Machine learning -2
Business Report

Sanjana K Venkatesh

04-08-2024

Table of Contents

List of Figures

List of Tables

Executive Summary

CNBE, a prominent news channel, is gearing up to provide insightful coverage of recent elections, recognizing the importance of data-driven analysis. A comprehensive survey has been conducted, capturing the perspectives of 1525 voters across various demographic and socio-economic factors. This dataset encompasses 9 variables, offering a rich source of information regarding voters' characteristics and preferences.

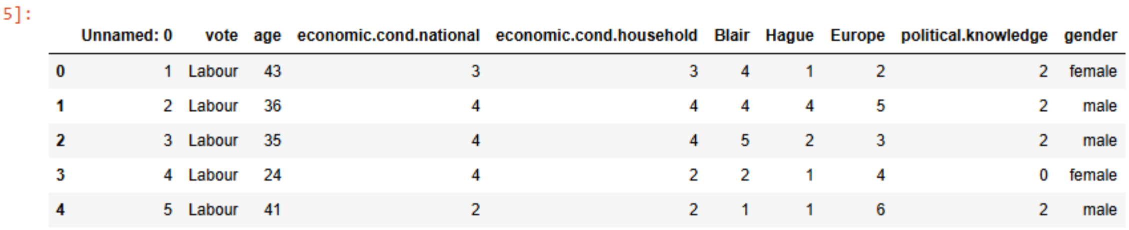
Objective

The primary objective is to leverage machine learning to build a predictive model capable of forecasting which political party a voter is likely to support. This predictive model, developed based on the provided information, will serve as the foundation for creating an exit poll. The exit poll aims to contribute to the accurate prediction of the overall election outcomes, including determining which party is likely to secure the majority of seats.

Data Description

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.

Sample of the dataset:



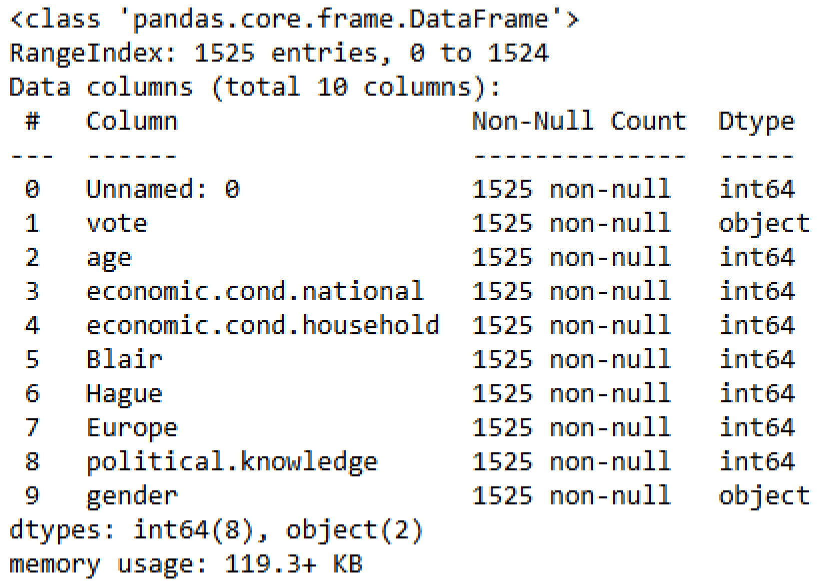
5]:

	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

Dataset is in shape of 1525, 10

Exploratory Data Analysis

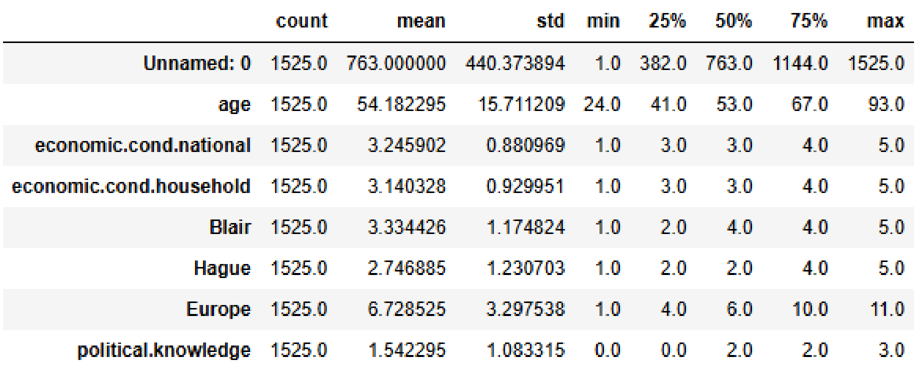
Let us check the types of variables in the data frame.



```
<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
```

There are total 10 columns. 8 are integer type and two are object.

Check for summary statistics



	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	1525.0	763.000000	440.373894	1.0	382.0	763.0	1144.0	1525.0
age	1525.0	54.182295	15.711209	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	3.245902	0.880969	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	3.140328	0.929951	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	3.334426	1.174824	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	2.746885	1.230703	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	6.728525	3.297538	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	1.542295	1.083315	0.0	0.0	2.0	2.0	3.0

Observations:

1. The min age group of people comes in 20's and max is going up to 93. So, there are no underage group of people who don't have right to vote.
2. The national economic conditional is a measure of economic health of the country at given time. Right from GDP, Inflation Rate, Unemployment rate, Business investment etc. The min is 1 and max is 5. The average stands around 3 and assessment of national for average 3 is 607 in total.

3. Economic conditions of households refer to how the individual house is affected with the economy change. For instance, increase of rent, Cost of living, increase in debt, income levels. Here the min value is 1 and max is 5 with average is 3. i.e. there are mid-range families more 648.

