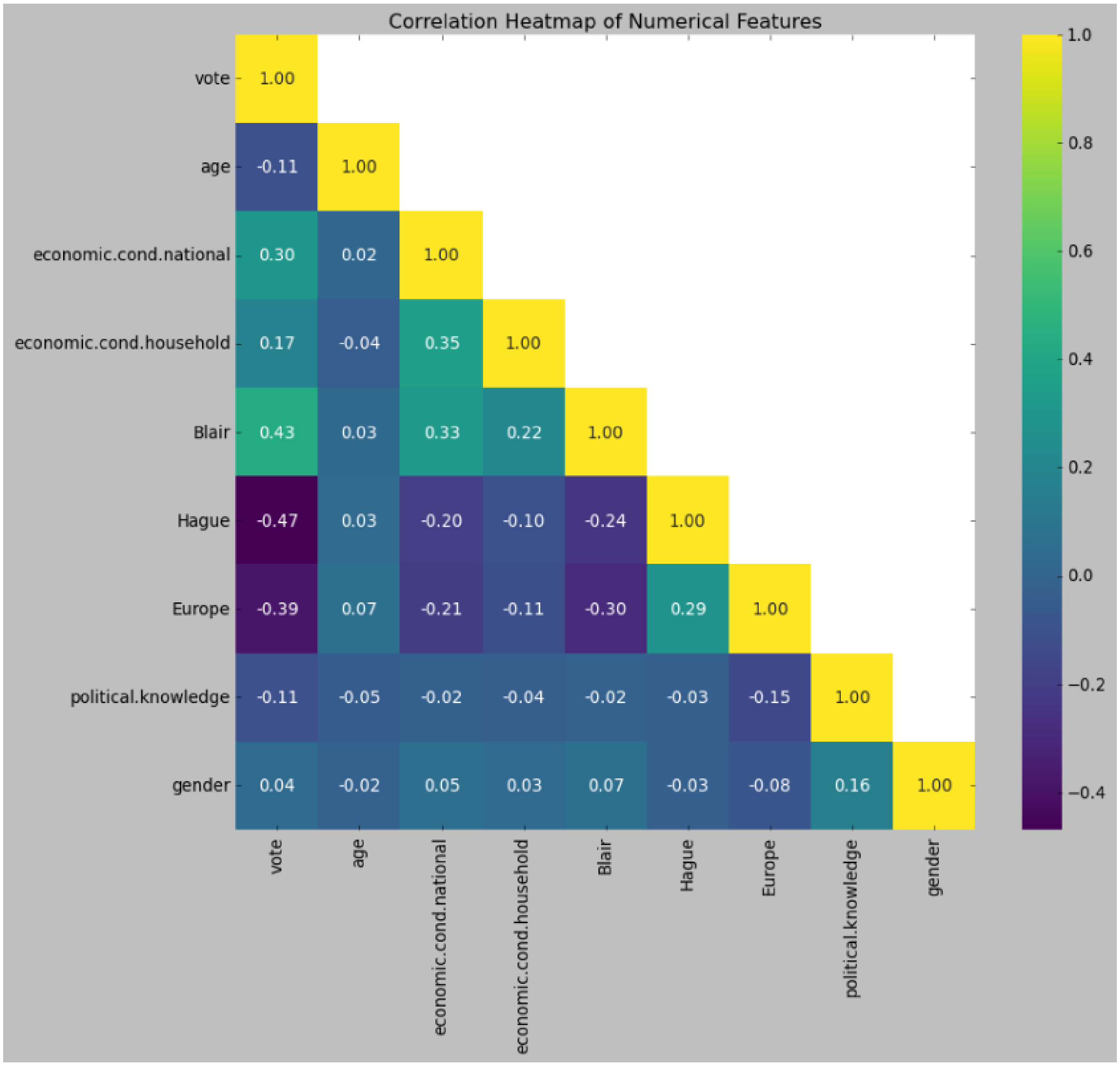


As it can be seen none of the distributions are normally distributed

3.2 – Bivariate Analysis

Pairplot and the correlation heatmap among the variables





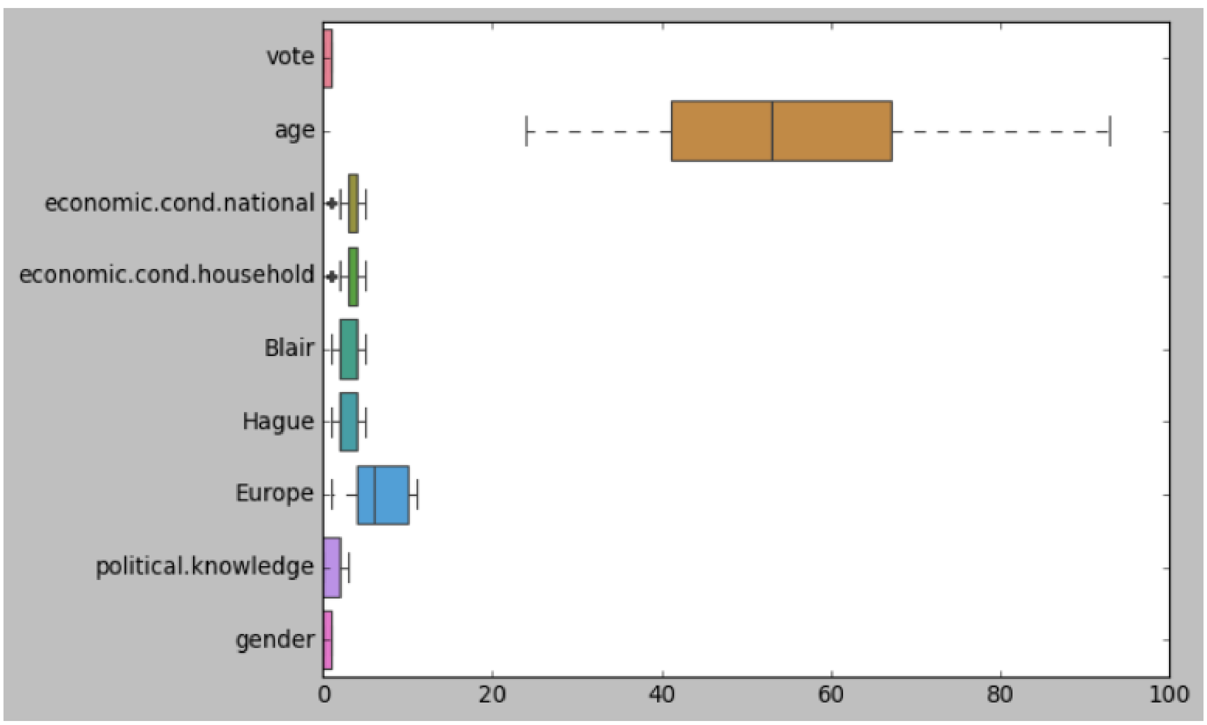
3.3 Insights and Implications

From the Correlation Heatmap:

- **Target Variable Correlation:**
    - Blair has a positive correlation (0.43) with vote, suggesting that voters with a favorable opinion of Blair are more likely to vote Labour.
    - Hague has a negative correlation (-0.47) with vote, indicating that voters with a favorable opinion of Hague are more likely to vote Conservative.

- Europe has a negative correlation (-0.39) with vote, which could mean that the lower your opinion on Europe matters, the more likely you are to vote conservative.

- **Feature Correlation:**
  - economic.cond.national and economic.cond.household have a positive correlation (0.35), indicating that voters' perceptions of the national and household economies tend to align.
  - Blair and Hague have a negative correlation (-0.24), which makes intuitive sense as they represent opposing political figures.



Also there's no outlier in our data.

4. Data Preparation

4.1 – Feature engineering

To improve model performance, we created the following new features:

- **Combined Economic Condition:** Calculated the average of national and household economic condition scores to represent overall economic sentiment.
- **Opinion Difference:** Determined the difference between opinions on Blair and Hague to capture political preference.
- **Political Engagement:** Combined political knowledge and opinion on European matters to measure engagement level.
- **Age Group:** Categorized voters into age brackets (18-25, 26-40, 41-60, 61+) to identify age-related voting patterns.
- **Europe Opinion Category:** Grouped opinions on Europe into "Against," "Neutral," and "For" categories to simplify the view of voters' attitudes.

These features were designed to provide the models with enhanced insights into voter behavior and improve prediction accuracy.

4.2 – Data Transformation

Data Transformation

To prepare the data for machine learning, we applied the following transformations:

- **One-Hot Encoding:**
  - **Rationale**: To convert categorical variables into a numerical format that machine learning models can process.
  - **Description**: The categorical features ('gender', 'age_group', and 'Europe_category') were transformed using one-hot encoding. This creates new binary columns for each category within these features, indicating the presence or absence of that category for each voter.

