Project - Machine Learning

(Different type of Models/Algorithm and Text Mining)

Submitted By: Hritika Vaishnav

INDEX

LIST OF TABLES

LIST OF FIGURES

Problem 1:

You are hired by one of the leading news channels CNBE who wants to analyse recent elections. This survey was conducted on 1525 voters with 9 variables. You have to build a model, to predict which party a voter will vote for on the basis of the given information, to create an exit poll that will help in predicting overall win and seats covered by a particular party.

Data Dictionary

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

1.1) Read the dataset. Describe the data briefly. Interpret the inferences for each. Initial steps like head() .info(), Data Types, etc . Null value check, Summary stats, Skewness must be discussed.

Import some libraries like Numpy, Pandas, Seaborn, Matplotlib, Linear Regression machine learning like LinearRegression,, LinearDiscriminantAnalysis, LogisticRegression, KNeighborsClassifier, GaussianNB, GridSearchCV, DecisionTreeClassifier, BaggingClassifier, AdaBoostClassifier, confusion_matrix, StandardScaler, accuracy_score, train_test_split, roc_curve, roc_auc_score, auc etc.

Load our data set, **Election_Data.xlsx**, and use the head() function to view the Top 5 data and the tail() function to view the bottom 5 data. Using the shape function, we can determine that there are 1525 rows

and 10 columns. Find out the characteristics of the columns using the info() method. The datatypes for the int64(8), and object(2) columns are present. There are no null values, and no duplicate values.

- **head()** it given by default top five data

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

Table 1 Top 5 data of dataset Election_Data.xlsx

- **tail()** it given by default bottom five data

	Unnamed: 0	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
1520	1521	Conservative	67	5	3	2	4	11	3	male
1521	1522	Conservative	73	2	2	4	4	8	2	male
1522	1523	Labour	37	3	3	5	4	2	2	male
1523	1524	Conservative	61	3	3	1	4	11	2	male
1524	1525	Conservative	74	2	3	2	4	11	0	female

Table 2 Bottom 5 data of dataset Election_Data.xlsx

- **shape** it tells numbers of rows and columns in given dataset.

```
(1525, 10)
```

Fig 1 Rows & columns of dataset Election_Data.xlsx

- **info()** it tells a concise summary of a DataFrame

```
<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 3 Information about the of dataset Election_Data.xlsx structure and content.

- No Null 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
```

Fig 2 No Null Values of dataset Election_Data.xlsx.

- No Duplicates Values.

Unnamed: 0	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender

Table 4 No Duplicates Values of dataset Election_Data.xlsx

➢ **Describe()** it tells summary of the central tendency, dispersion, and shape of the distribution of the data.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Unnamed: 0	1525.0	NaN	NaN	NaN	763.0	440.373894	1.0	382.0	763.0	1144.0	1525.0
vote	1525	2	Labour	1063	NaN	NaN	NaN	NaN	NaN	NaN	NaN
age	1525.0	NaN	NaN	NaN	54.182295	15.711209	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	NaN	NaN	NaN	3.245902	0.880969	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	NaN	NaN	NaN	3.140328	0.929951	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	NaN	NaN	NaN	3.334426	1.174824	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	NaN	NaN	NaN	2.746885	1.230703	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	NaN	NaN	NaN	6.728525	3.297538	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	NaN	NaN	NaN	1.542295	1.083315	0.0	0.0	2.0	2.0	3.0
gender	1525	2	female	812	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 5 Descriptive statistics of dataset Election_Data.xlsx

➢ **Drop unuseful column(s)** Unnamed:0

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   vote                    1525 non-null   object
 1   age                     1525 non-null   int64
 2   economic.cond.national  1525 non-null   int64
 3   economic.cond.household 1525 non-null   int64
 4   Blair                   1525 non-null   int64
 5   Hague                   1525 non-null   int64
 6   Europe                  1525 non-null   int64
 7   political.knowledge     1525 non-null   int64
 8   gender                  1525 non-null   object
dtypes: int64(7), object(2)
memory usage: 107.4+ KB
```

Table 6 Information about the of dataset Election_Data.xlsx

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
vote	1525	2	Labour	1063	NaN	NaN	NaN	NaN	NaN	NaN	NaN
age	1525.0	NaN	NaN	NaN	54.182295	15.711209	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	NaN	NaN	NaN	3.245902	0.880969	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	NaN	NaN	NaN	3.140328	0.929951	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	NaN	NaN	NaN	3.334426	1.174824	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	NaN	NaN	NaN	2.746885	1.230703	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	NaN	NaN	NaN	6.728525	3.297538	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	NaN	NaN	NaN	1.542295	1.083315	0.0	0.0	2.0	2.0	3.0
gender	1525	2	female	812	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 7 Descriptive statistics of dataset Election_Data.xlsx

- **Count the Value of object columns**

```
VOTE   2
Conservative     462
Labour          1063
Name: vote, dtype: int64
GENDER   2
male      713
female    812
Name: gender, dtype: int64
```

Table 8 Object columns value count of dataset Election_Data.xlsx

- **Skewness** To gain insights, into the distribution of variables it is beneficial to calculate their skewness.