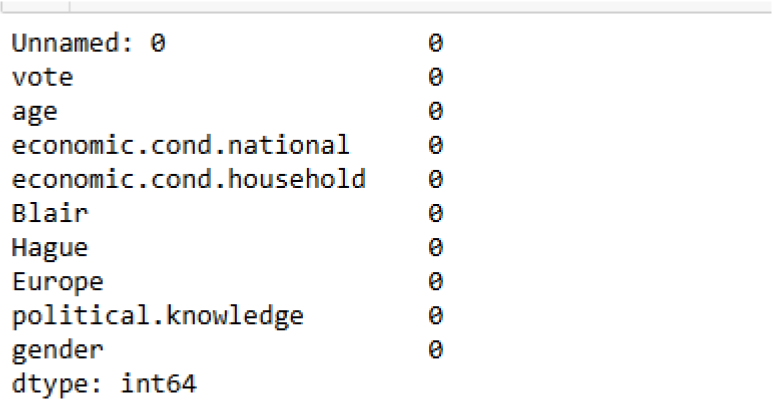
4. The Person Blair is evaluated for the position of labour leader from a scale of 1 to 5. The min assessment was 1 given by just 3 people that means he has performed very poorly. But looks like he has served pretty well since he is rated 4 by majority of people.

5. The Person Hague is evaluated for the position of conservative leader, an opposition party of labour leader. The min assessment was 1 given by 233 people that means he has performed very poorly. He was rated 4 by 558 people. But majority seems to rate him around 2. Hence, he might not be good choice.

6. A survey on political, social, economic, culture integration with European countries. The scale of highest vote refers people being not okay with the change. 338 people are feeling negative about the change and only 79 people are okay with it. On average 101 people are in between.

7. The knowledge on European parties towards the integration is in average numbers. There are 455 people who are unaware of this and only 250 of them are aware of the change.

Check for null values in the dataset:



```
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
```

There are no null values in the data.

Check for duplicate values in the dataset:

There are no duplicates in the dataset.

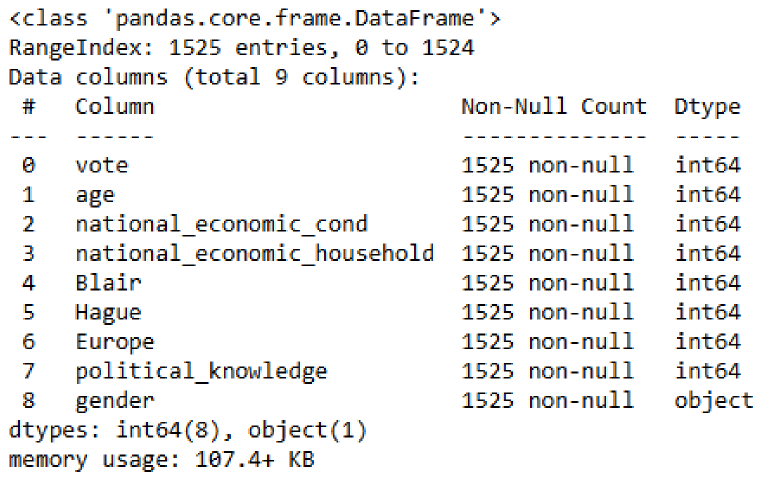
Drop the Unnamed column in data_df dataframe and rename columns and convert the categorical variable vote to numeric.

Columns renamed as below:

economic.cond.national : national_economic_cond

economic.cond.household :national_economic_household

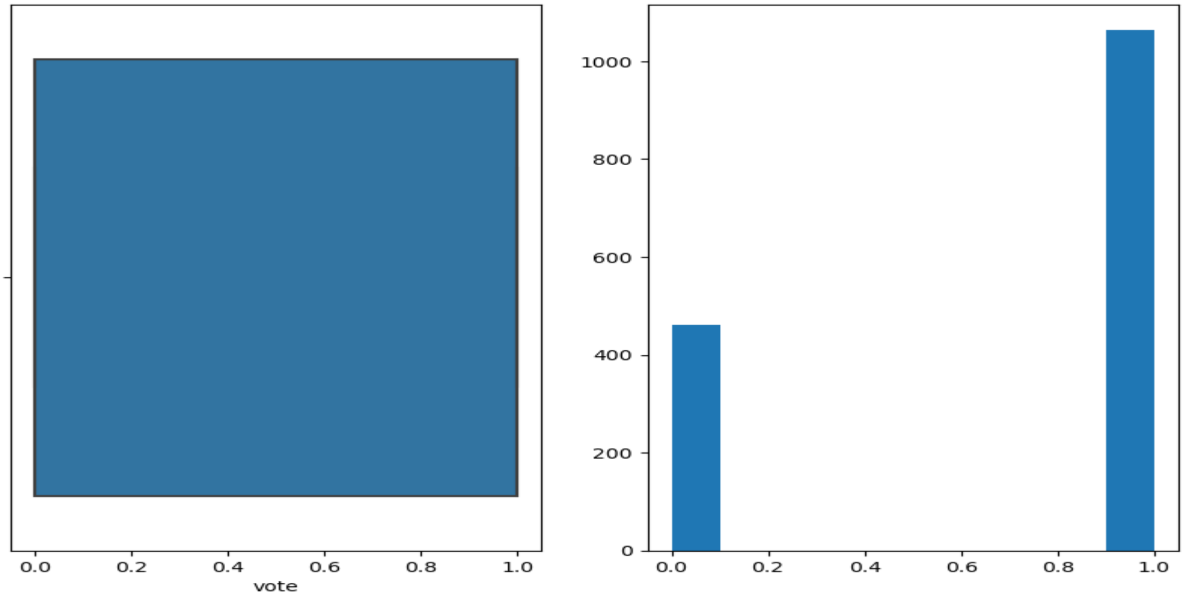
political.knowledge: political_knowledge