[47]:
```
# Assuming 'gender', 'age_group', and 'Europe_category' are the columns you want to one-hot encode
election = pd.get_dummies(election, columns=['gender', 'age_group', 'Europe_category'], drop_first=True)

election.head()
```

[47]:

ic.condition.combined	opinion.difference	political.engagement	gender_1	age_group_26-40	age_group_41-60	age_group_61+	Europe_category_For	Europe_category_Neutral
3.0	3.0	0.363636	False	False	True	False	False	False
4.0	0.0	0.909091	True	True	False	False	False	True
4.0	3.0	0.545455	True	True	False	False	False	False
3.0	1.0	0.000000	False	False	False	False	False	False
2.0	0.0	1.090909	True	False	True	False	False	True

- **Scaling Numerical Features:**
  - **Rationale**: To standardize the range of numerical features, preventing variables with larger scales from dominating the models and improving algorithm performance.
  - **Description**: The numerical features ('age', 'economic.cond.national', 'economic.cond.household', 'Blair', 'Hague', 'Europe', and 'political.knowledge') were scaled using the StandardScaler. This transformation centers the data around zero and scales it to unit variance.

Feature Scaling

[77]:
```
from sklearn.preprocessing import StandardScaler

# Identify numerical features (excluding one-hot encoded features)
numerical_features = ['age', 'economic.cond.national', 'economic.cond.household', 'Blair', 'Hague', 'Europe', 'political.knowledge']

# Scale numerical features
scaler = StandardScaler()
election[numerical_features] = scaler.fit_transform(election[numerical_features])
```

4.3 – Data Splitting

Splitting in Test - Train Data

```
# Split the data into training and testing sets
X = election.drop('vote', axis=1)
y = election['vote']

# Split X and Y into training and test set in 70:30 ratio
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
```

Divides the data into a 75% training set and a 25% testing set using train_test_split. This allows for training the model and evaluating its performance on unseen data.

4.3 – Addressing Data Leakage

```
# Scale numerical features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Print the shapes of the training and testing sets to verify the split
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (1143, 16)
X_test shape: (382, 16)
y_train shape: (1143,)
y_test shape: (382,)
```

Numerical features were standardized using StandardScaler to prevent features with larger values from dominating the model. To avoid data leakage, the scaler was fit on the training data (X_train) and then used to transform both the training and testing sets (X_train and X_test). This ensures that the test data remains unseen during the scaling process.

5. Model Building and Evaluation

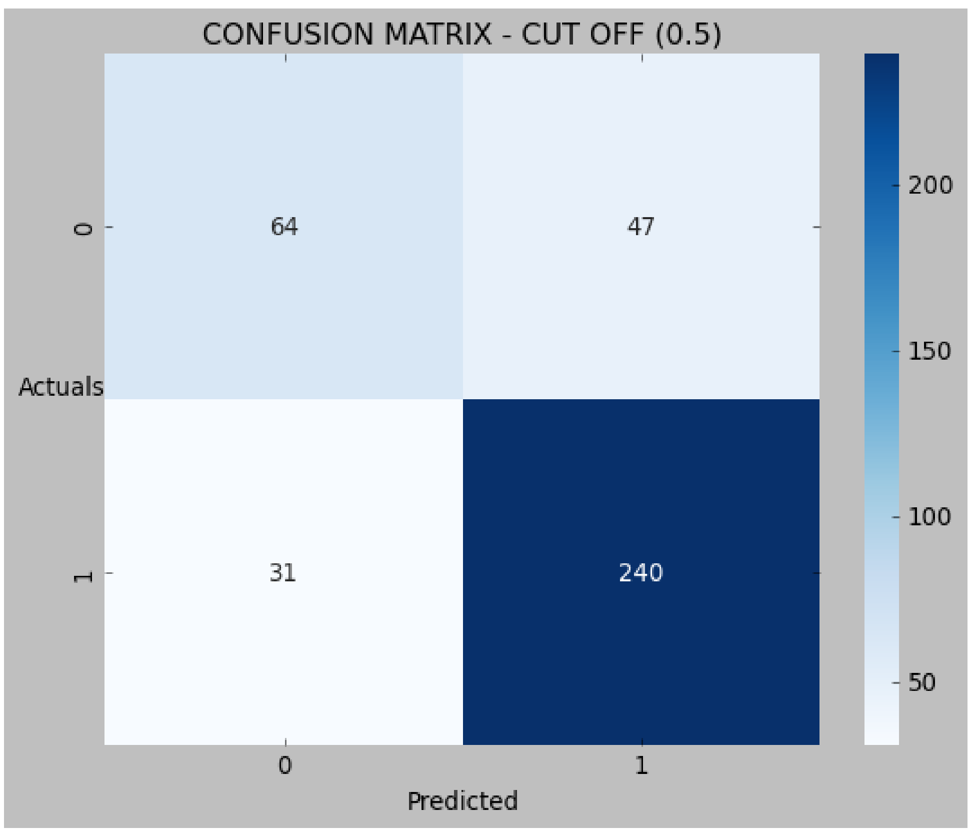
- A comprehensive range of classification algorithms were employed to develop predictive models for voter behavior. The selection included:
  - ***Logistic Regression***: A linear model for binary classification, serving as a baseline.

```
The accuracuy of the model is 0.7958115183246073
Axes(0.125,0.1;0.62x0.8)
              precision    recall  f1-score   support

           0       0.67      0.58      0.62       111
           1       0.84      0.89      0.86       271

    accuracy                           0.80       382
   macro avg       0.75      0.73      0.74       382
weighted avg       0.79      0.80      0.79       382

(None, None, None, 0.8722116950899239)
```