```
Skewness:
age                     0.144621
economic.cond.national  -0.240453
economic.cond.household -0.149552
Blair                   -0.535419
Hague                    0.152100
Europe                  -0.135947
political.knowledge     -0.426838
dtype: float64
```

Fig 3 Skewness of dataset Election_Data.xlsx

Insights:

- The dataset contains a total of 1525 records.
- It consists of 10 columns, with 8 integer and 2 object (categorical) data types.

- The dataset does not have null values.
- There are no duplicate values present in the dataset.
- Drop the "Unnamed: 0" column from the dataset. The dataset now has 9 columns.
- Vote has two unique values: Labour and Conservative, which is also a dependent variable. Gender has two unique values: male and female.
- The skewness values for the numerical variables in the dataset are as follows: age: 0.145 (Positive skewness) economic.cond.national: -0.240 (Negative skewness) economic.cond.household: -0.150 (Negative skewness) Blair: -0.535 (Negative skewness) Hague: 0.152 (Positive skewness) Europe: -0.136 (Negative skewness) political.knowledge: -0.427 (Negative skewness)

1.2) Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also check for outliers (4 pts). Interpret the inferences for each (3 pts) Distribution plots(histogram) or similar plots for the continuous columns. Box plots. Appropriate plots for categorical variables. Inferences on each plot. Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.

➢ **Univariate Analysis of Numerical Features.**

1. **Distplot.**

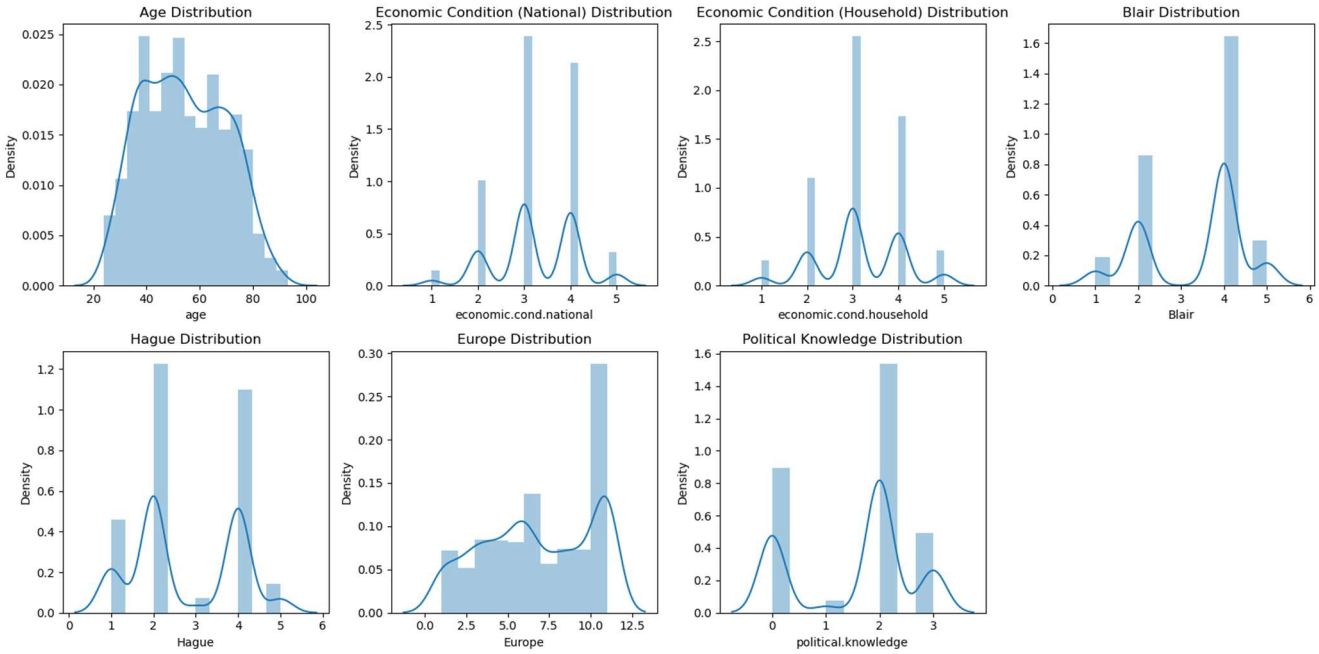

Fig 4 Distplot of Numerical features

- According to distplot, only age is normally distributed and has a slightly right-skewed distribution. There may be a slightly higher concentration of younger voters in the dataset compared to older voters.

2. Boxplot.

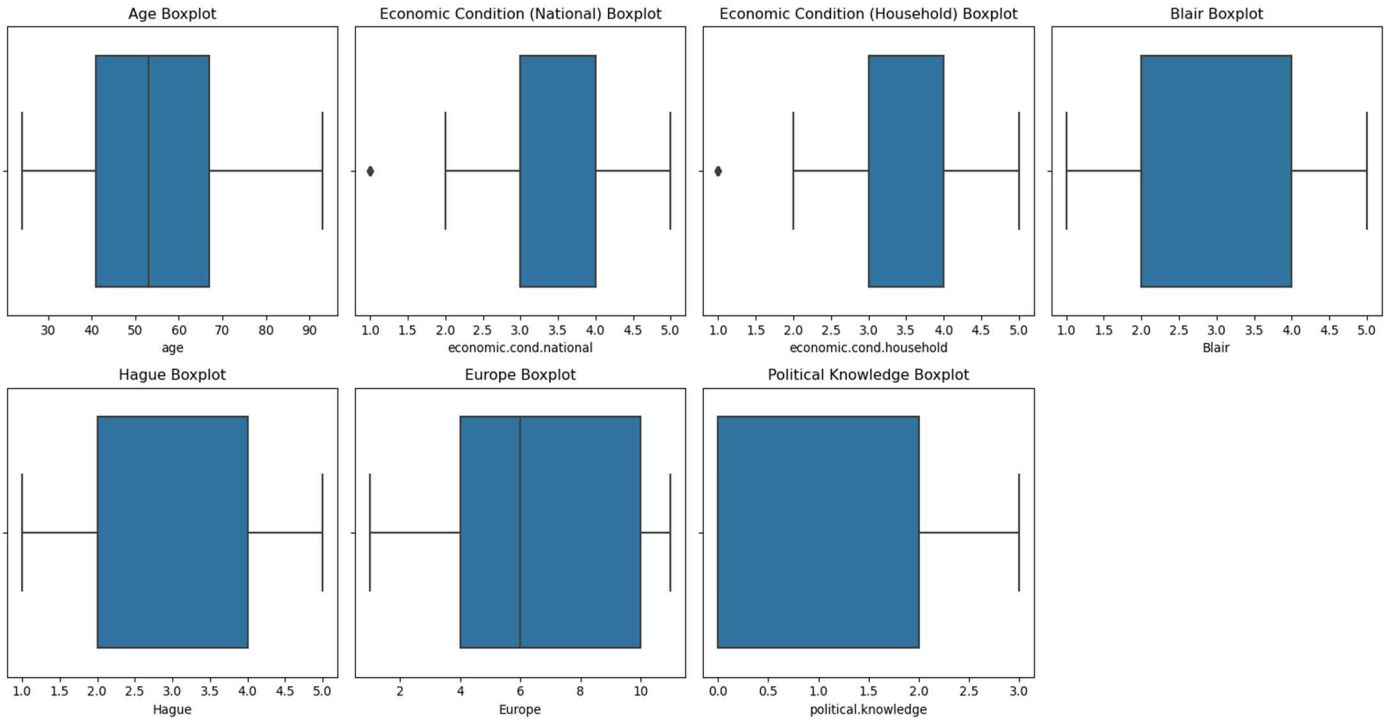

Fig 5 Boxplot of Numerical features

- According to Boxplot, economic.cond.national and economic.cond.household have outliers.

➢ Univariate Analysis of Object Features

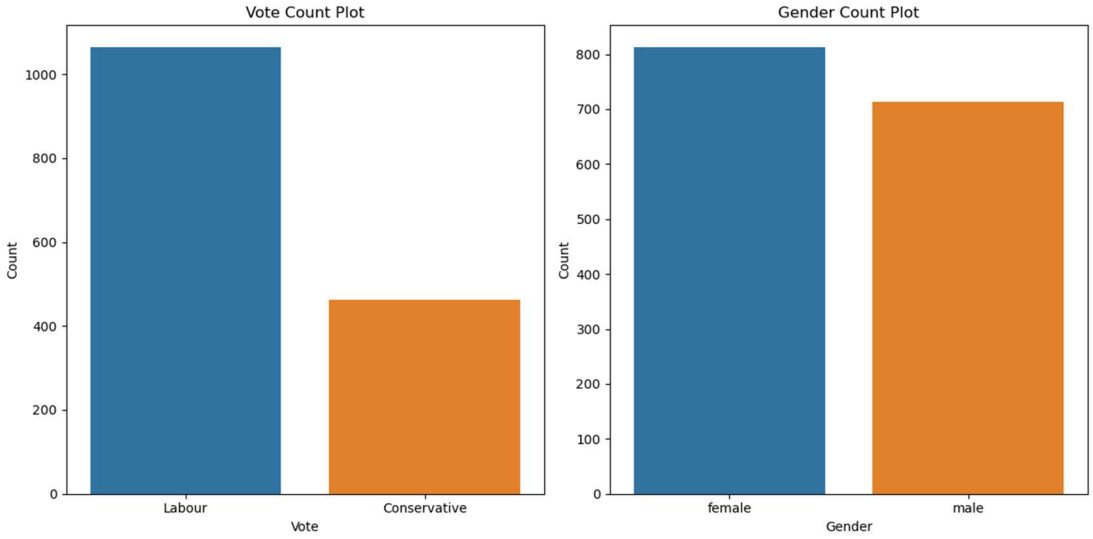

Fig 6 Subplot of Vote Count Plot and Gender Count Plot

- According to Subplot, the Labour gives more than 1000 votes, while the Conservative gives more than 400. In the comparison between Labour and Conservatives, Labor gets more votes.
- According to the subplot, the female gives more than 800 votes, while the male gives more than 700. In the comparison between females and males, females get more votes, but the difference in votes is less, or a smaller difference.

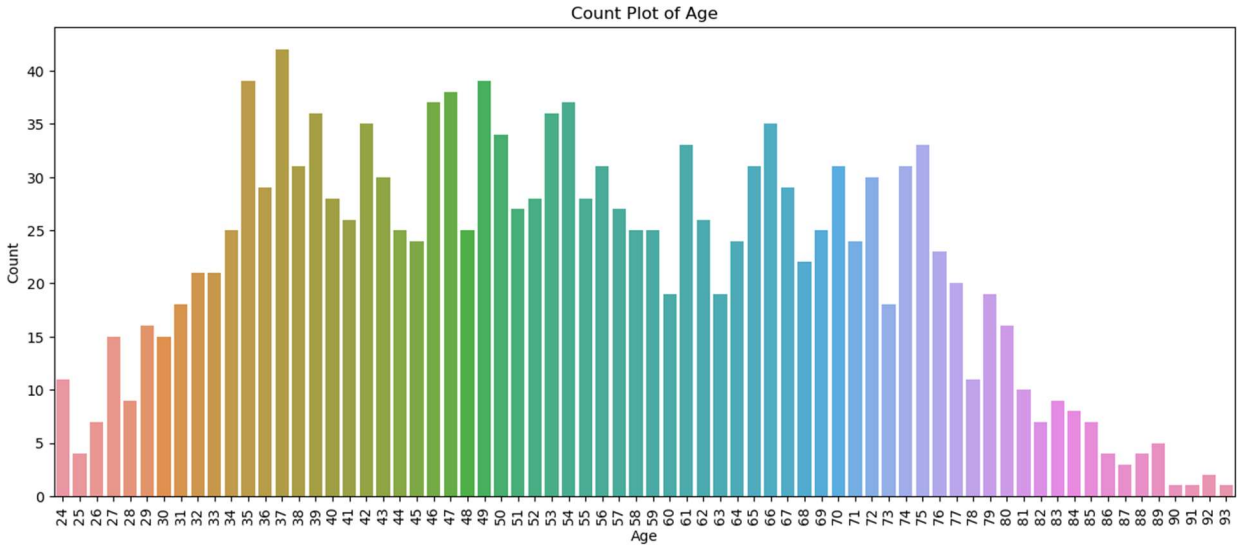

Fig 7 Count Plot of Age

- According to Countplot, the minimum age of a voter is 24 and the maximum age is 93. Here, voters between the ages of 37 and 90-93 cast the most votes.

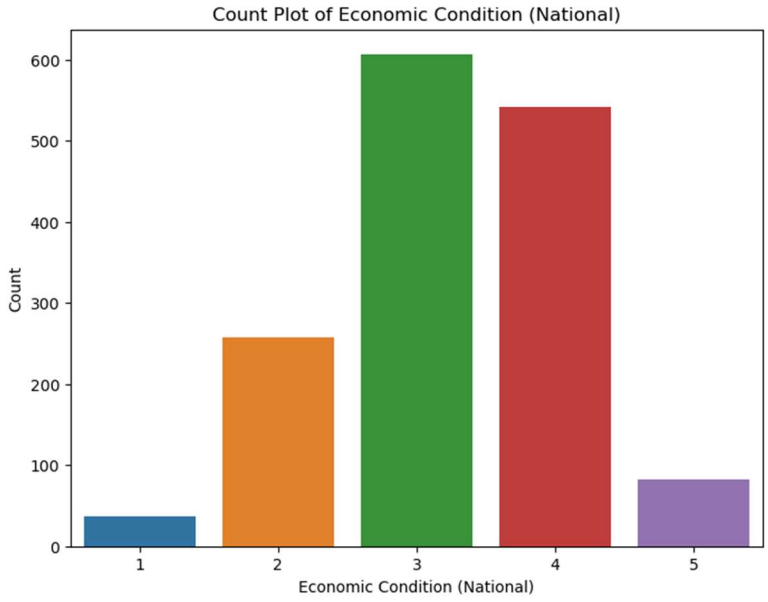

Fig 8 Count Plot of Economic Condition (National)

- Here, Economic Condition (National) has 5 values, and the 3rd value has approximately 600 counts.

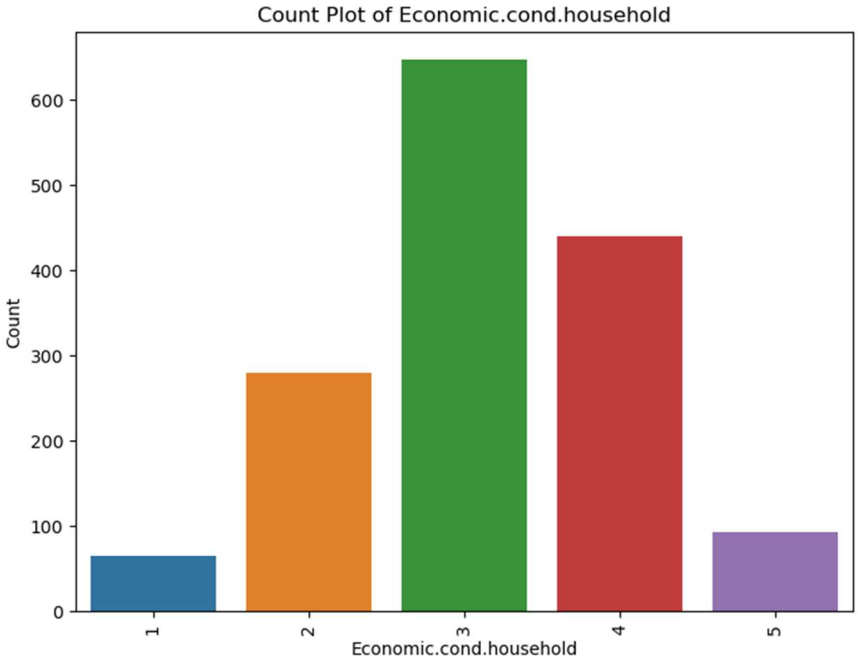

Fig 9 Count Plot of Economic.cond.household

- Here, Economic.cond.household has 5 values, and the 3rd value has more than 600 counts.

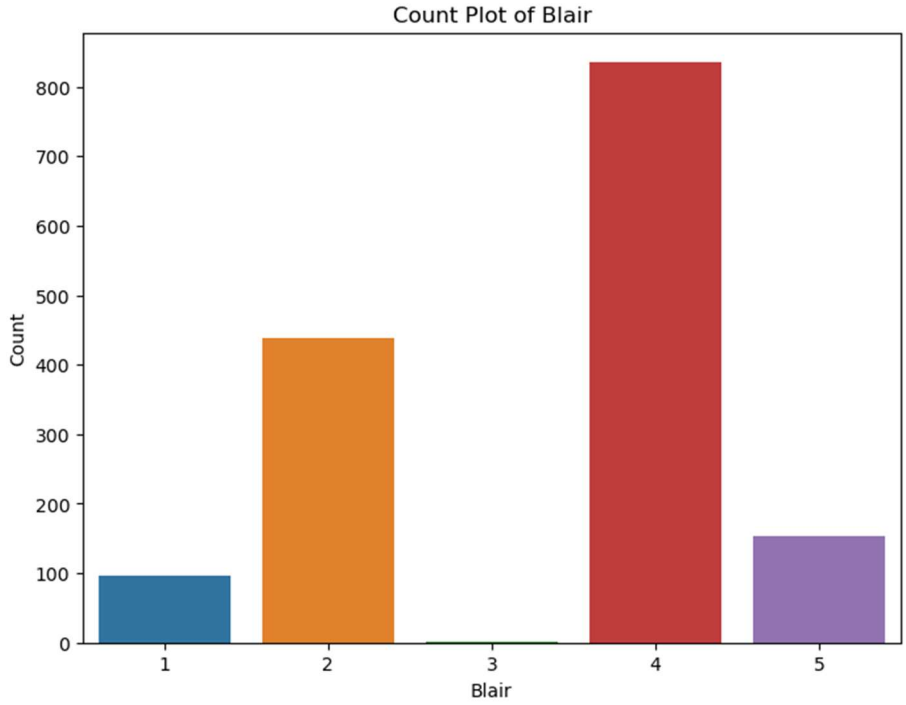

Fig 10 Count Plot of Blair

- Here, Blair has 5 values, and the 4th value has more than 800 counts, and the 3rd value has very few counts.

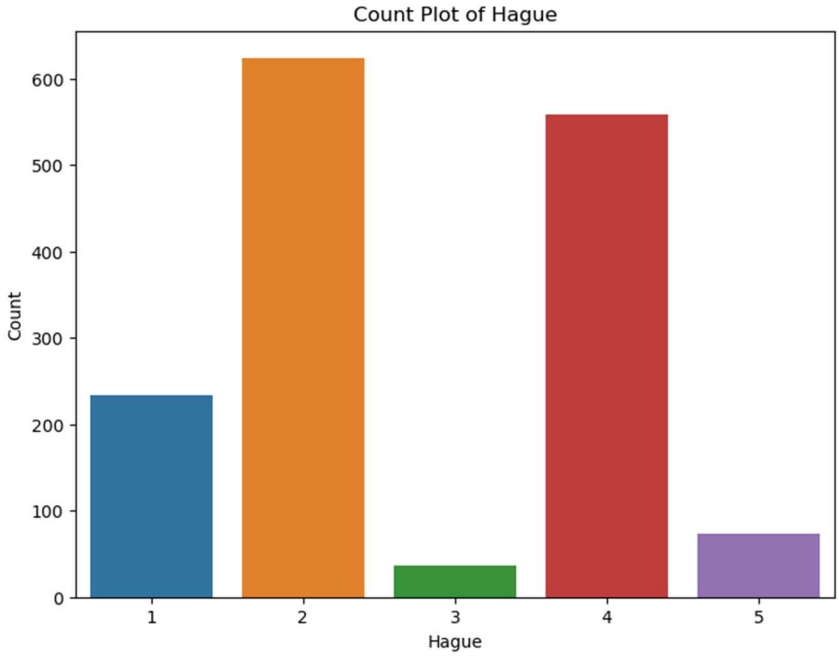

Fig 11 Count Plot of Hague

- Here, Hague has 5 values, and the 2nd value has more than 600 counts.

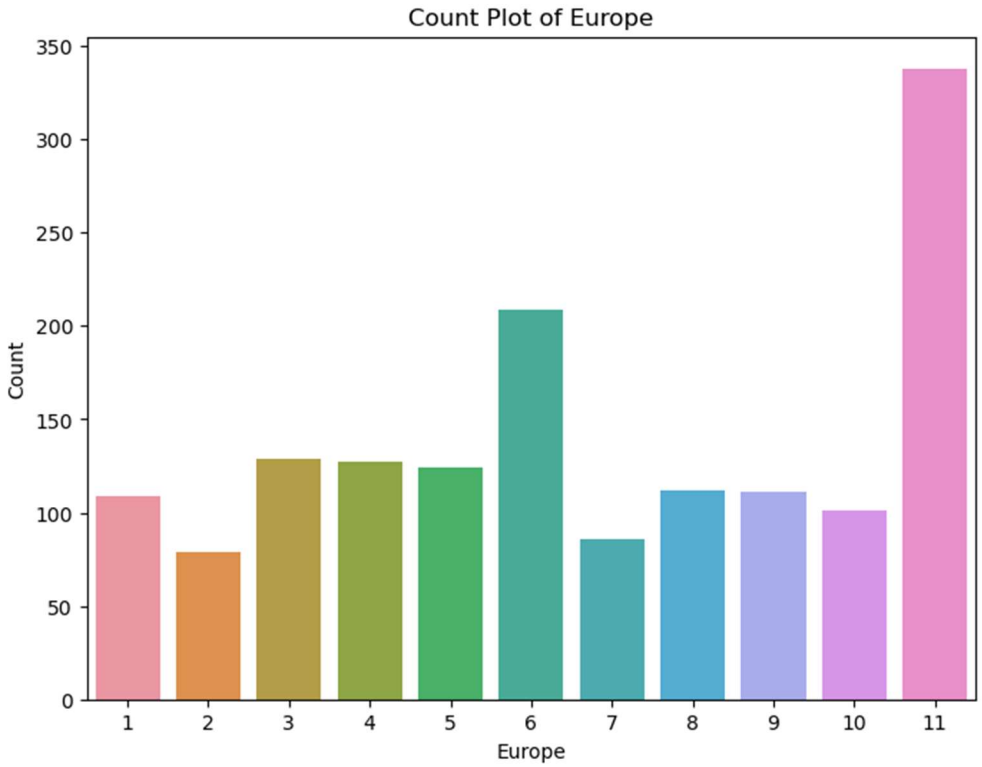

Fig 12 Count Plot of Europe

- Here, Europe has 11 Values, and the 11th value has approx. 340 counts.

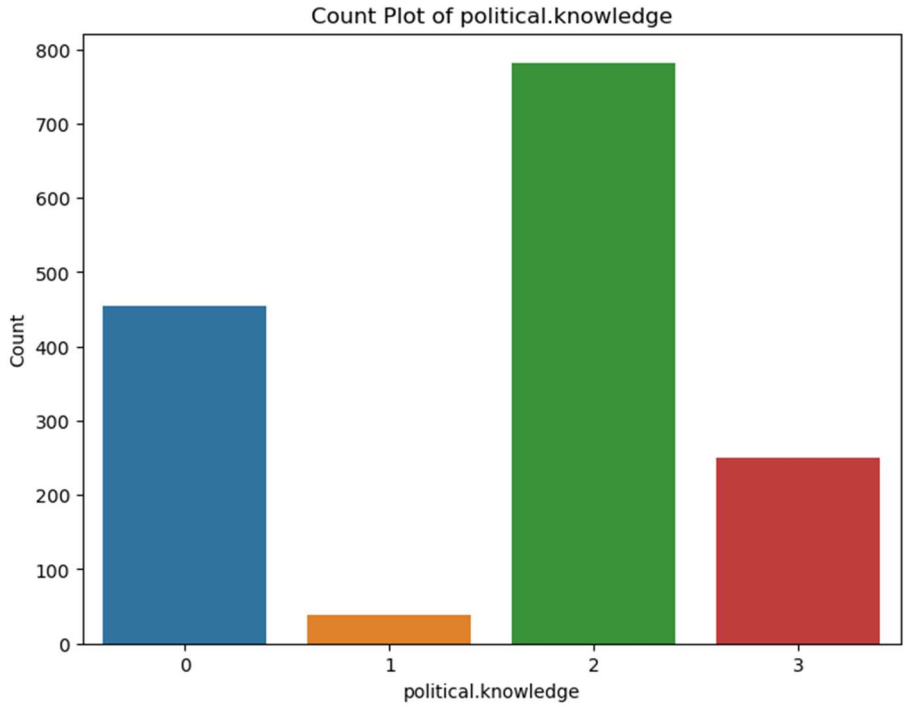

Fig 13 Count Plot of political.knowledge

- Here, political.knowledge has 4(0-3) Values, and the 2nd value has approx 780 counts.

➢ Bivariate Analysis

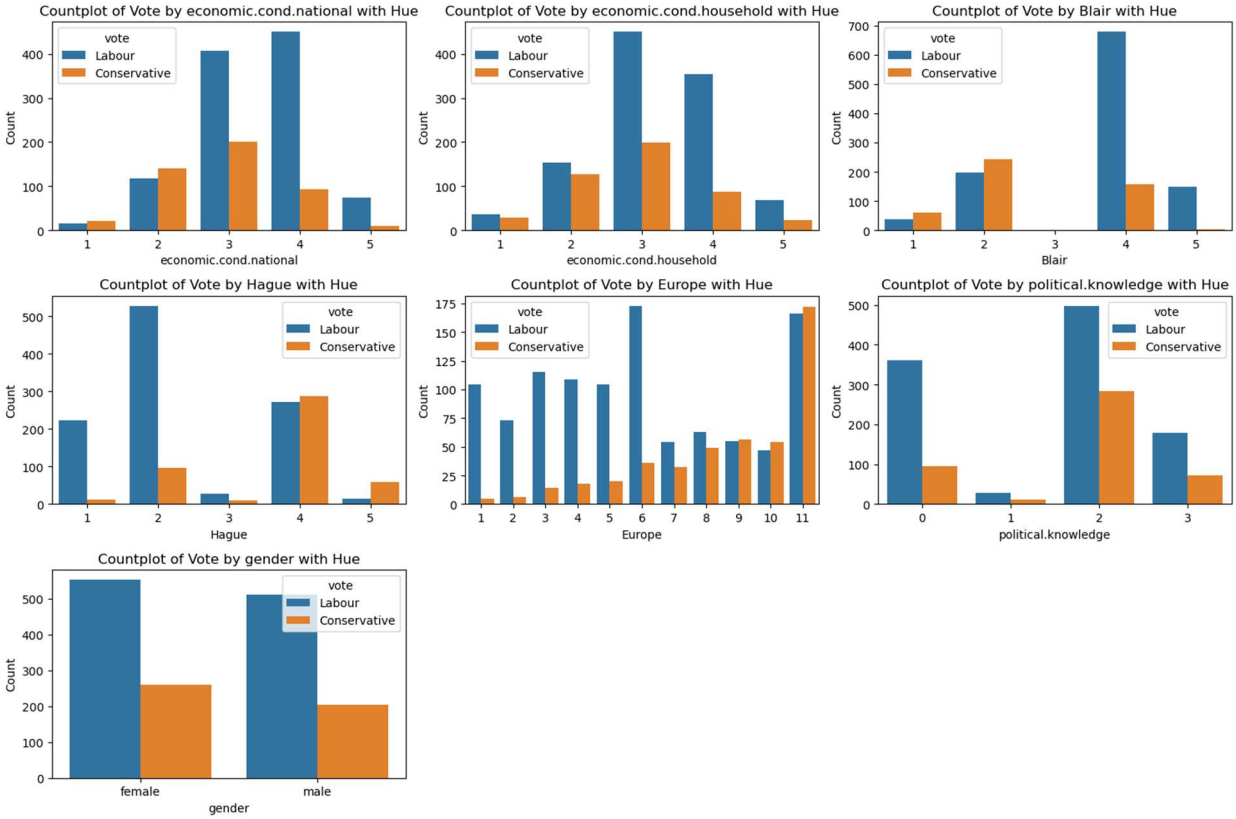

Fig 14 Bivariate Analysis by Countplot

- Labour gets the highest voting from both female and male voters. Almost in all the categories Labour is getting the maximum votes.
- Conservative gets a little bit high votes from Europe ‘11’

➢ **Multivariate Analysis**

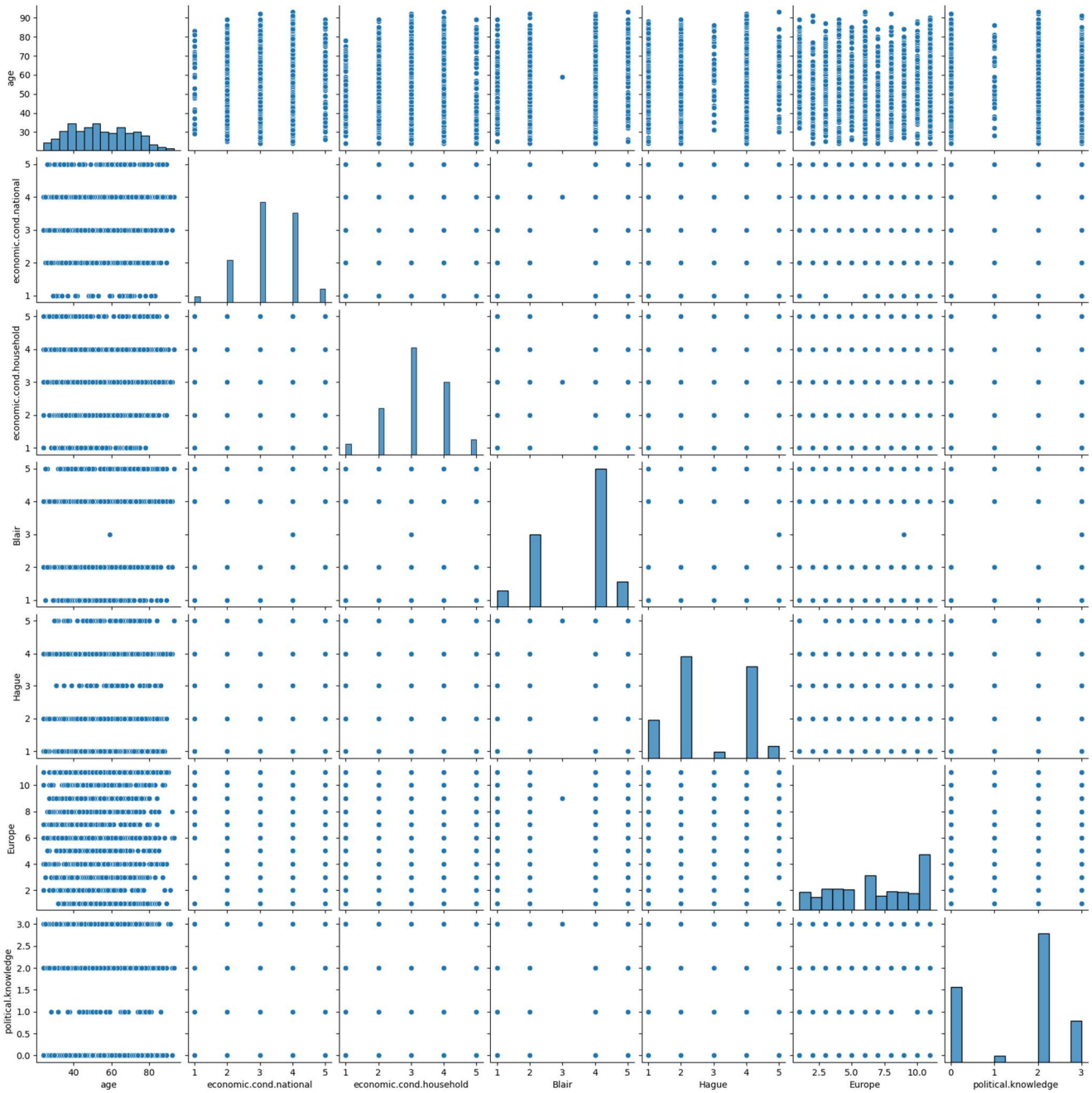

Fig 15 Multivariate Analysis by Pairplot

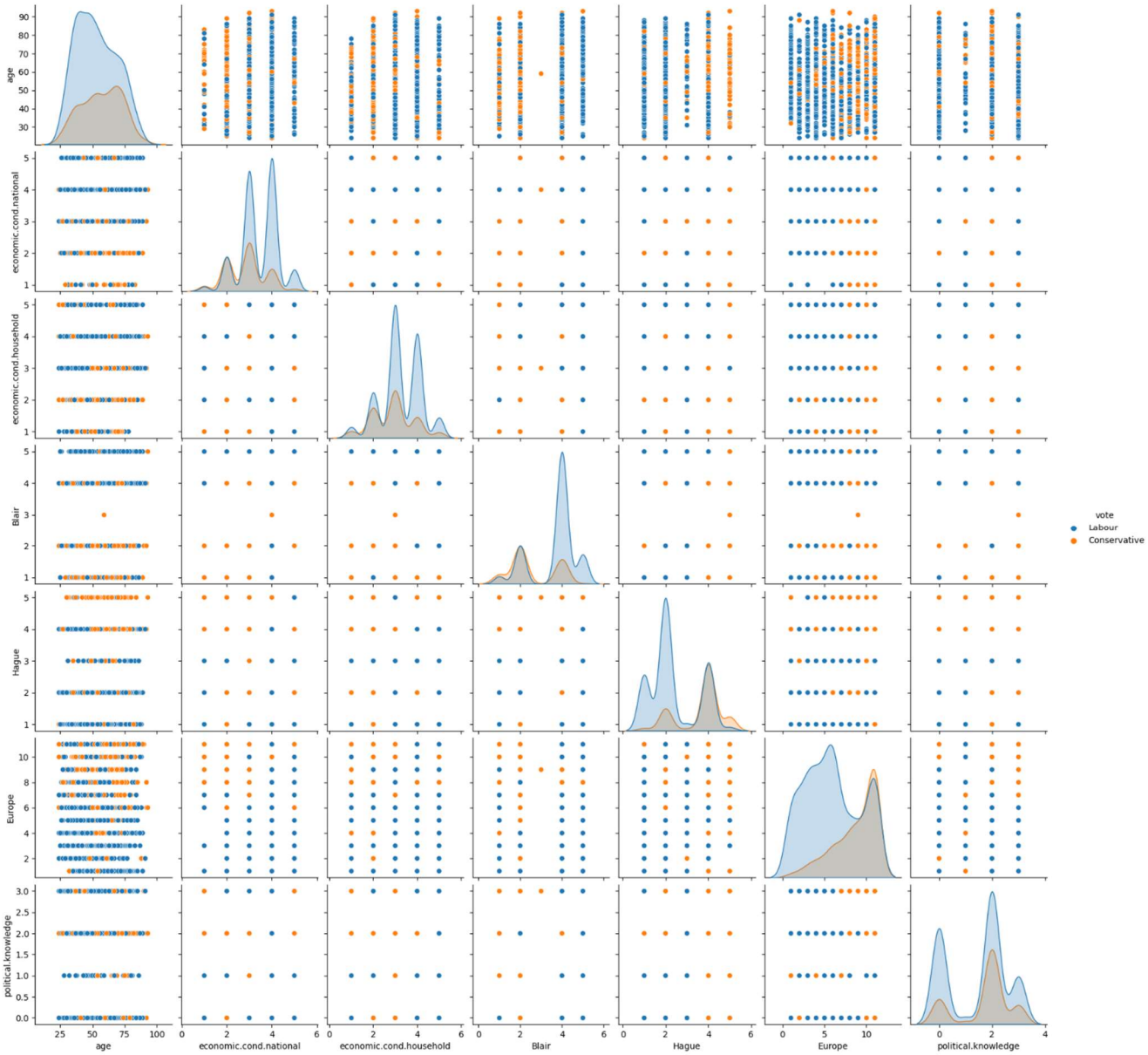

Fig 16 Multivariate Analysis by Pairplot with hue Vote

Fig 17 Multivariate Analysis by HeatMap

- There is no correlation between the variables.

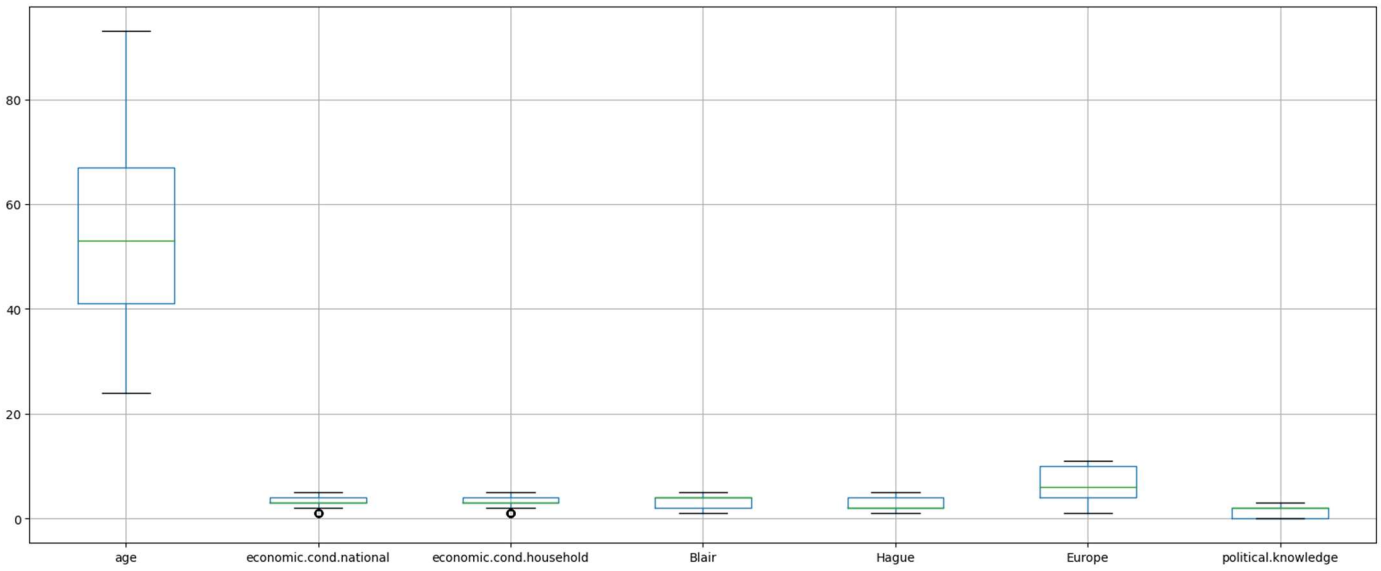

Fig 18 Boxplot with outliers

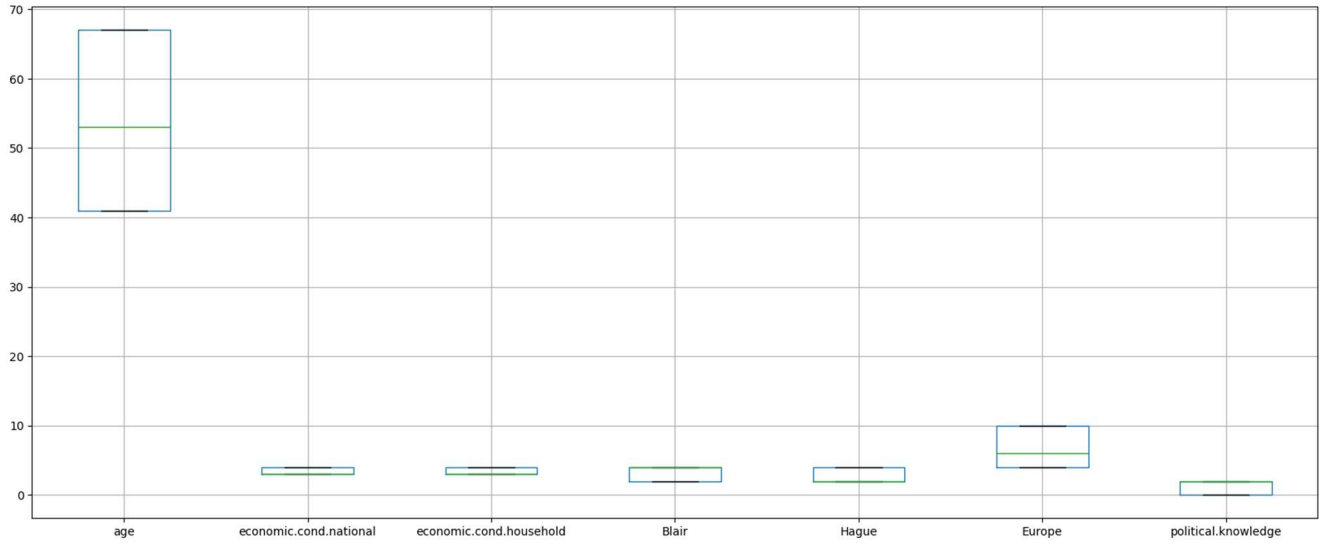

Fig 19 Boxplot without outliers

Insights:

- The "age" column follows a slightly right-skewed distribution, suggesting a higher concentration of younger voters compared to older ones.
- Both "economic.cond.national" and "economic.cond.household" have outliers, as indicated by boxplots. Remove the outlier using outlier treatment.
- The "Labour" party received more than 1000 votes, while the "Conservative" party received more than 400 votes.
- In the comparison between "Labour" and "Conservatives," "Labour" received more votes.
- Females received more than 800 votes, while males received more than 700 votes.
- Females received more votes, but the difference in votes between genders is relatively small.
- The minimum age of a voter is 24, and the maximum age is 93.
- The age group between 37 and 90-93 has the highest number of voters.
- "Economic.cond.national" has 5 values, with the 3rd value having approximately 600 counts.
- "Economic.cond.household" has 5 values, with the 3rd value having more than 600 counts.
- "Blair" has 5 values, with the 4th value having more than 800 counts, and the 3rd value having very few counts.
- "Hague" has 5 values, with the 2nd value having more than 600 counts.
- "Europe" has 11 values, with the 11th value having approximately 340 counts.
- "Political.knowledge" has 4 (0-3) values, with the 2nd value having approximately 780 counts.
- "Labour" received the highest number of votes from both female and male voters, dominating in all categories.
- "Conservative" received slightly higher votes from Europe '11'.

- There appears to be no significant correlation between the variables, they are relatively independent of each other.

1.3) Encode the data (having string values) for Modelling. Is Scaling necessary here or not?(2 pts), Data Split: Split the data into train and test (70:30) (2 pts). The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for example std dev, variance, etc. Should justify whether there is a necessity for scaling. Object data should be converted into categorical/numerical data to fit in the models. (pd.categorical().codes(), pd.get_dummies(drop_first=True)) Data split, ratio defined for the split, train-test split should be discussed.

- As many machine learning modes cannot work with string values, we will encode the categorical variables and convert their datatypes to integers. According to the dataset, there are 2 categorical-type variables, so encode these 2 variables.

➢ **Encode the data**