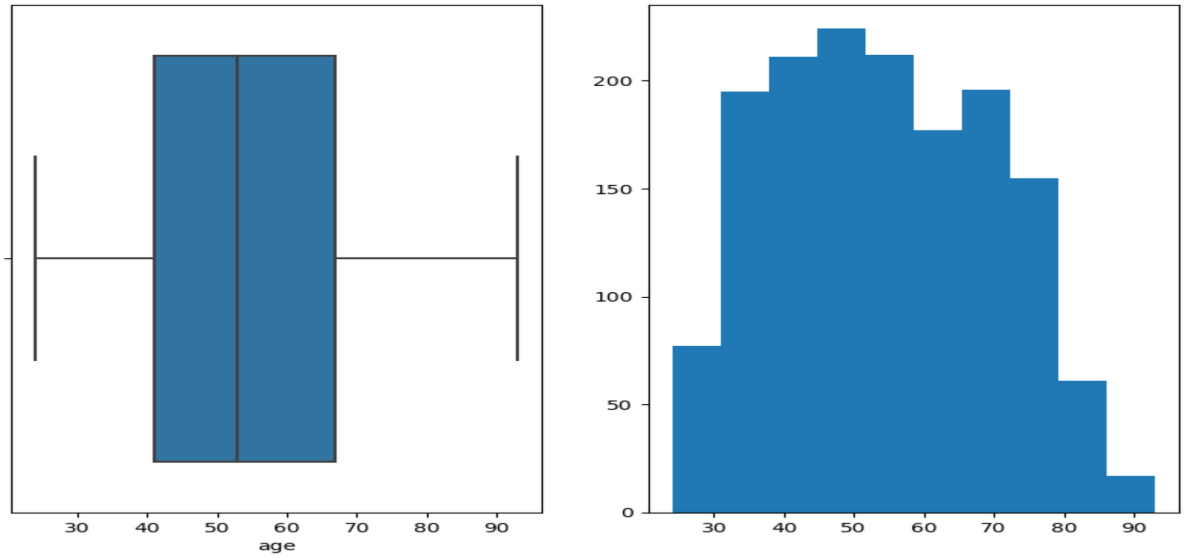
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 9 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   vote                         1525 non-null   int64 
 1   age                          1525 non-null   int64 
 2   national_economic_cond       1525 non-null   int64 
 3   national_economic_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(8), object(1)
memory usage: 107.4+ KB
```

Univariant analysis

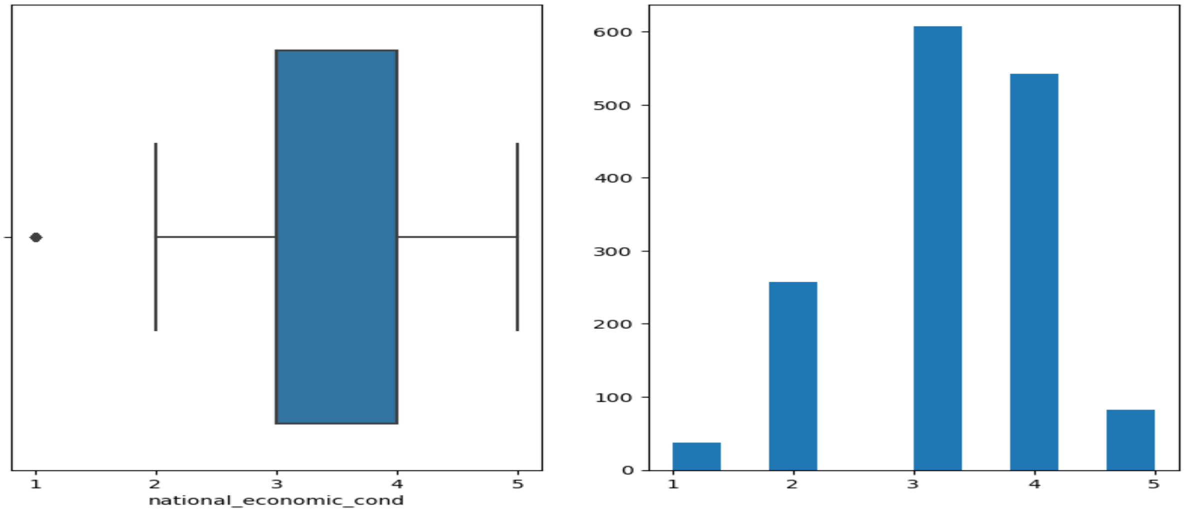
vote



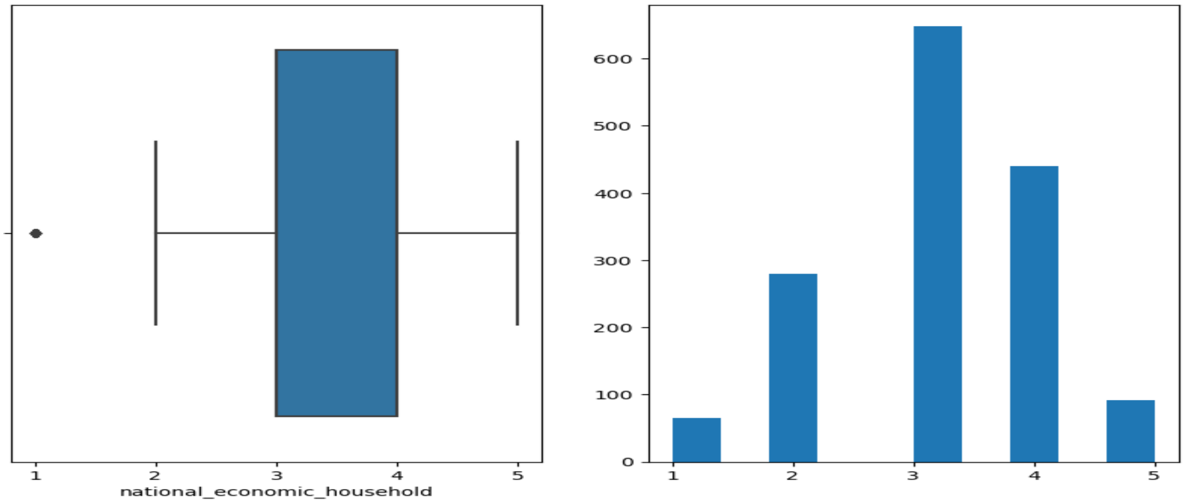
age