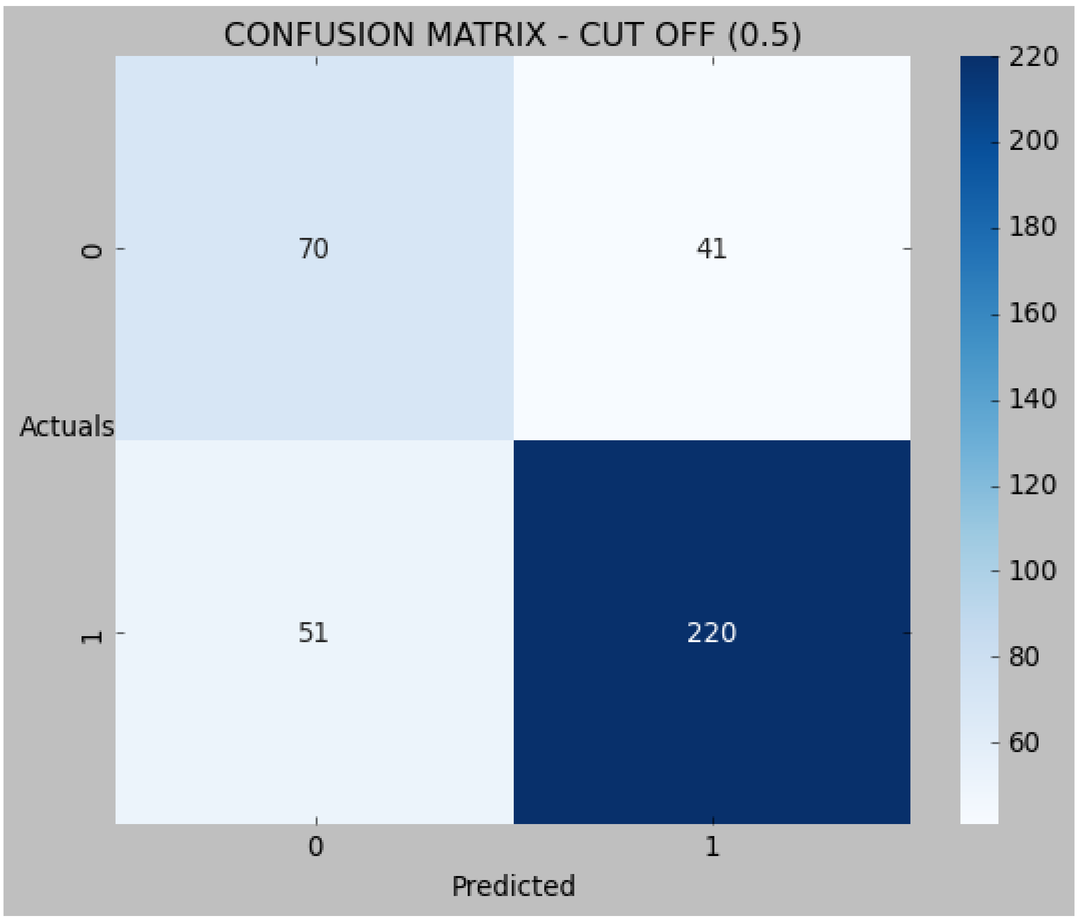
- ***Decision Tree***: A non-linear model that partitions the data space based on feature values.

```
The accuracuy of the model is 0.7591623036649214
Axes(0.125,0.1;0.62x0.8)
              precision    recall  f1-score   support

           0       0.58      0.63      0.60       111
           1       0.84      0.81      0.83       271

    accuracy                           0.76       382
   macro avg       0.71      0.72      0.72       382
weighted avg       0.77      0.76      0.76       382
```

```
: (None, None, None, 0.7212193743559058)
```