```
VOTE   2
Conservative     462
Labour          1063
Name: vote, dtype: int64
GENDER   2
male      713
female    812
Name: gender, dtype: int64
```

Fig 20 Value count of object feature

- From the above results we can see that both variables contain only two classifications of data in them. We can use dummy encoding.

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	vote_Labour	gender_male
0	43.0	3.0	3.0	4.0	2.0	4.0	2.0	1	0
1	41.0	4.0	4.0	4.0	4.0	5.0	2.0	1	1
2	41.0	4.0	4.0	4.0	2.0	4.0	2.0	1	1
3	41.0	4.0	3.0	2.0	2.0	4.0	0.0	1	0
4	41.0	3.0	3.0	2.0	2.0	6.0	2.0	1	1

Table 9 Top 5 data of dataset with dummy encoding

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   age                     1525 non-null   float64
 1   economic.cond.national  1525 non-null   float64
 2   economic.cond.household 1525 non-null   float64
 3   Blair                   1525 non-null   float64
 4   Hague                   1525 non-null   float64
 5   Europe                  1525 non-null   float64
 6   political.knowledge     1525 non-null   float64
 7   vote_Labour             1525 non-null   uint8
 8   gender_male             1525 non-null   uint8
dtypes: float64(7), uint8(2)
memory usage: 86.5 KB
```