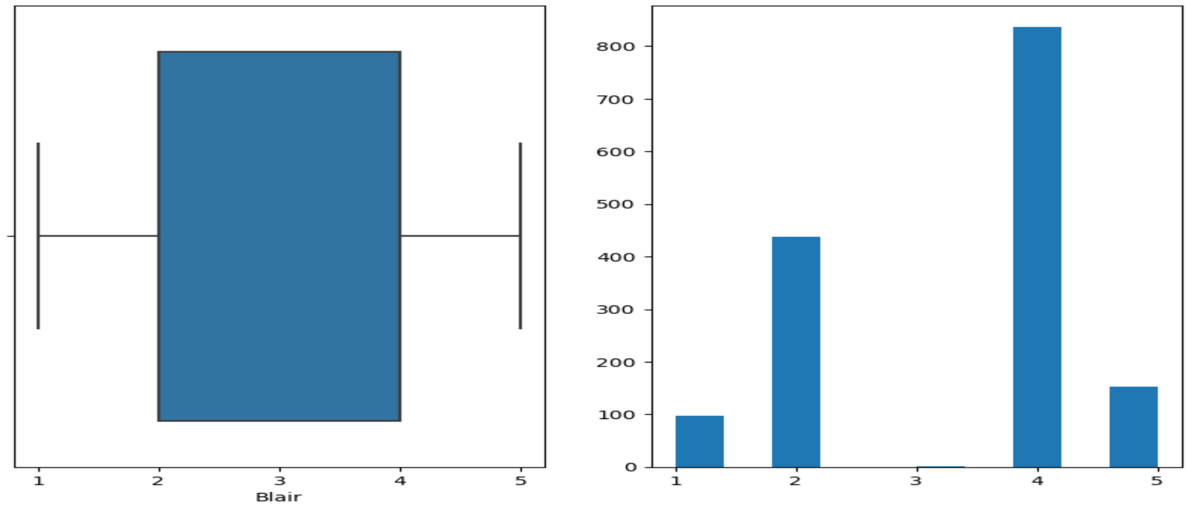
national_economic_cond



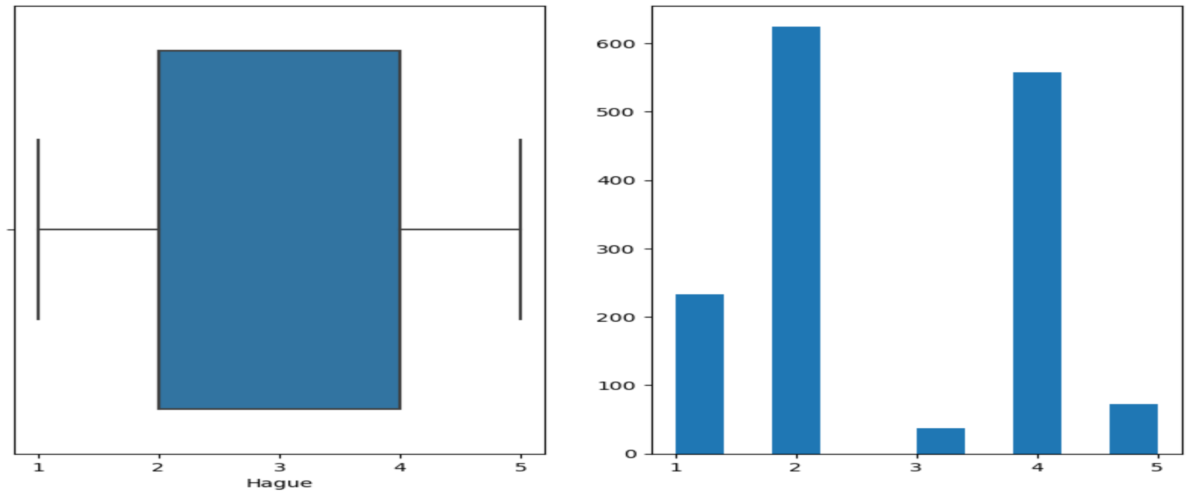
national_economic_household



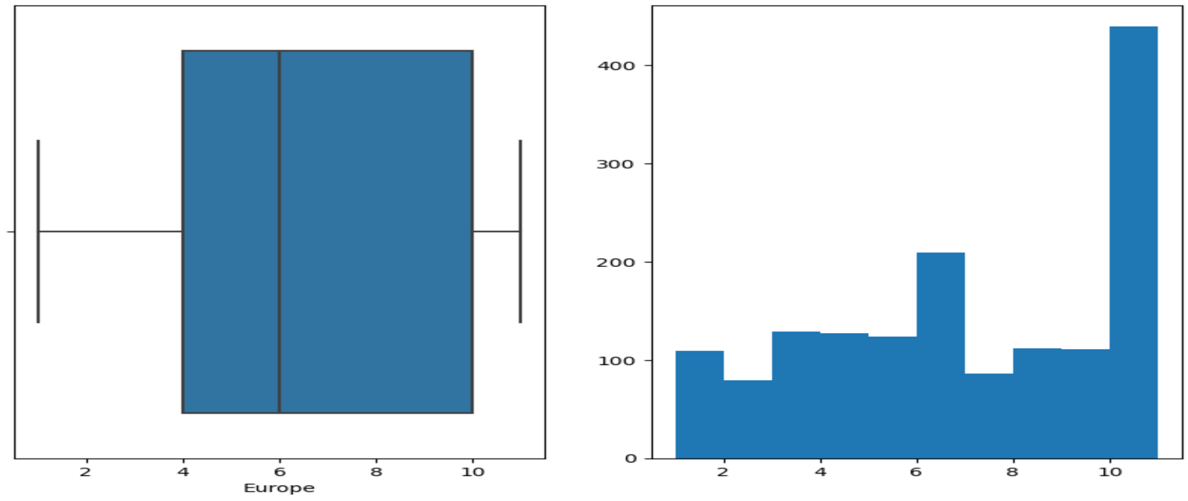
Blair



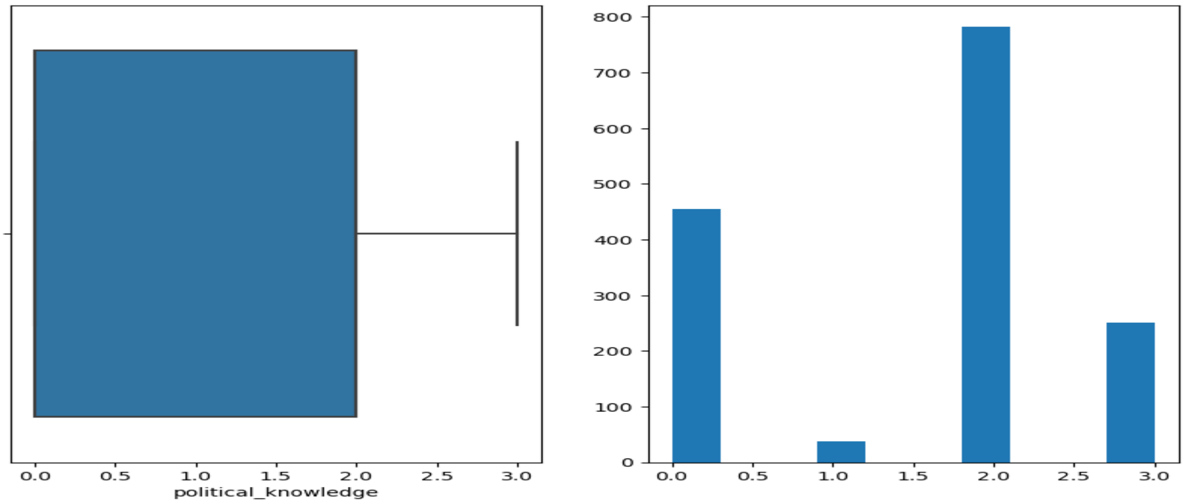
Hague



Europe



political_knowledge

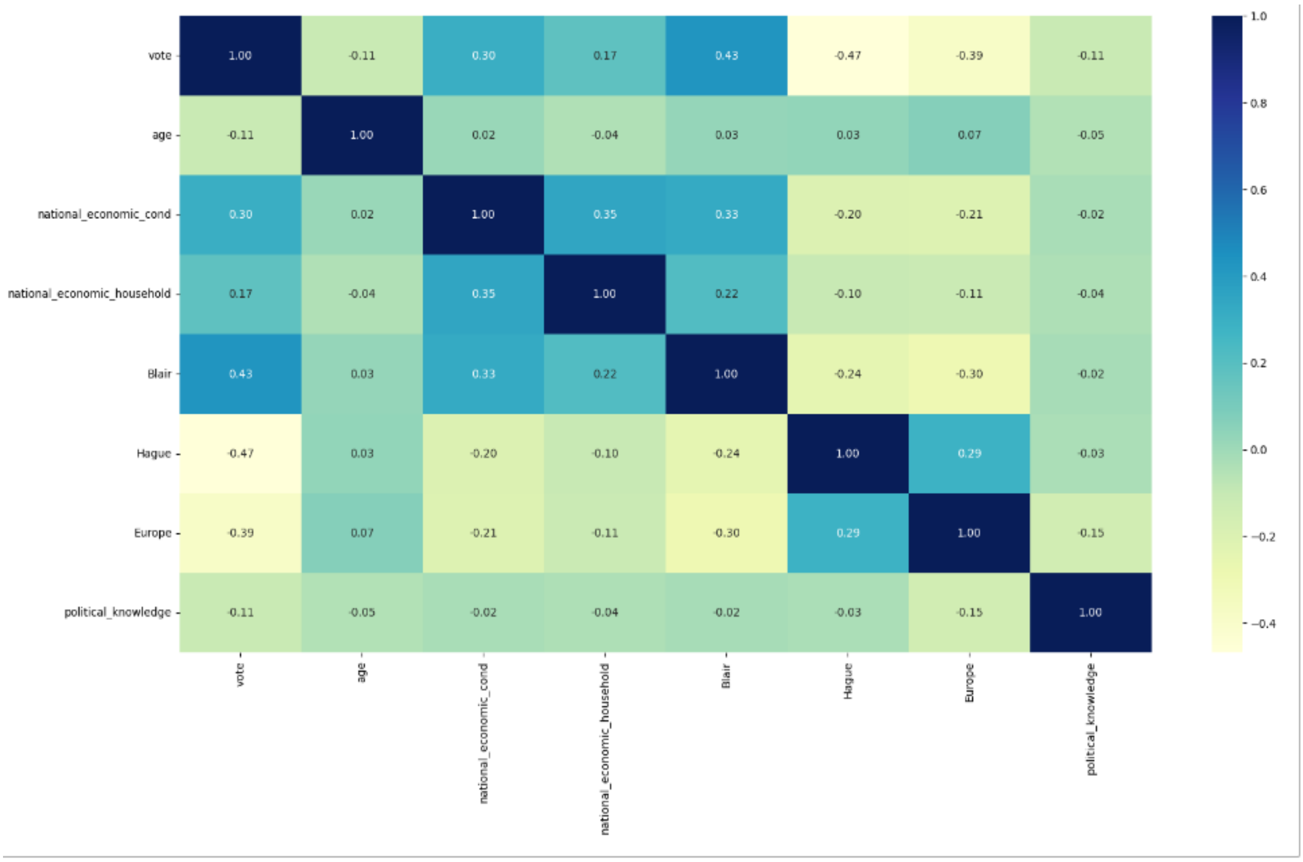


Observations:

1. The age is normally distributed.The median is 53 and mean is 54.
2. national_economic_cond and national_economic_household is mean is 3 and median is also 3 with max value of 5. Both are left skewed.