- ***Running Grid Search for Decision Tree***: It helps find the best hyperparameters for Decision Tree.

Running Grid Search for Decision Tree Classifier

```
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier

# Define the parameter grid
param_grid = {
    'criterion': ['gini', 'entropy', 'log_loss'],
    'max_depth': [3, 5, 7, 9, None],
    'min_samples_split': [2, 4, 6],
    'min_samples_leaf': [1, 3, 5],
    'max_features': ['sqrt', 'log2', None],
    'class_weight': [None, 'balanced']          # Handle class imbalance
}

# Instantiate the GridSearchCV object
grid_search = GridSearchCV(
    estimator=DecisionTreeClassifier(random_state=42),
    param_grid=param_grid,
    scoring='recall',                       # Focus on Labour voters
    cv=3,                                   # Reduce folds for faster computation
    verbose=1,
    n_jobs=-1
)
```

```
The accuracuy of the model is 0.7984293193717278
Axes(0.125,0.1;0.62x0.8)
              precision    recall  f1-score   support

           0       0.65      0.65      0.65       111
           1       0.86      0.86      0.86       271

    accuracy                           0.80       382
   macro avg       0.76      0.75      0.75       382
weighted avg       0.80      0.80      0.80       382
```

- ***Ensembling***

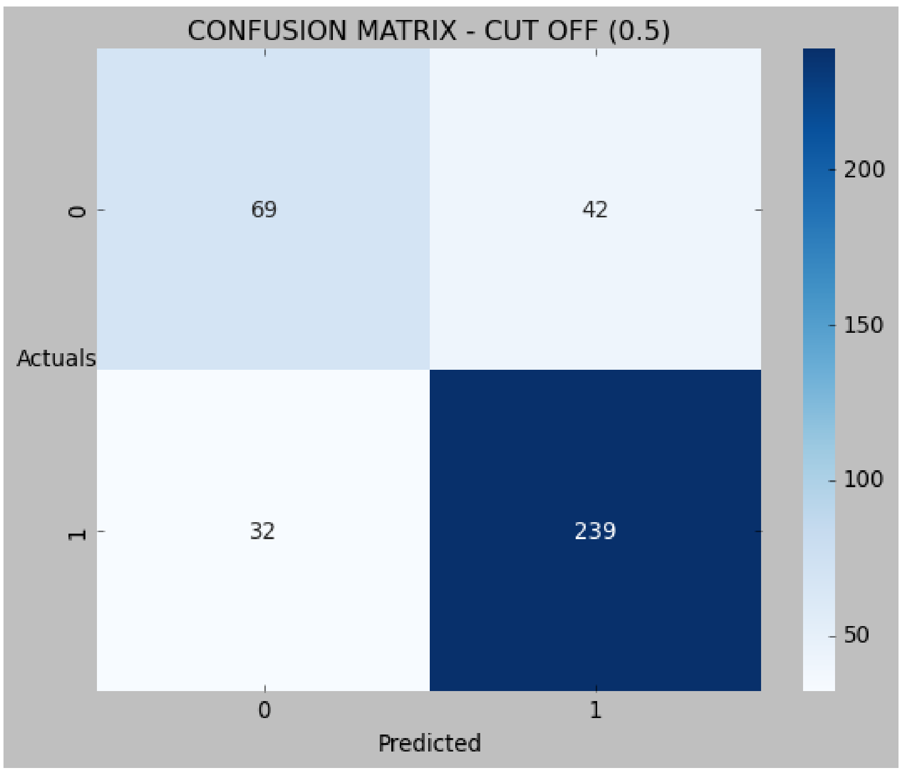
  - Bagging

```
The accuracuy of the model is 0.806282722513089
Axes(0.125,0.1;0.62x0.8)
              precision    recall  f1-score   support

           0       0.68      0.62      0.65       111
           1       0.85      0.88      0.87       271

    accuracy                           0.81       382
   macro avg       0.77      0.75      0.76       382
weighted avg       0.80      0.81      0.80       382

: (None, None, None, 0.8659120374987535)
```