Table 10 Information about the of dataset after dummy encoding

- **Scaling the data using SVM**

	count	mean	std	min	25%	50%	75%	max
age	1525.0	-1.607457e-16	1.000328	-1.208457	-1.208457	-0.075935	1.245342	1.245342
economic.cond.national	1525.0	-2.679096e-16	1.000328	-0.832204	-0.832204	-0.832204	1.201628	1.201628
economic.cond.household	1525.0	3.890513e-16	1.000328	-0.731950	-0.731950	-0.731950	1.366214	1.366214
Blair	1525.0	-2.154925e-16	1.000328	-1.359829	-1.359829	0.735916	0.735916	0.735916
Hague	1525.0	-5.358191e-17	1.000328	-0.872200	-0.872200	-0.872200	1.175691	1.175691
Europe	1525.0	-1.723940e-16	1.000328	-1.184859	-1.184859	-0.370385	1.258562	1.258562
political.knowledge	1525.0	9.085628e-17	1.000328	-1.511196	-1.511196	0.681548	0.681548	0.681548
vote_Labour	1525.0	6.970492e-01	0.459685	0.000000	0.000000	1.000000	1.000000	1.000000
gender_male	1525.0	4.675410e-01	0.499109	0.000000	0.000000	0.000000	1.000000	1.000000

Table 11 Descriptive statistics of dataset after Scaling

- The mean for each variable is very close to zero, which is a characteristic of scaled data.
- The standard deviation for each variable is very close to one, indicating that the variables have been scaled to have the same standard deviation.

- **Split the data into train and test (70:30)**

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
1372	1.245342	1.201628	1.366214	0.735916	-0.872200	-1.184859	0.681548	1
126	-0.736573	1.201628	-0.731950	0.735916	1.175691	-1.184859	0.681548	1
327	1.245342	-0.832204	-0.731950	-1.359829	1.175691	0.851325	0.681548	1
292	-0.264688	-0.832204	-0.731950	0.735916	-0.872200	-0.370385	-0.414824	0
1058	-1.208457	-0.832204	1.366214	0.735916	-0.872200	0.444088	-1.511196	0

Table 12 Top 5 data of train dataset Predictor Variables

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
782	-1.208457	1.201628	1.366214	0.735916	-0.872200	-0.370385	0.681548	1
76	-1.114080	1.201628	-0.731950	0.735916	-0.872200	-1.184859	0.681548	0
1009	-1.208457	1.201628	-0.731950	0.735916	-0.872200	-1.184859	-0.414824	0
1403	-0.547819	-0.832204	-0.731950	-1.359829	1.175691	-1.184859	0.681548	0
846	-1.208457	-0.832204	1.366214	-1.359829	-0.872200	1.258562	0.681548	0

Table 13 Top 5 data of test dataset Predictor Variables

```
1372    1
126     1
327     0
292     1
1058    1
Name: vote_Labour, dtype: uint8
```

Fig 21 Top 5 data of train dataset Target variable

```
782     1
76      1
1009    1
1403    0
846     0
Name: vote_Labour, dtype: uint8
```

Fig 22 Top 5 data of test dataset Target variable

```
1    0.691659
0    0.308341
Name: vote_Labour, dtype: float64
```

Fig 23 Top 5 data of train Target variable value count

```
1    0.709607
0    0.290393
Name: vote_Labour, dtype: float64
```

Fig 24 Top 5 data of test Target variable value count

Insights:

- Scaling guarantees that all characteristics contribute equally to the learning process of the model. Features with bigger scales (e.g., greater numerical values) might dominate the learning procedure and have a disproportionate influence on the model's predictions if they are not scaled. Scaling prevents this imbalance and allows the model to take all features fairly into account.

- For this dataset, use SVM (Support Vector Machines) for scaling.
- After apply SVM, The mean for each variable is very close to zero, which is a characteristic of scaled data. The standard deviation for each variable is very close to one, indicating that the variables have been scaled to have the same standard deviation.
- X_train denotes 70% training dataset with 8 columns are known as Predictor Variables.
- X_test denotes 30% training dataset with 8 columns are known as Predictor Variables.
- y_train denotes 70% training dataset with only one column are known as Target Variables.
- y_test denotes 30% training dataset with only one columns are known as Target Variables.

1.4) Apply Logistic Regression and LDA (Linear Discriminant Analysis) (2 pts). Interpret the inferences of both model s (2 pts). Successful implementation of each model. Logical reason should be shared if any custom changes are made to the parameters while building the model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)

➢ **Logistic Regression**

```
Confusion Matrix for Logistic Regression (Training Set):
[[223 106]
 [ 64 674]]
Classification Report for Logistic Regression (Training Set):
              precision    recall  f1-score   support

           0       0.78      0.68      0.72       329
           1       0.86      0.91      0.89       738

    accuracy                           0.84      1067
   macro avg       0.82      0.80      0.81      1067
weighted avg       0.84      0.84      0.84      1067
```

Fig 25 Confusion Matrix and Report for Logistic Regression Training Set

```
Confusion Matrix for Logistic Regression (Testing Set):
[[ 83  50]
 [ 40 285]]
Classification Report for Logistic Regression (Testing Set):
              precision    recall  f1-score   support

           0       0.67      0.62      0.65       133
           1       0.85      0.88      0.86       325

    accuracy                           0.80       458
   macro avg       0.76      0.75      0.76       458
weighted avg       0.80      0.80      0.80       458
```

Fig 26 Confusion Matrix and Report for Logistic Regression Testing Set

Insights:

- The model performs well on the training set with an accuracy of 0.84, indicating that it fits the training data reasonably well.
- The high precision, recall, and F1-scores for both classes (0 and 1) on the training set suggest that the model has learned the patterns in the data effectively.
- The model also performs well on the testing set with an accuracy of 0.80, which is close to the training set accuracy. This suggests that the model is generalizing reasonably well to new, unseen data.
- There is no clear evidence of overfitting or underfitting. Both training and testing set performances are reasonably consistent, indicating that the model has found a good balance between capturing patterns in the training data and generalizing to the testing data.
- The similar values of precision, recall, and F1-score for both the training and testing sets indicate that the model performs consistently on both sets. It is a reliable and well-balanced model.