3. Blair the labour leader is distributed with max value as 5.
4. Hague the conservative leaders are distributed with mean and median as 2 and max value of 5.
5. sceptic political knowledge for the people is very minimal with min as 0 around 400+ people have no knowledge on europe integration.. Hence from visualization irs clear that there is no left whisker.
6. Europe feature has max value of 11 i.e there is a Eurosceptic sentiment.

Correlation plot:

Relation between all numeric variables



From the correlation plot, we can see that various attributes of the car are highly correlated to each other. Correlation values near to 1 or -1 are highly positively correlated and highly negatively correlated respectively. Correlation values near to 0 are not correlated to each other.

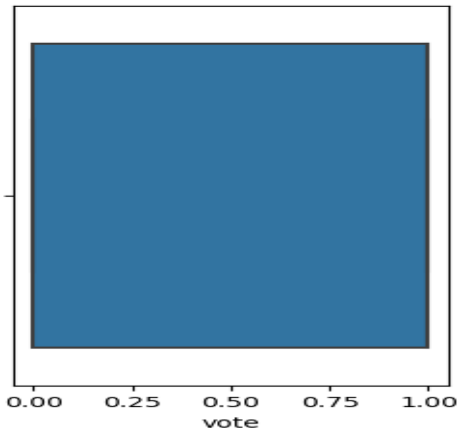
From the visualization:

1. national_economic_cond and national_economic_housegold are postively correlated and vicer versa.
2. Blair and Europe are negatively correlated.

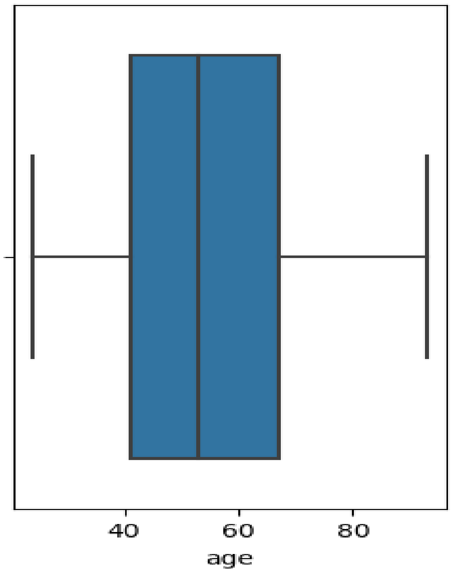
Outlier treatment