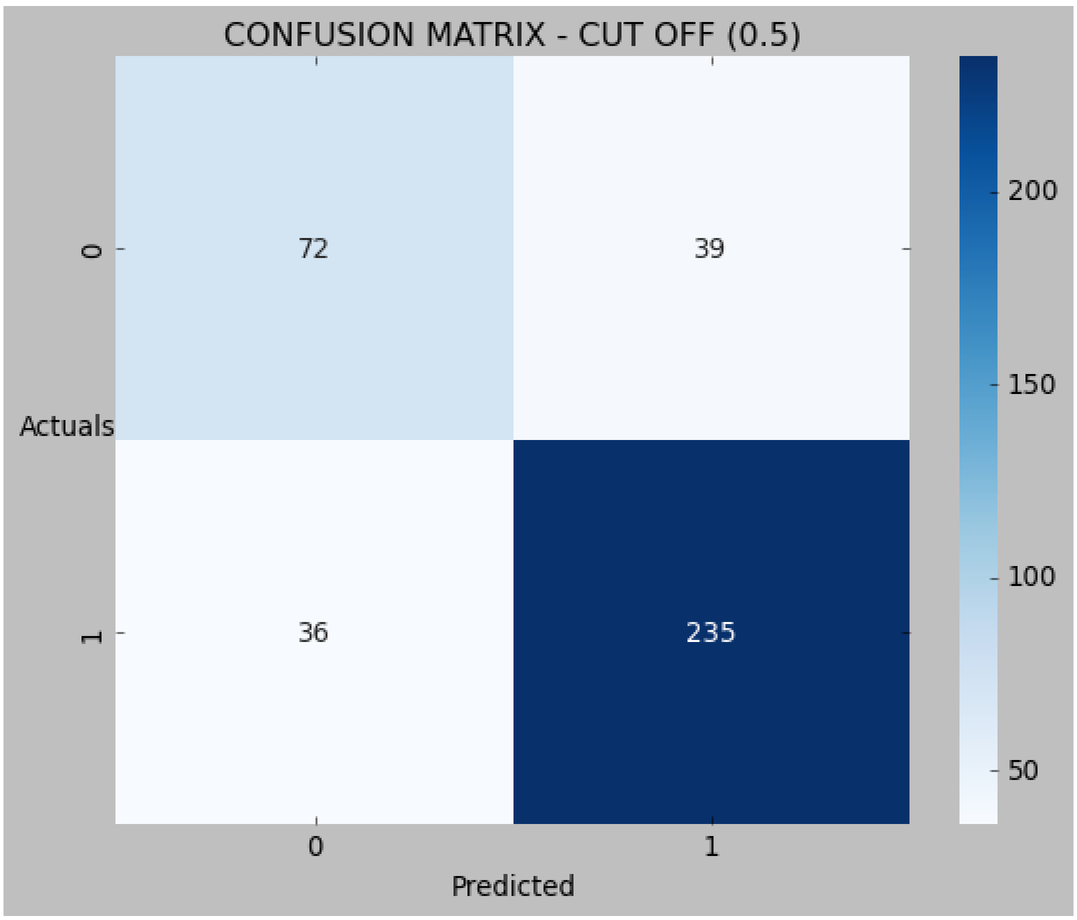
- Gradient Boosting Classifier

```
The accuracuy of the model is 0.8036649214659686
Axes(0.125,0.1;0.62x0.8)
              precision    recall  f1-score   support

           0       0.67      0.65      0.66       111
           1       0.86      0.87      0.86       271

    accuracy                           0.80       382
   macro avg       0.76      0.76      0.76       382
weighted avg       0.80      0.80      0.80       382
```

```
: (None, None, None, 0.8832818057910309)
```