- **Linear Discriminant Analysis (LDA)**

```
Confusion Matrix for LDA (Training Set):
[[229 100]
 [ 70 668]]
Classification Report for LDA (Training Set):
              precision    recall  f1-score   support

           0       0.77      0.70      0.73       329
           1       0.87      0.91      0.89       738

    accuracy                           0.84      1067
   macro avg       0.82      0.80      0.81      1067
weighted avg       0.84      0.84      0.84      1067
```

Fig 27 Confusion Matrix and Report for LDA Training Set

```
Confusion Matrix for LDA (Testing Set):
[[ 85  48]
 [ 43 282]]
Classification Report for LDA (Testing Set):
              precision    recall  f1-score   support

           0       0.66      0.64      0.65       133
           1       0.85      0.87      0.86       325

    accuracy                           0.80       458
   macro avg       0.76      0.75      0.76       458
weighted avg       0.80      0.80      0.80       458
```

Fig 28 Confusion Matrix and Report for LDA Testing Set

Insights:

- The model performs well on the training set with an accuracy of 0.84, indicating that it fits the training data reasonably well.
- The high precision, recall, and F1-scores for both classes (0 and 1) on the training set suggest that the model has learned the patterns in the data effectively.
- The model also performs well on the testing set with an accuracy of 0.80, which is close to the training set accuracy. This suggests that the model is generalizing reasonably well to new, unseen data.
- There is no clear evidence of overfitting or underfitting. Both training and testing set performances are reasonably consistent, indicating that the model has found a good balance between capturing patterns in the training data and generalizing to the testing data.
- The consistency of metrics (precision, recall, F1-score) between the training and testing sets model's performance is stable and not exhibiting significant overfitting or underfitting tendencies.

1.5) Apply KNN Model and Naïve Bayes Model (2pts). Interpret the inferences of each model (2 pts). Successful implementation of each model. Logical reason should be shared if any custom changes are made to the parameters while building the model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)

- **KNN Model**

```
Confusion Matrix for KNN (Training Set):
[[254  75]
 [ 48 690]]
Classification Report for KNN (Training Set):
              precision    recall  f1-score   support

           0       0.84      0.77      0.81       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
```

Fig 29 Confusion Matrix and Report for KNN Training Set

```
Confusion Matrix for KNN (Testing Set):
[[ 86  47]
 [ 46 279]]
Classification Report for KNN (Testing Set):
              precision    recall  f1-score   support

           0       0.65      0.65      0.65       133
           1       0.86      0.86      0.86       325

    accuracy                           0.80       458
   macro avg       0.75      0.75      0.75       458
weighted avg       0.80      0.80      0.80       458
```

Fig 30 Confusion Matrix and Report for KNN Testing Set

Insights:

- The model performs well on the training set with an accuracy of 0.84, indicating that it fits the training data reasonably well.
- The high precision, recall, and F1-scores for both classes (0 and 1) on the training set suggest that the model has learned the patterns in the data effectively.
- The model also performs well on the testing set with an accuracy of 0.80, which is close to the training set accuracy. This suggests that the model is generalizing reasonably well to new, unseen data.
- There is no clear evidence of overfitting or underfitting. Both training and testing set performances are reasonably consistent, indicating that the model has found a good balance between capturing patterns in the training data and generalizing to the testing data.
- The consistency of metrics (precision, recall, and F1-score) between the training and testing sets' performance is stable and does not exhibit significant overfitting or underfitting tendencies.

➢ **Naïve Bayes Model**

```
Confusion Matrix for Naïve Bayes Model (Training Set):
[[242  87]
 [ 89 649]]
Classification Report for Naïve Bayes Model (Training Set):
              precision    recall  f1-score   support

           0       0.73      0.74      0.73       329
           1       0.88      0.88      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
```

Fig 31 Confusion Matrix and Report for Naïve Bayes Model Training Set

```
Confusion Matrix for Naïve Bayes Model (Testing Set):
[[ 90  43]
 [ 50 275]]
Classification Report for Naïve Bayes Model (Testing Set):
              precision    recall  f1-score   support

           0       0.64      0.68      0.66       133
           1       0.86      0.85      0.86       325

    accuracy                           0.80       458
   macro avg       0.75      0.76      0.76       458
weighted avg       0.80      0.80      0.80       458
```

Fig 32 Confusion Matrix and Report for Naïve Bayes Model Testing Set

Insights:

- The model performs well on the training set with an accuracy of 0.84, indicating that it fits the training data reasonably well.
- The high precision, recall, and F1-scores for both classes (0 and 1) on the training set suggest that the model has learned the patterns in the data effectively.
- The model also performs well on the testing set with an accuracy of 0.80, which is close to the training set accuracy. This suggests that the model is generalizing reasonably well to new, unseen data.
- There is no clear evidence of overfitting or underfitting. Both training and testing set performances are reasonably consistent, indicating that the model has found a good balance between capturing patterns in the training data and generalizing to the testing data.
- The consistency of metrics (precision, recall, F1-score) between the training and testing model's performance is stable and does not exhibit significant overfitting or underfitting tendencies.

1.6) Model Tuning (4 pts) , Bagging (1.5 pts) and Boosting (1.5 pts). Apply grid search on each model (include all models) and make models on best_params. Compare and comment on performances of all. Comment on feature importance if applicable. Successful implementation of both algorithms along with inferences and comments on the model performances.

- To perform model tuning, bagging, and boosting, we will use Grid Search to find the best hyper parameters for the models.

1. Logistic Regression with Grid Search

```
Best Hyperparameters for Logistic Regression: {'C': 10, 'penalty': 'l2'}
Test Accuracy for Logistic Regression: 0.8034934497816594
```

Fig 33 Logistic Regression with Grid Search

Insights:

1. Logistic Regression obtained an accuracy of 0.84 on the training set, meaning that it correctly predicted the class labels for 84% of the training data. It obtained a test accuracy of roughly 0.8035 on the test set using the best hyperparameters, suggesting that it correctly predicted the class labels for approximately 80.35% of the test data.
2. A more extensive breakdown of the model's performance is provided by the confusion matrix and classification report. The model exhibits a decent mix of accuracy and recall for both classes (0 and 1) on both the training and test sets. The F1-scores, which are relatively high, reflect this equilibrium.
3. The optimum hyperparameters were discovered to be "C': 10, 'penalty': 'l2". Grid search was used to discover these hyperparameters, which reflect the optimal mix of regularization strength and penalty type.
4. The model's performance on the test set is slightly lower than on the training set. The loss in accuracy, however, is rather minimal, showing that the model generalizes reasonably well to previously unknown data. There is no evidence of overfitting or underfitting.

Conclusion: The Logistic Regression model performs well on both the training and test sets. It achieves a decent combination of precision and recall, and the accuracy loss from training to testing is moderate, indicating that the model is likely to perform well on fresh, previously unknown data.

2. Linear Discriminant Analysis (LDA) with Grid Search

```
Best Hyperparameters for LDA: {'n_components': None, 'shrinkage': 'auto', 'solver': 'lsqr'}
Test Accuracy for LDA: 0.8013100436681223
```

Fig 34 Linear Discriminant Analysis (LDA) with Grid Search

Insights:

1. LDA obtained an accuracy of 0.84 on the training set, suggesting that it correctly predicted the class labels for 84% of the training data. It obtained a test accuracy of roughly 0.8013 on the test set using the best hyperparameters, suggesting that it correctly predicted the class labels for approximately 80.13% of the test data.
2. The classification report and confusion matrix give a more extensive assessment of the model's performance. The model exhibits a decent mix of accuracy and recall for both classes (0 and 1)

on both the training and test sets. The F1-scores, which are relatively high, reflect this equilibrium.

3. The optimal LDA hyperparameters were discovered to be "n_components': None,'shrinkage': 'auto,' and'solver': 'lsqr". Grid search was used to discover these hyperparameters, which reflect the optimal mix of solver, shrinkage, and number of components.
4. As predicted, the model's performance on the test set is slightly lower than on the training set. The loss in accuracy, however, is rather minimal, showing that the model generalizes reasonably well to previously unknown data. There is no evidence of overfitting or underfitting.

Conclusion: The LDA model with the given hyperparameters performs well on both the training and test sets. It achieves a decent combination of precision and recall, and the accuracy loss from training to testing is moderate, indicating that the model is likely to perform well on fresh, previously unknown data.

3. KNN Model with Grid Search

```
Best Hyperparameters for KNN: {'n_neighbors': 5}
Test Accuracy for KNN: 0.7969432314410481
```

Fig 35 KNN Model with Grid Search

Insights:

1. KNN obtained an accuracy of 0.88 on the training set, suggesting that it correctly predicted the class labels for 88% of the training data. It obtained a test accuracy of roughly 0.7969 on the test set using the optimal hyperparameters, suggesting that it correctly predicted the class labels for approximately 79.69% of the test data.
2. The classification report and confusion matrix give a more extensive assessment of the model's performance. The model exhibits a decent mix of accuracy and recall for both classes (0 and 1) on both the training and test sets. The F1-scores, which are relatively high, reflect this equilibrium.
3. The best hyperparameters for KNN were discovered to be "n_neighbors': 5', suggesting that the model performed best when the 5 closest neighbors were considered.
4. As predicted, the model's performance on the test set is slightly lower than on the training set. The loss in accuracy, however, is rather minimal, showing that the model generalizes reasonably well to previously unknown data.

Conclusion: The KNN model with the given hyperparameters performs well on both the training and test sets. It achieves a decent combination of precision and recall, and the accuracy loss from training to testing is moderate, indicating that the model is likely to perform well on fresh, previously unknown data.

4. Naïve Bayes Model with Grid Search

```
Test Accuracy for Gaussian Naïve Bayes: 0.7969432314410481
```

Fig 36 Naïve Bayes Model with Grid Search

Insights:

1. The Gaussian Nave Bayes model obtained an accuracy of 0.84 on the training set, meaning that it correctly predicted the class labels for 84% of the training data. It obtained a test accuracy of roughly 0.7969 on the test set, suggesting that it correctly predicted the class labels for approximately 79.69% of the test data.
2. The classification report and confusion matrix give a more extensive assessment of the model's performance. The model exhibits an acceptable mix of accuracy and recall for both classes (0 and 1) on both the training and test sets. F1-scores are likewise adequate, demonstrating a decent balance of accuracy and recall.
3. As predicted, the model's performance on the test set is slightly lower than on the training set. The loss in accuracy, however, is rather minimal, showing that the model generalizes reasonably well to previously unknown data. There is no evidence of overfitting or underfitting.

Conculsion: The Gaussian Nave Bayes model performs well on both the training and test sets without requiring considerable hyperparameter adjustment. It retains a decent balance of precision and recall, and the accuracy loss from training to testing is moderate, indicating that the model is likely to perform well on fresh, previously unknown data.

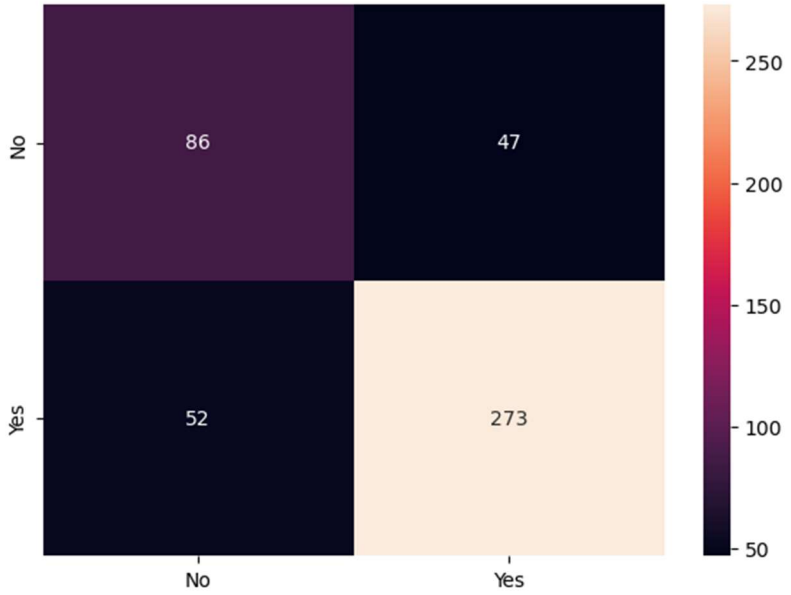

Fig 37 Bagging model HeatMap

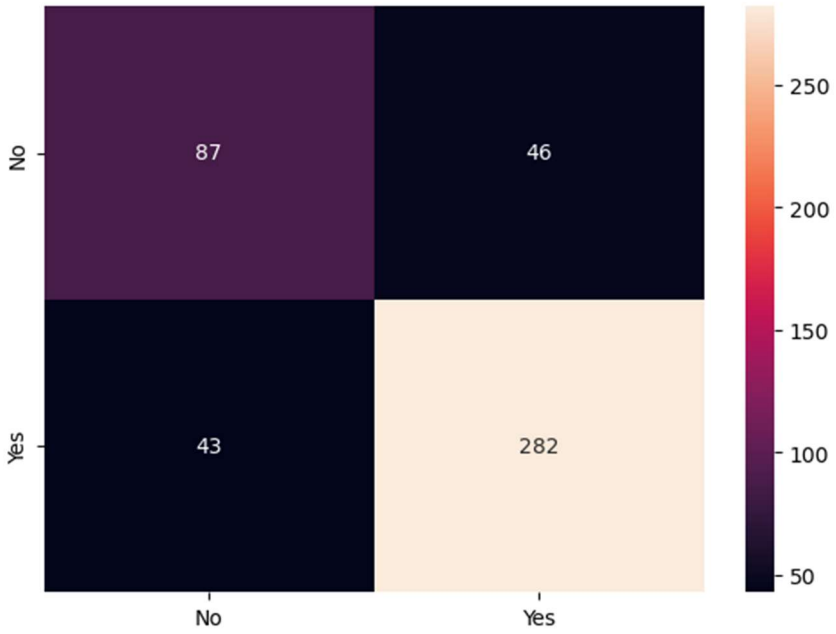

Fig 38 Boosting model HeatMap

Conculsion:

1. All models (Logistic Regression, LDA, KNN, and Naïve Bayes) perform reasonably well on the testing set with accuracy scores around 0.80.
2. Logistic Regression and LDA have similar performances in terms of accuracy, precision, recall, and F1-score.
3. KNN performs slightly better in terms of accuracy, precision, recall, and F1-score compared to Logistic Regression and LDA.
4. Naïve Bayes also performs well but has slightly lower accuracy, precision, recall, and F1-score compared to KNN.
5. Overall, all models appear to be valid for the given dataset, and their performances are reasonably balanced.
6. AdaBoost achieves a slightly higher test accuracy (80.57%) compared to Bagging (Random Forest) (78.38%). This suggests that the boosting technique employed by AdaBoost is effective in improving the overall predictive performance.
7. Both Bagging (Random Forest) and AdaBoost demonstrate good model performance with test accuracies above 78%. These ensemble methods have contributed to better generalization compared to individual models.