After the box plot technique of outlier treatment, the outliers are removed. It uses a technique of lower index and upper index for eliminating the outliers.

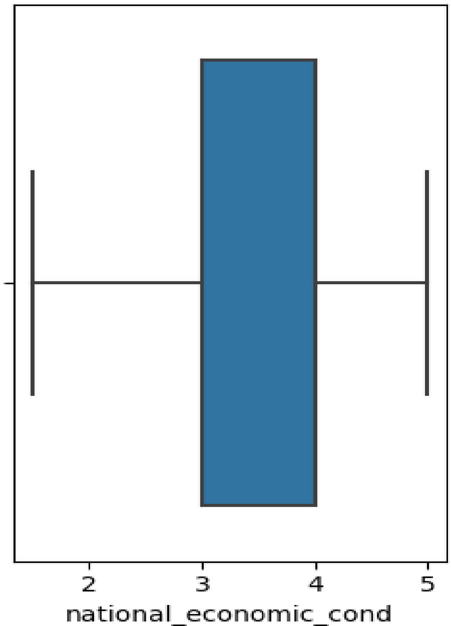
lower range -1.5 and upper range 2.5



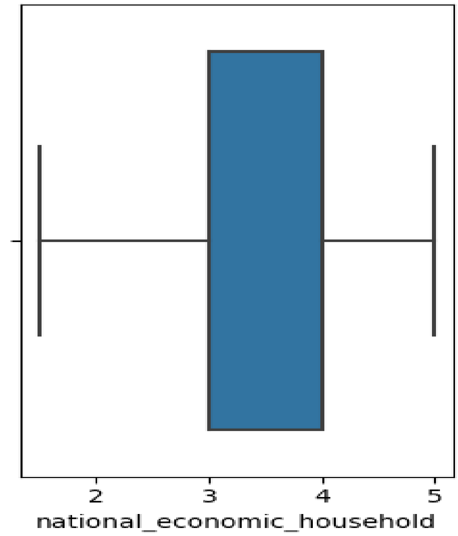
lower range 2.0 and upper range 106.0



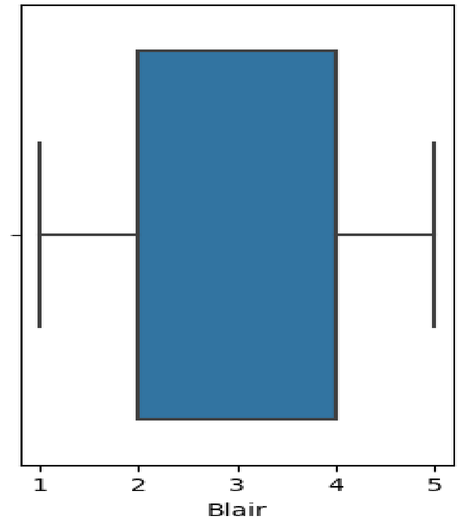
lower range 1.5 and upper range 5.5



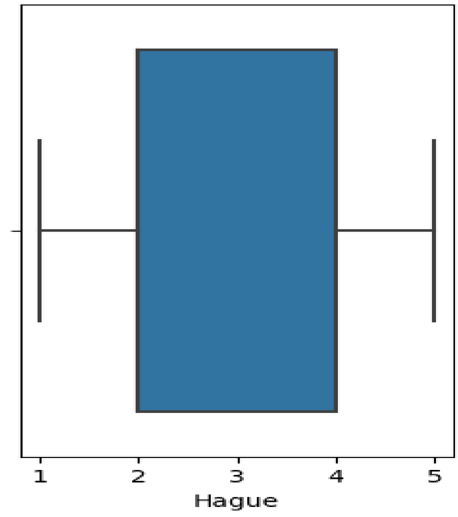
lower range 1.5 and upper range 5.5



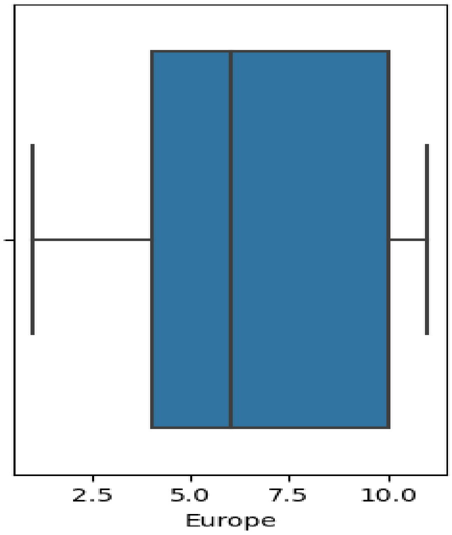
lower range -1.0 and upper range 7.0



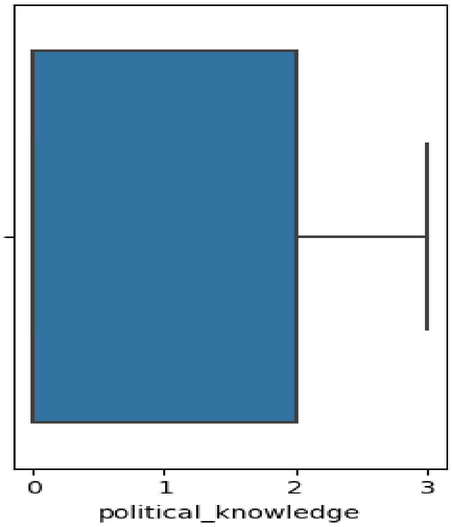
lower range -1.0 and upper range 7.0