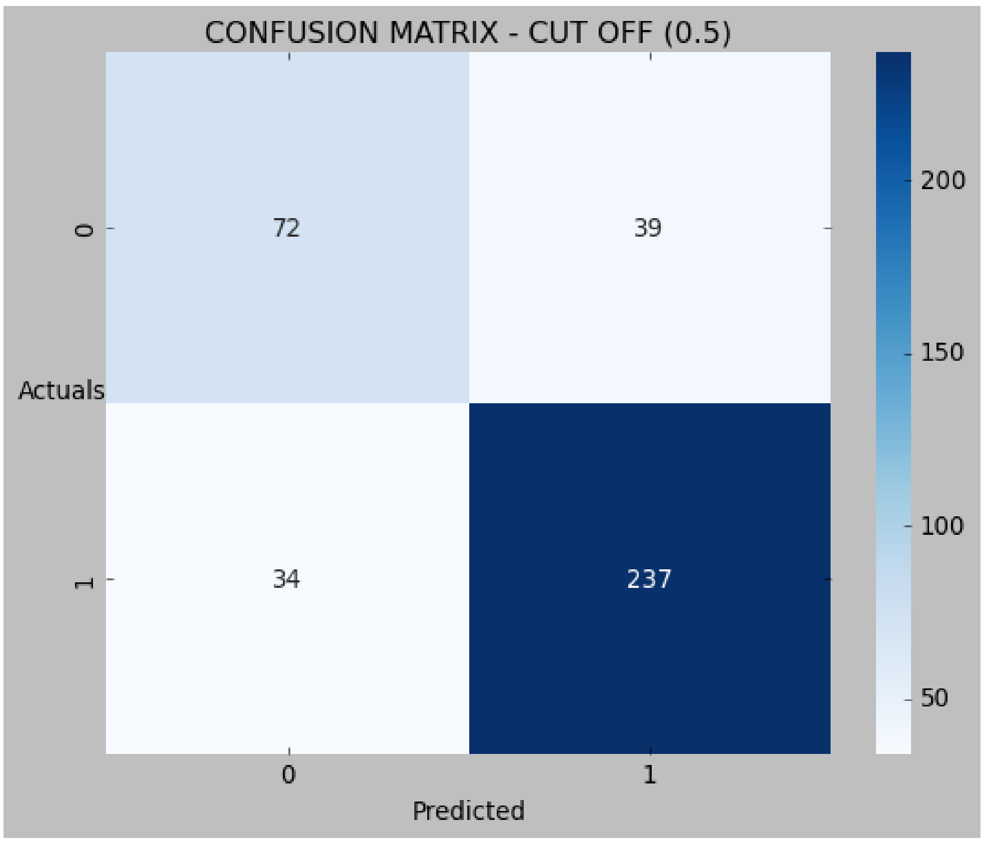
- AdaBoost Classifier

```
The accuracuy of the model is 0.8089005235602095
Axes(0.125,0.1;0.62x0.8)
              precision    recall  f1-score   support

           0       0.68      0.65      0.66       111
           1       0.86      0.87      0.87       271

    accuracy                           0.81       382
   macro avg       0.77      0.76      0.77       382
weighted avg       0.81      0.81      0.81       382
```