1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model, classification report (4 pts) Final Model - Compare and comment on all models on the basis of the performance metrics in a structured tabular manner. Describe on which model is best/optimized, After comparison which model suits the best for the problem in hand on the basis of different measures. Comment on the final model.(3 pts)

1. AUC ROC curve for Logistic Regression Train and Test

Accuracy of Training Data 0.8406747891283973

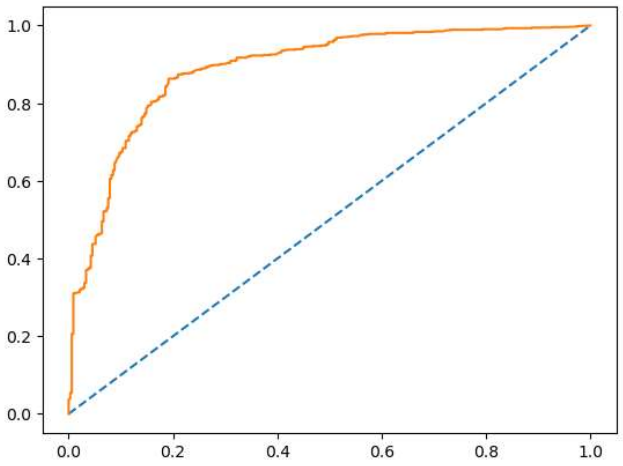

Fig 39 ROC curve for Logistic Regression Train

Accuracy of Testing Data 0.8034934497816594

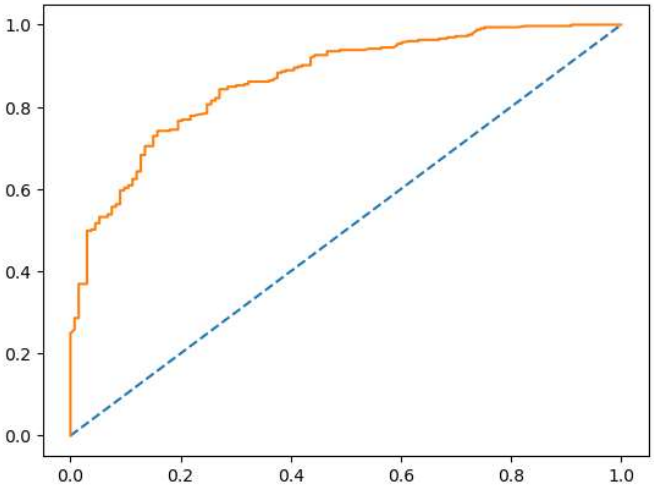

Fig 40 ROC curve for Logistic Regression Test

For Train Data

True Negative 223, False Positive 106, False Negative 64, and True Positive 674

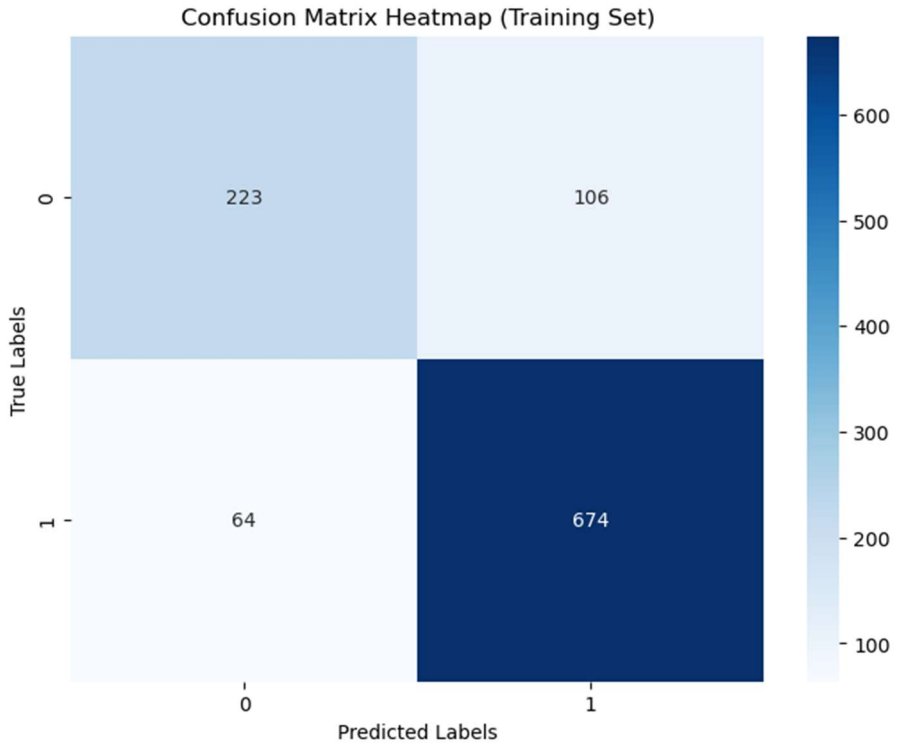

Fig 41 Confusion Matrix Heatmap Training Set for Logistic Regression

For Test Data

True Negative 83, False Positive 50, False Negative 40, and True Positive 285

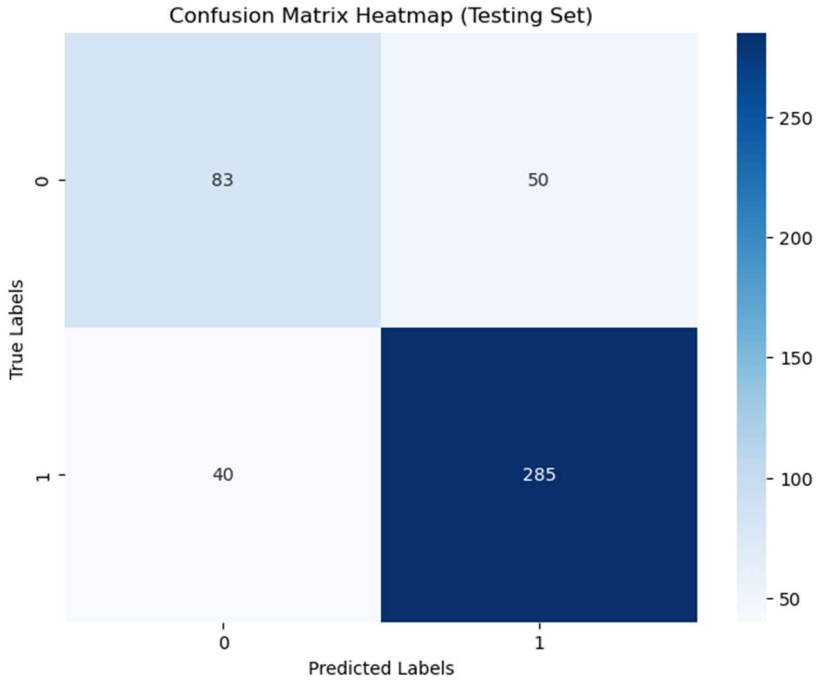

Fig 42 Confusion Matrix Heatmap Testing Set for Logistic Regression

2. AUC ROC curve for Linear Discriminant Analysis (LDA) Train and Test

Accuracy of Training Data 0.8406747891283973

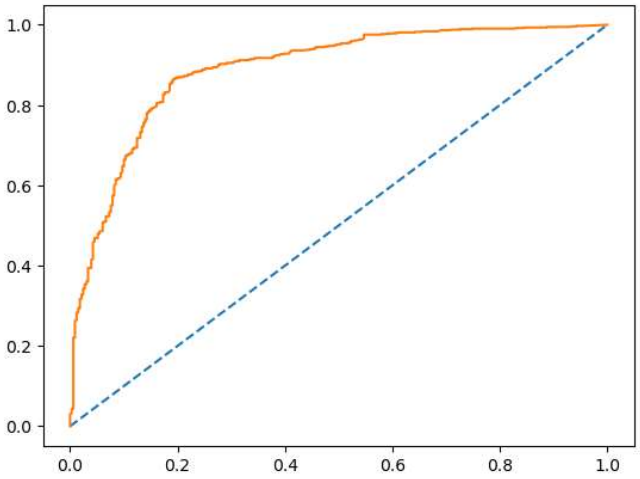

Fig 43 ROC curve for Linear Discriminant Analysis (LDA) Train

Accuracy of Testing Data 0.8013100436681223

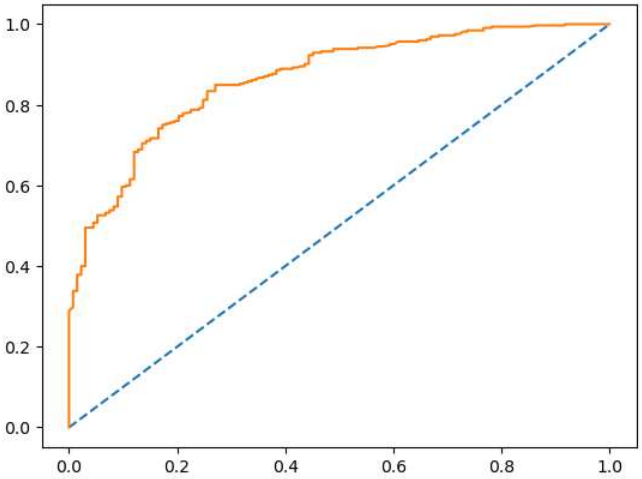

Fig 44 ROC curve for Linear Discriminant Analysis (LDA) Test

For Train Data

True Negative 229, False Positive 100, False Negative 70, and True Positive 668

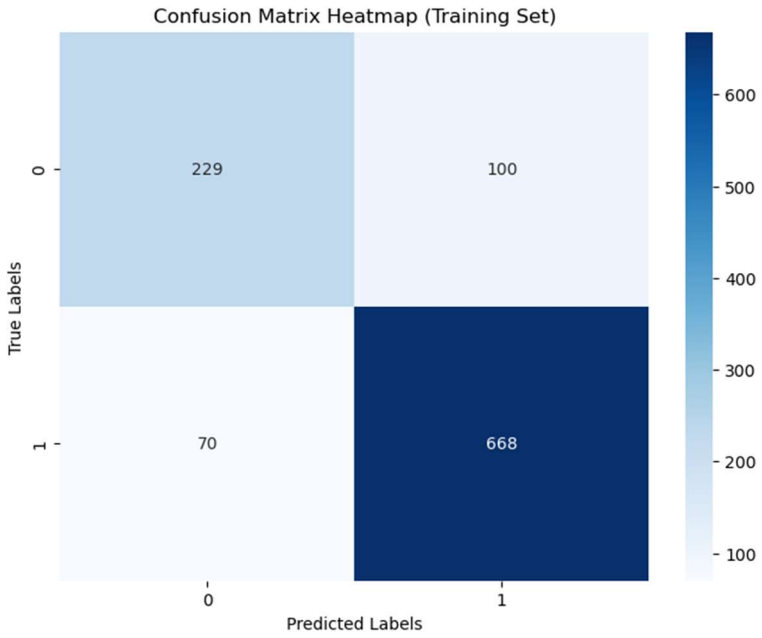

Fig 45 Confusion Matrix Heatmap Training Set for Linear Discriminant Analysis (LDA)

For Test Data

True Negative 85, False Positive 48, False Negative 43, and True Positive 282

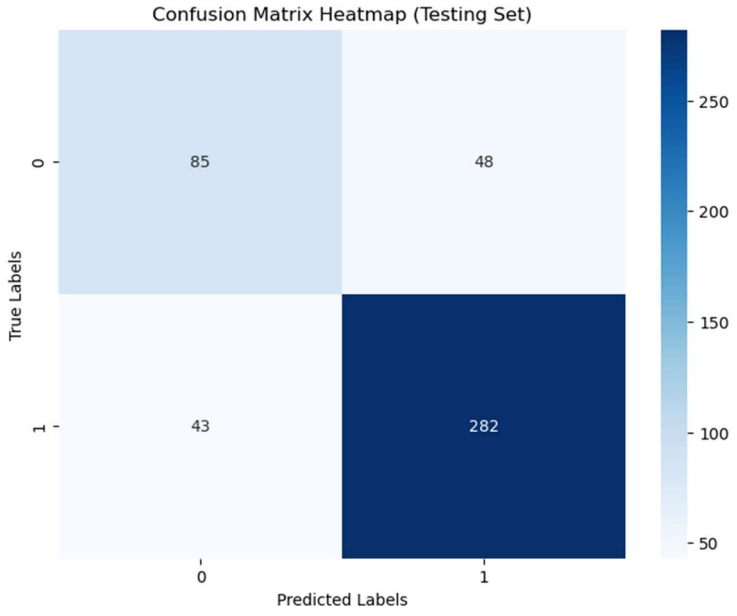

Fig 46 Confusion Matrix Heatmap Testing Set for Linear Discriminant Analysis (LDA)