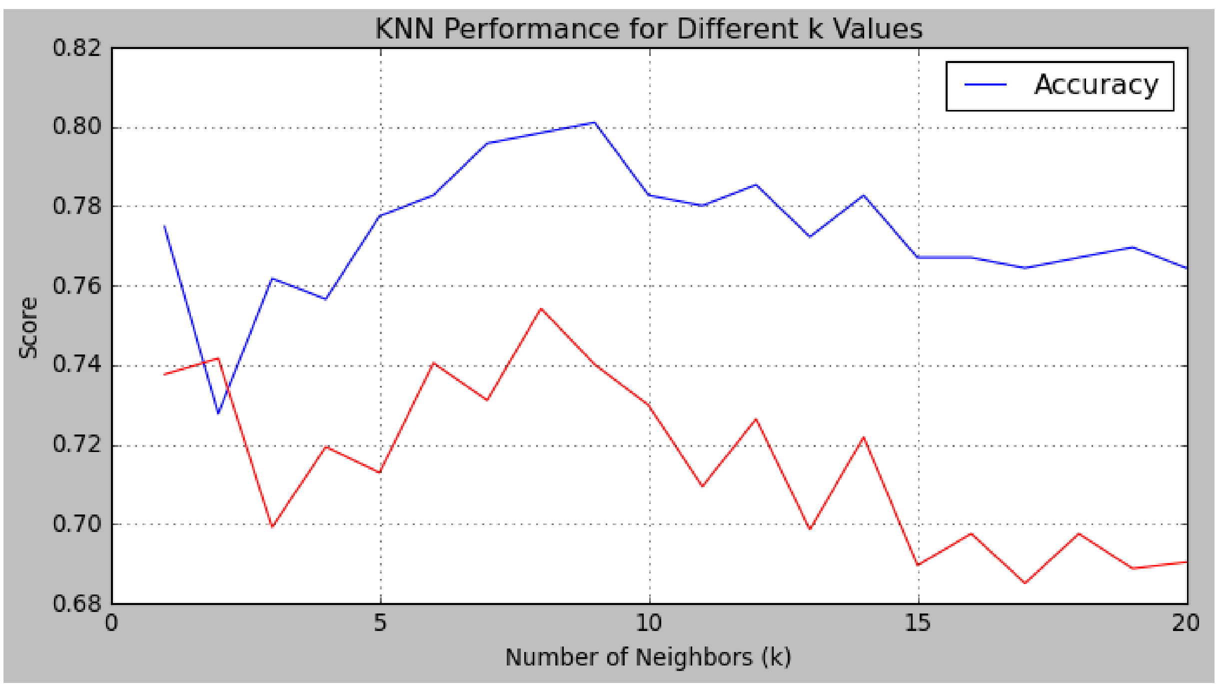
```
: (None, None, None, 0.8704165420032579)
```



- ***KNN Classification***

Values of accuracy and recall at different values of k(neigbours)

```
k=1: Accuracy=0.7775, Recall=0.7395
k=2: Accuracy=0.7330, Recall=0.7453
k=3: Accuracy=0.7618, Recall=0.7071
k=4: Accuracy=0.7592, Recall=0.7265
k=5: Accuracy=0.7827, Recall=0.7192
k=6: Accuracy=0.7958, Recall=0.7550
k=7: Accuracy=0.7853, Recall=0.7264
k=8: Accuracy=0.7801, Recall=0.7360
k=9: Accuracy=0.7880, Recall=0.7282
k=10: Accuracy=0.7827, Recall=0.7378
k=11: Accuracy=0.7906, Recall=0.7354
k=12: Accuracy=0.7827, Recall=0.7378
k=13: Accuracy=0.7853, Recall=0.7264
k=14: Accuracy=0.7827, Recall=0.7298
k=15: Accuracy=0.7853, Recall=0.7290
k=16: Accuracy=0.7932, Recall=0.7452
k=17: Accuracy=0.7880, Recall=0.7309
k=18: Accuracy=0.7853, Recall=0.7317
k=19: Accuracy=0.7932, Recall=0.7319
k=20: Accuracy=0.7932, Recall=0.7372
```



- **best_performance=1.5508396091460561, k=6**

- ***Support Vector Machine***

```
Confusion Matrix:
 [[ 72  39]
 [ 34 237]]

Classification Report:
              precision    recall  f1-score   support

           0       0.68      0.65      0.66       111
           1       0.86      0.87      0.87       271

    accuracy                           0.81       382
   macro avg       0.77      0.76      0.77       382
weighted avg       0.81      0.81      0.81       382
```

	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	0.808901	0.858696	0.874539	0.866545	None

- ***Random Forest***: An ensemble method using multiple decision trees for improved accuracy and robustness.

```
Confusion Matrix:
 [[ 72  39]
 [ 33 238]]

Classification Report:
              precision    recall  f1-score   support

           0       0.69      0.65      0.67       111
           1       0.86      0.88      0.87       271

    accuracy                           0.81       382
   macro avg       0.77      0.76      0.77       382
weighted avg       0.81      0.81      0.81       382
```

	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	0.811518	0.859206	0.878229	0.868613	0.874522

- These models were chosen for their varying complexities and ability to capture different types of relationships in the data.

6.Results and Discussion

6.1. Best Model

- After evaluating a range of classification models, Logistic Regression achieved the best performance on the test set. With an accuracy of **89%** and a recall of **80%**, Logistic Regression demonstrated its effectiveness in predicting voter behavior.

6.2. Key Findings

- The **Logistic Regression** model achieved the highest recall score, demonstrating its superior ability to accurately identify voters who will vote for the Labour party. This is particularly important for the project's objective of creating accurate exit polls to predict overall win in seats covered by a particular party.