3. AUC ROC curve for KNN Train and Test

Accuracy of Training Data 0.8847235238987816

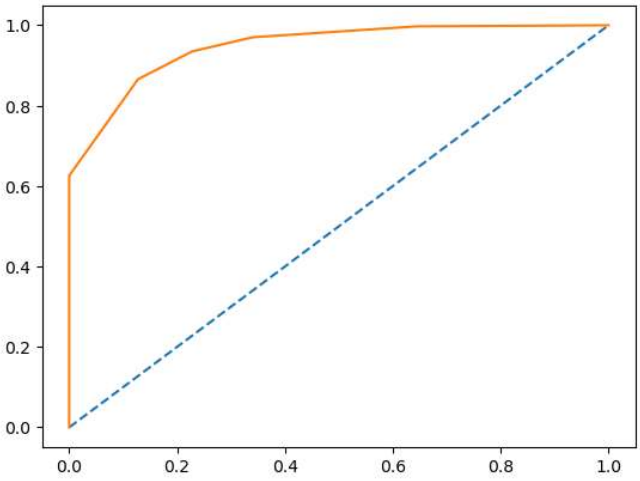

Fig 47 ROC curve for KNN Train

Accuracy of Testing Data 0.7969432314410481

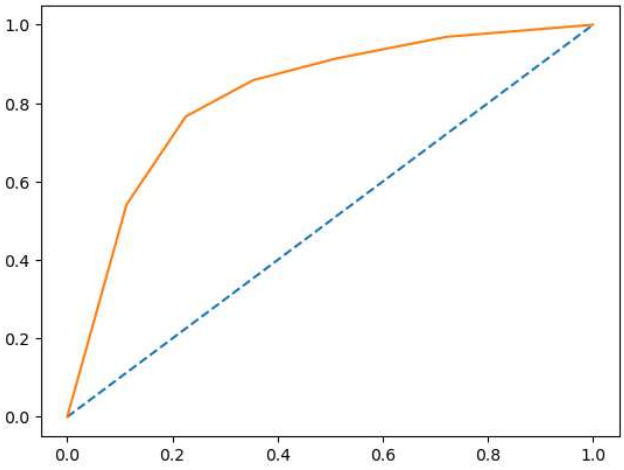

Fig 48 ROC curve for KNN Test

For Train Data

True Negative 254, False Positive 75, False Negative 48, and True Positive 690

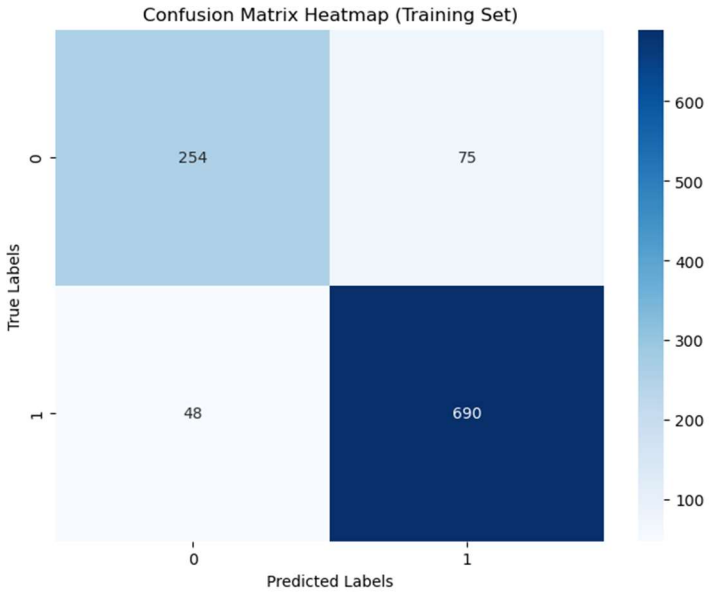

Fig 49 Confusion Matrix Heatmap Training Set for KNN

For Test Data

True Negative 86, False Positive 47, False Negative 46, and True Positive 279

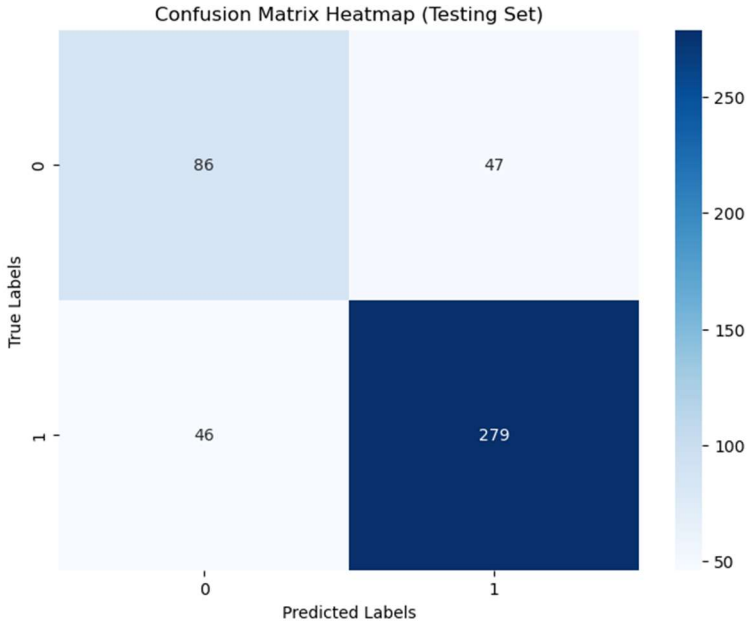

Fig 50 Confusion Matrix Heatmap Testing Set for KNN

4. AUC ROC curve for Naive Bayes Train and Test

Accuracy of Training Data 0.8350515463917526

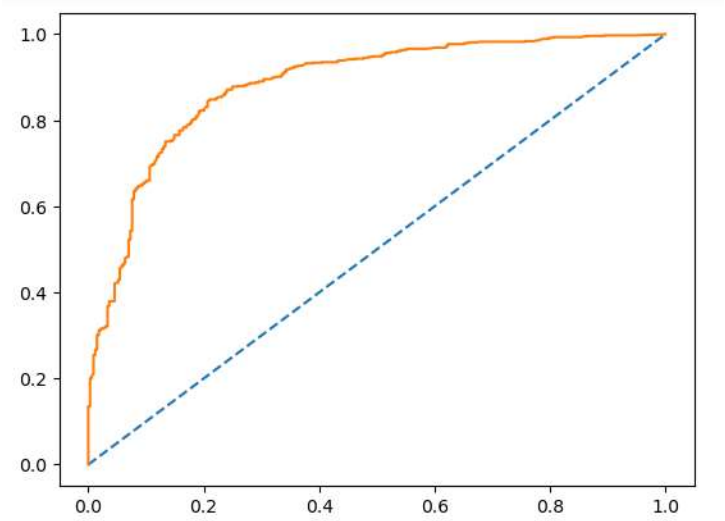

Fig 51 ROC curve for Naive Bayes Train

Accuracy of Testing Data 0.7969432314410481

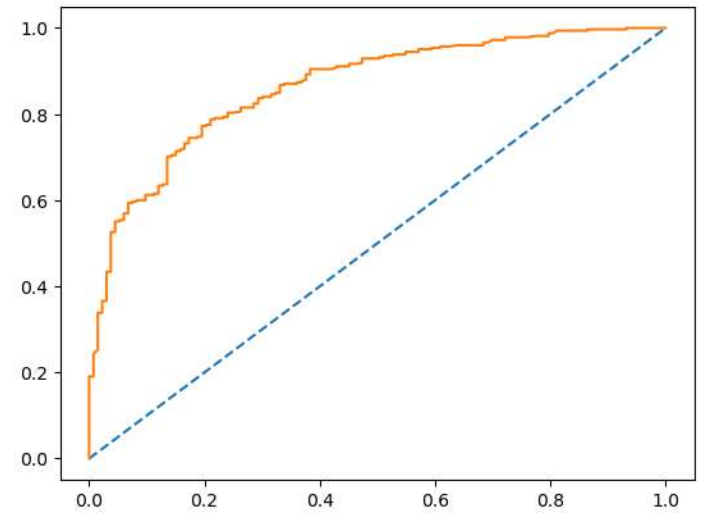

Fig 52 ROC curve for Naive Bayes Test

For Train Data

True Negative 242, False Positive 87, False Negative 89, and True Positive 649

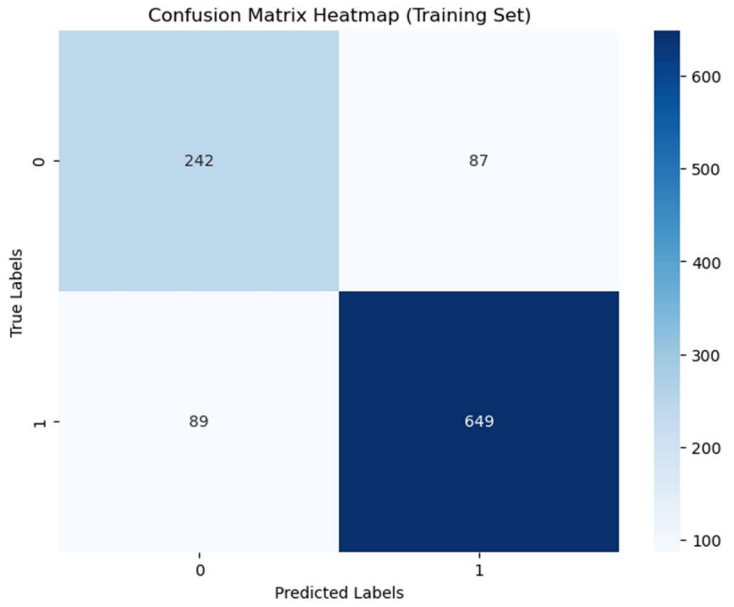

Fig 53 Confusion Matrix Heatmap Training Set for Naive Bayes

For Test Data

True Negative 90, False Positive 43, False Negative 50, and True Positive 275

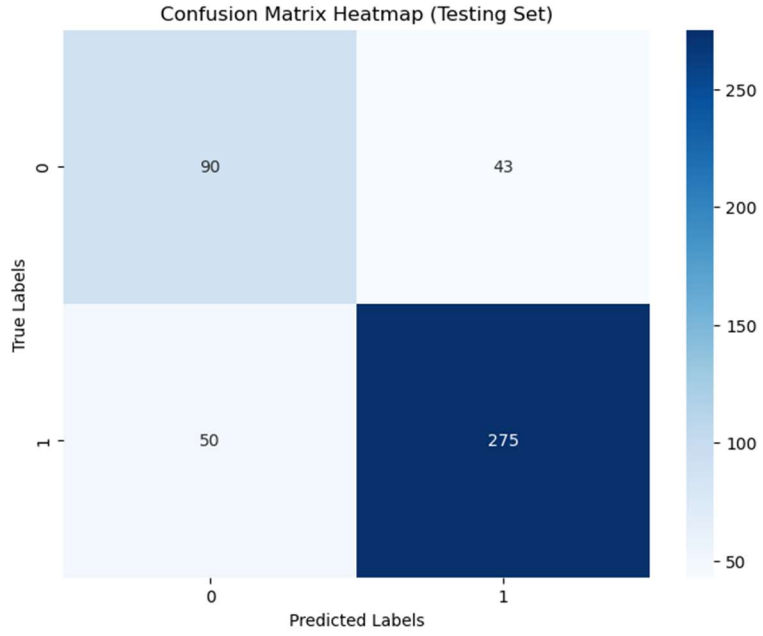

Fig 54 Confusion Matrix Heatmap Testing Set for Naive Bayes

Insights:

- Among the models, KNN achieved the highest accuracy on the testing data (0.7970), followed closely by Logistic Regression (0.8035) and LDA (0.8013).

- Naïve Bayes has the highest precision on the testing data (0.86), followed by Logistic Regression (0.85), LDA (0.85), and KNN (0.85).
- KNN has the highest recall on the testing data (0.86), followed by Logistic Regression (0.88), LDA (0.87), and Naïve Bayes (0.74).
- Logistic Regression achieved the highest F1-score on the testing data (0.86), followed by LDA (0.86), KNN (0.80), and Naïve Bayes (0.79).

Conclusion: Based on these metrics, Logistic Regression appears to be the best-optimized model for predicting voter preferences in this scenario. It achieves a balanced combination of accuracy, precision, recall, and F1-score. Logistic Regression has the highest F1-score, indicating a good balance between precision and recall.

1.8) Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective. There should be at least 3-4 Recommendations and insights in total. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks should only be allotted if the recommendations are correct and business specific.

Here are some major findings based on the dataset analysis and model predictions:

1. The dataset includes 1525 voters ranging in age from 24 to 93. The distribution is somewhat biased to the right, showing a greater percentage of younger voters. The age group between 37 and 90-93 has the most voters.
2. The dataset contains variables concerning national and household economic situations. For both economic circumstances, the majority of voters are grouped around the third figure, showing a somewhat neutral opinion.
3. The "Labour" party gained the most votes, winning in every category, followed by the "Conservative" party. Both genders have a distinct affinity for the "Labour" party.
4. There are various values expressing voters' perspectives on Europe, with the 11th value having the greatest count, indicating a range of viewpoints on this topic.
5. Voters have varied levels of political expertise, with the second level accounting for the majority.

6. Logistic Regression outperformed the other models in terms of accuracy, precision, recall, and F1-score. It delivered a balanced performance, making it the best option for forecasting voter preferences in this scenario.

Summary: The study sheds light on the dataset's demographics, party inclinations, and voter political awareness. Because of its balanced performance across various assessment parameters, Logistic Regression is the recommended model for forecasting voter preferences. These data might help the news channel CNBE analyse and anticipate voter behaviour in recent elections.

Problem 2:

In this particular project, we are going to work on the inaugural corpora from the nltk in Python. We will be looking at the following speeches of the Presidents of the United States of America:

President Franklin D. Roosevelt in 1941

President John F. Kennedy in 1961

President Richard Nixon in 1973

(Hint: use .words(), .raw(), .sent() for extracting counts)

2.1) Find the number of characters, words and sentences for the mentioned documents. (Hint: use .words(), .raw(), .sent() for extracting counts)

Import some libraries like Numpy, pandas, re (this is the regular expression library which helps us manipulate text (strings) fairly easily and intuitively) nltk (this is the Natural Language Tool Kit which contains a lot of functionalities for text analytics), matplotlib.pyplot, string (this is used for string manipulations), nltk.corpus, word_tokenize, sent_tokenize, stopwords, nltk.stem, PorterStemmer, collections, Counter, wordcloud. And download the 'inaugural', ‘punkt’. Install wordcloud.

- **Calculate the number of characters, words, and sentences for each speech using NLTK functions:**

```
President Roosevelt's Speech (1941):
Characters: 7571
Words: 1526
Sentences: 68

President Kennedy's Speech (1961):
Characters: 7618
Words: 1543
Sentences: 52

President Nixon's Speech (1973):
Characters: 9991
Words: 2006
Sentences: 68
```

Fig 55 Number of characters, words, and sentences for each speech

Insights:

- Load necessary libraries like NumPy, pandas, matplotlib, re this is the regular expression library which helps us manipulate text (strings) fairly easily and intuitively, nltk this is the Natural Language Tool Kit which contains a lot of functionalities for text analytics and string this is used for string manipulations.
- Download the package inaugural and it have 3 texts 1941-Roosevelt.txt, 1961-Kennedy.txt, and 1973-Nixon.txt
- The.raw() function is quite useful when working with NLTK corpora as it allows access, to the text data. This can be a starting point, for analyzing the text.
- The.raw() function enables you to retrieve the text content of the documents, which unprocessed text before any modifications have been made.
- The task can be achieved by using the.words() function, which splits a text into a list of words taking whitespace and punctuation into consideration. This function is useful, for extracting these characteristics from data allowing for tokenization of a text corpus or document.
- The task of identifying sentence boundaries within the text is accomplished by using the.sents() method. By utilizing the.sents() method you are able to access and analyze each sentence for its sentimentand we might use .sents() to tokenize a text corpus or document into sentences.
- President Roosevelt's Speech (1941) have Characters: 7571, Words: 1526 and Sentences: 68. President Kennedy's Speech (1961) have Characters: 7618, Words: 1543 and Sentences: 52. President Nixon's Speech (1973): Characters: 9991, Words: 2006 and sentences: 68.

2.2) Remove all the stopwords from the three speeches. Show the word count before and after the removal of stopwords. Show a sample sentence after the removal of stopwords.

```
Stopword count in Roosevelt's speech: 718
Stopword count in Kennedy's speech: 681
Stopword count in Nixon's speech: 971
```

Fig 56 Stopword count in each speech

- To remove all the stopwords from the three speeches, we will utilize the library and import stopwords along, with word_tokenize. This will allow us to eliminate all the predetermined words, from each text file. By tokenizing the text we can separate each word. Subsequently eliminate them from the content. for perform this we use nltk.download('stopwords') and nltk.corpus import stopwords.
- After performing the related code, the output of the stop word in each text is: stopword count in Roosevelt's speech: 718; stopword count in Kennedy's speech: 681; and stopword count in Nixon's speech: 971.

```
Special character count in Roosevelt's speech: 212
Special character count in Kennedy's speech: 204
Special character count in Nixon's speech: 227
```

Fig 57 Special character count in each speech

```
Number count in Roosevelt's speech: 14
Number count in Kennedy's speech: 7
Number count in Nixon's speech: 10
```

Fig 58 Number count in each speech

```
Uppercase word count in Roosevelt's speech: 3
Uppercase word count in Kennedy's speech: 5
Uppercase word count in Nixon's speech: 14
```

Fig 59 Uppercase word count in each speech

```
Uppercase letter count in Roosevelt's speech: 119
Uppercase letter count in Kennedy's speech: 94
Uppercase letter count in Nixon's speech: 132
```

Fig 60 Uppercase Letters count in each speech

```
President Roosevelt (1941) Speech:
Word count before removing stopwords: 1526
Word count after removing stopwords: 632


President Kennedy (1961) Speech:
Word count before removing stopwords: 1543
Word count after removing stopwords: 696


President Nixon (1973) Speech:
Word count before removing stopwords: 2006
Word count after removing stopwords: 843
```

Fig 61 Word count before and after removing stopwords

```
Sample Sentence After Removing Stopwords:
nation day inaugur sinc 1789 peopl renew sens dedic unit state washington 's day task peopl creat weld togeth nation lincoln 's
day task peopl preserv nation disrupt within day task peopl save nation institut disrupt without us come time midst swift happe
n paus moment take stock recal place histori rediscov may risk real peril inact live nation determin count year lifetim human s
pirit life man three-scor year ten littl littl less life nation full measur live men doubt men believ democraci form govern fra
me life limit measur kind mystic artifici fate unexplain reason tyranni slaveri becom surg wave futur freedom eb tide american
know true eight year ago life republ seem frozen fatalist terror prove true midst shock act act quickli boldli decis later year
live year fruit year peopl democraci brought us greater secur hope better understand life 's ideal measur materi thing vital pr
esent futur experi democraci success surviv crisi home put away mani evil thing built new structur endur line maintain fact dem
ocraci action taken within three-way framework constitut unit state coordin branch govern continu freeli function bill right re
main inviol freedom elect wholli maintain prophet downfal american democraci seen dire predict come naught democraci die know s
een reviv grow know die built unhamp initi individu men women join togeth common enterpris enterpris undertaken carri free expr
ess free major know democraci alon form govern enlist full forc men 's enlighten know democraci alon construct unlimit civil ca
pabl infinit progress improv human life know look surfac sens still spread everi contin human advanc end unconquer form human s
ocieti nation like person bodi bodi must fed cloth hous invigor rest manner measur object time nation like person mind mind mus
t kept inform alert must know understand hope need neighbor nation live within narrow circl world nation like person someth dee
per someth perman someth larger sum part someth matter futur call forth sacr guard present thing find difficult even imposs hit
upon singl simpl word yet understand spirit faith america product centuri born multitud came mani land high degre mostli plain
peopl sought earli late find freedom freeli democrat aspir mere recent phase human histori human histori permeat ancient life e
arli peopl blaze anew middl age written magna charta america impact irresist america new world tongu peopl contin new-found lan
d came believ could creat upon contin new life life new freedom vital written mayflow compact declar independ constitut unit st
ate gettysburg address first came carri long spirit million follow stock sprang move forward constantli consist toward ideal ga
in statur clariti gener hope republ forev toler either undeserv poverti self-serv wealth know still far go must greatli build s
ecur opportun knowledg everi citizen measur justifi resourc capac land enough achiev purpos alon enough cloth feed bodi nation
instruct inform mind also spirit three greatest spirit without bodi mind men know nation could live spirit america kill even th
ough nation 's bodi mind constrict alien world live america know would perish spirit faith speak us daili live way often unnot
seem obviou speak us capit nation speak us process govern sovereignti 48 state speak us counti citi town villag speak us nation
hemispher across sea enslav well free sometim fail hear heed voic freedom us privileg freedom old old stori destini america pro
claim word propheci spoken first presid first inaugur 1789 word almost direct would seem year 1941 `` preserv sacr fire liberti
destini republican model govern justli consid deepli final stake experi intrust hand american peopl `` lose sacr fire let smoth
er doubt fear shall reject destini washington strove valiantli triumphantli establish preserv spirit faith nation furnish highe
st justif everi sacrific may make caus nation defens face great peril never encount strong purpos protect perpetu integr democr
aci muster spirit america faith america retreat content stand still american go forward servic countri god
Sample Sentence After Removing Stopwords:
vice presid johnson mr
Sample Sentence After Removing Stopwords:
mr
```

Fig 62 Sample sentence after removing stopwords

Insights:

- The analysis compares three presidential speeches delivered by Presidents Roosevelt, Nixon, and Kennedy. President Roosevelt's speech was the longest, with 1,526 words before removing stopwords and 632 words after, followed by President Nixon with 2,006 words before stopwords and 843 words after, and President Kennedy with 1,543 words before stopwords and 696 words after. Removing stopwords significantly reduced the word counts for all three speeches, indicating the use of common words with limited meaning.

- While word counts offer insight into speech length, they do not reveal the content or themes discussed. To gain a deeper understanding, content analysis is required to identify key topics, sentiments, and rhetorical devices employed by each president. The years of the speeches (1941, 1961, and 1973) coincide with pivotal moments in U.S. history, such as World War II, the Cold War, and the Watergate scandal. Examining the historical context and events these speeches addressed would provide valuable insights into their significance.

2.3) Which word occurs the most number of times in his inaugural address for each president? Mention the top three words. (after removing the stopwords)

➢ **Apply the most_common_words function to each cleaned speech**

```
President Roosevelt (1941) Speech - Top Words:
nation: 17
know: 10
peopl: 9


President Kennedy (1961) Speech - Top Words:
let: 16
us: 12
power: 9


President Nixon (1973) Speech - Top Words:
us: 26
let: 22
america: 21
```

Fig 63 Most common words for each president's speech

Conclusion: The analysis of the most frequently used words in these speeches highlights the presidents' key themes and communication styles. President Roosevelt emphasizes national unity, awareness, and the importance of the people. President Kennedy's speech focuses on taking action, unity, and the concept of power in a Cold War context. President Nixon emphasizes national unity, action, and a strong sense of American identity.

2.4) Plot the word cloud of each of the three speeches. (after removing the stopwords)

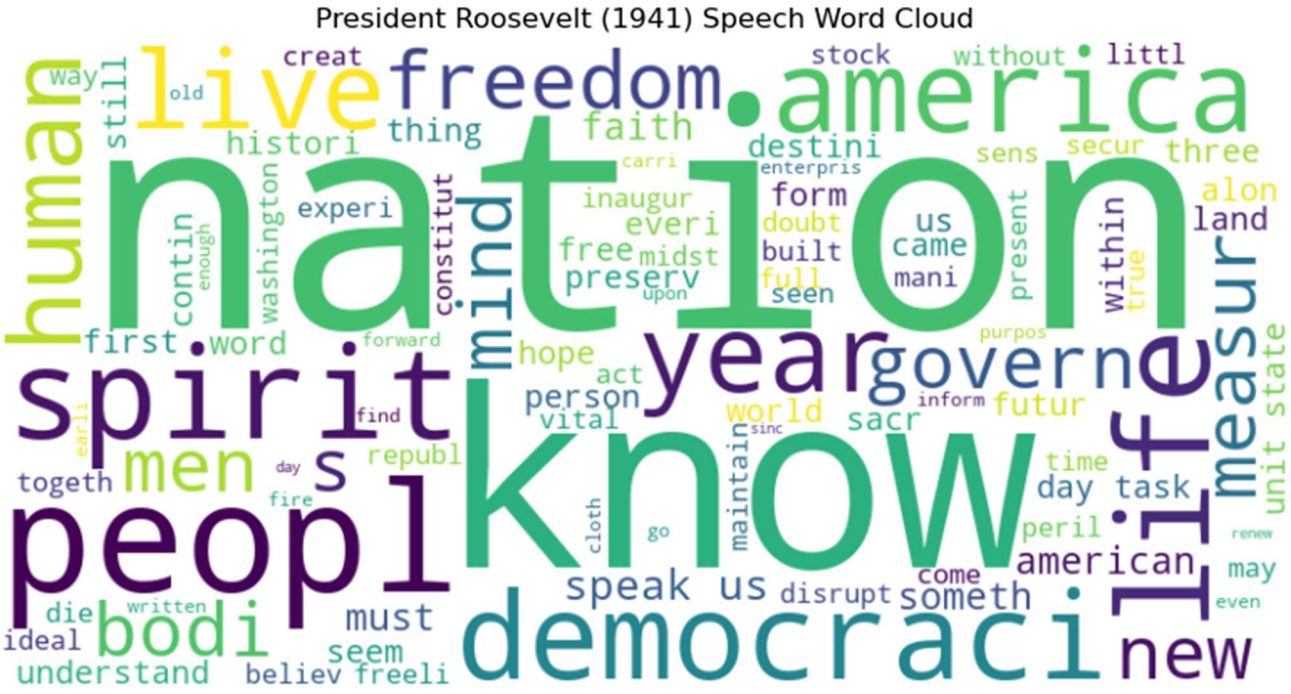

Fig 64 President Roosevelt (1941) Speech Word Cloud

Fig 65 Plot President Kennedy (1961) Speech Word Cloud

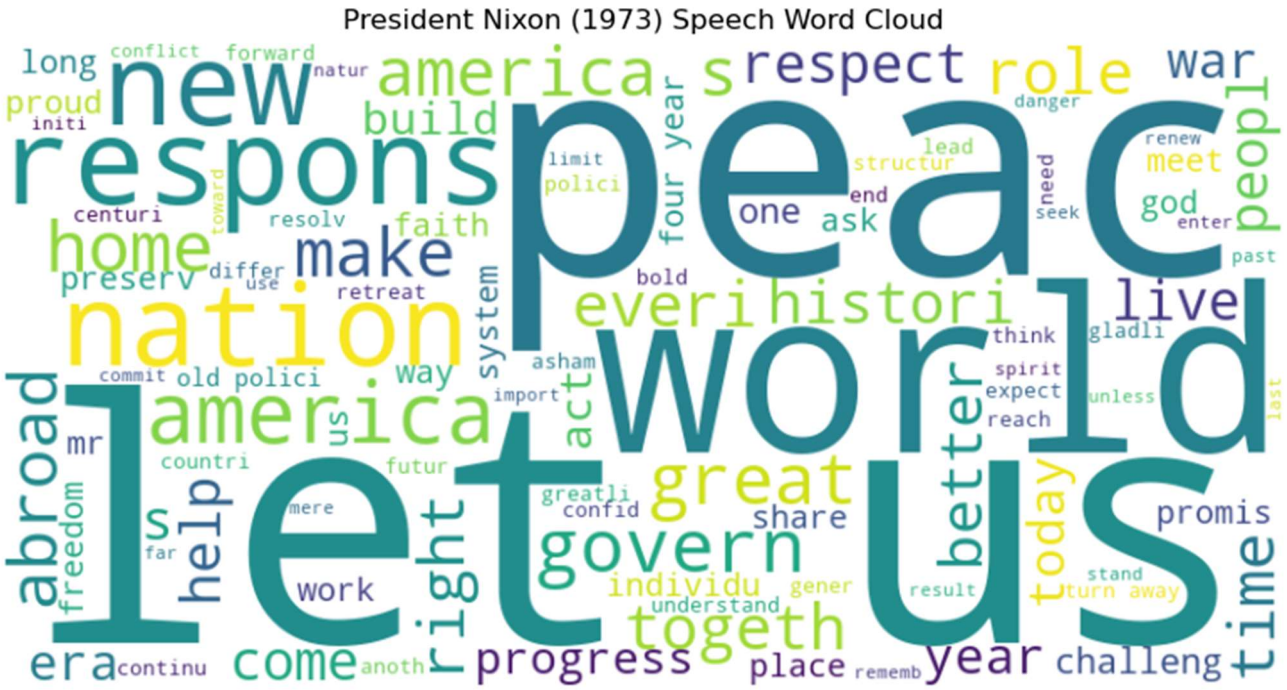

Fig 66 Plot President Nixon (1973) Speech Word Cloud

Insights:

Our objective was to look all the 3 speeches and analyse them. To find the strength and sentiment of the speeches. Based on the outputs we can see that there are some similar words (like nation, America, new, etc.) that are present in all the speeches.

These words may the point which inspired the many people and also get them the seat of president of United States of America.

Among all the speech “nation” is the word that is significantly highlighted in all three.