lower range -5.0 and upper range 19.0

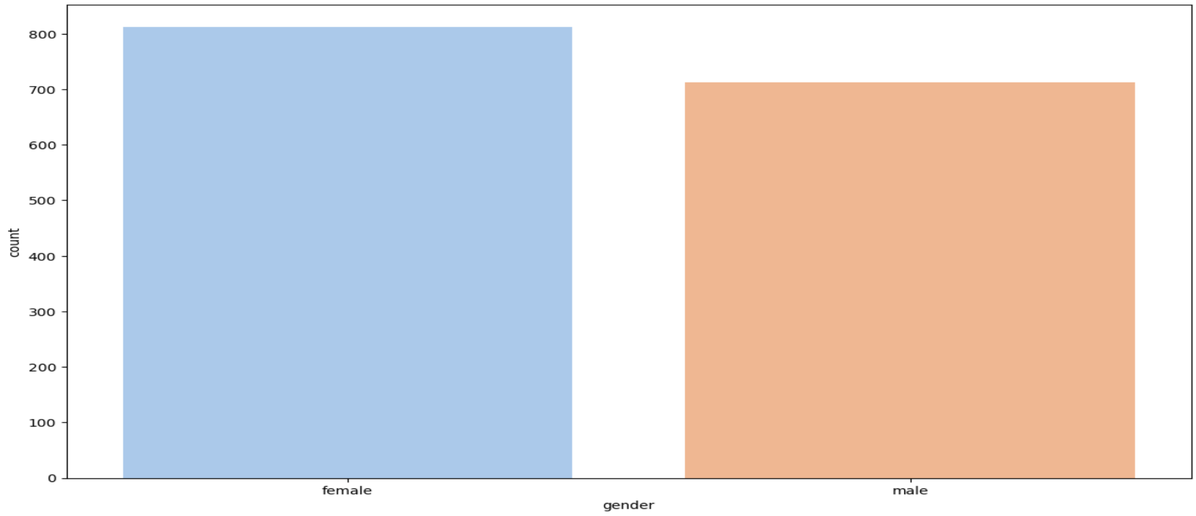


lower range -3.0 and upper range 5.0



Univariant analysis for Categorical variable

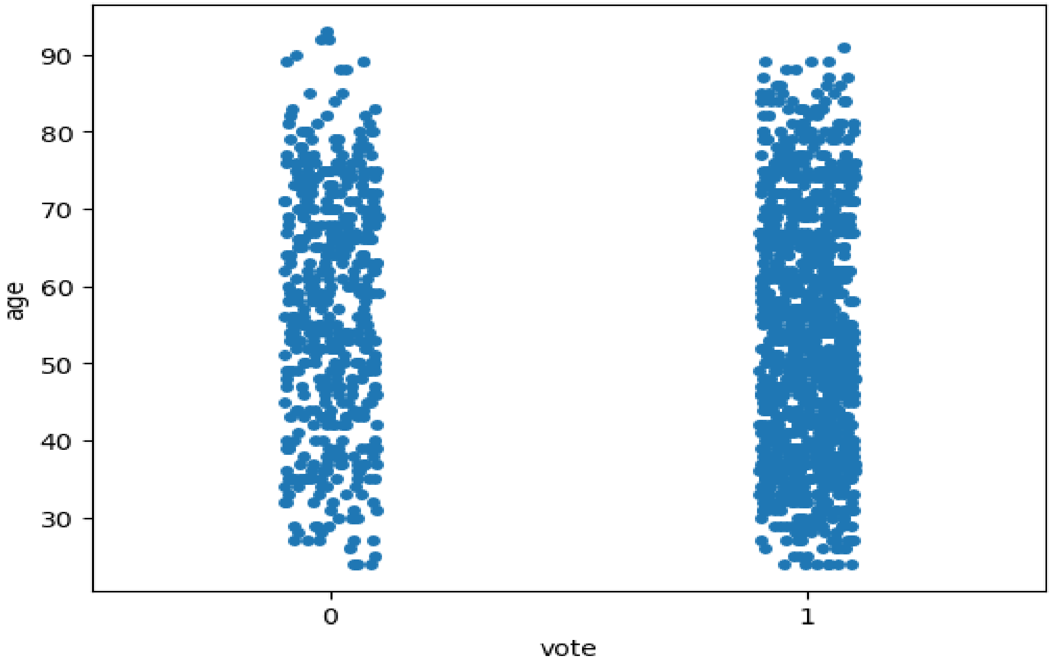
gender



Observation:

1. The Female population is more as compared to male. The population of female is 800+ and male is around 750+.

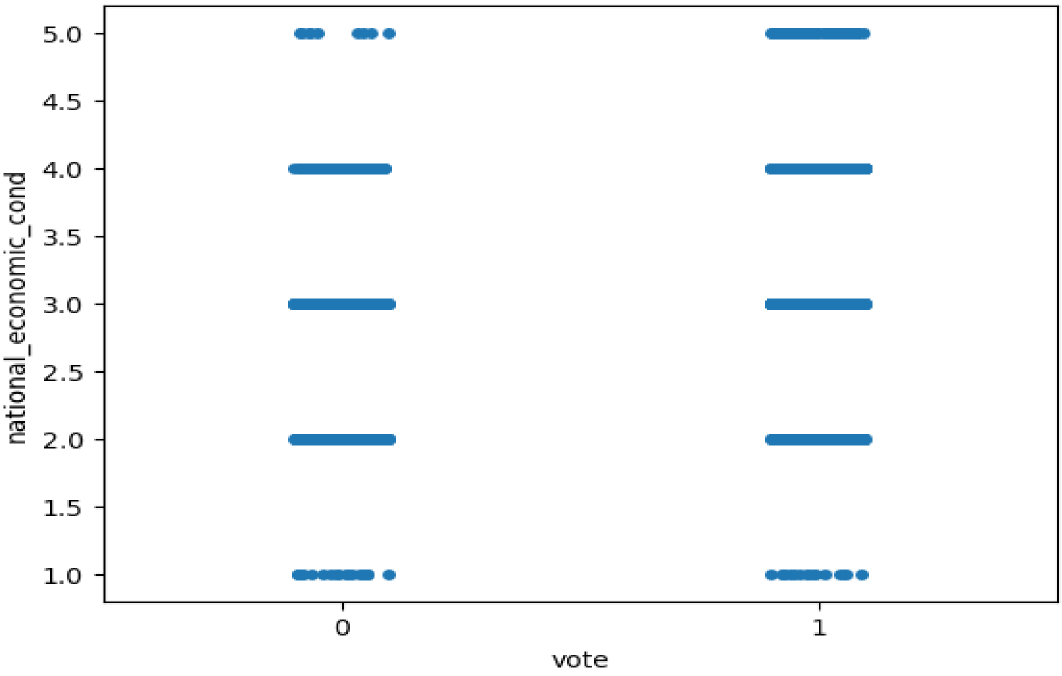
Bivariant analysis



Observation:

Different age group people are present. The graph depicts that most of the younger and middle age vote for Labour leader.

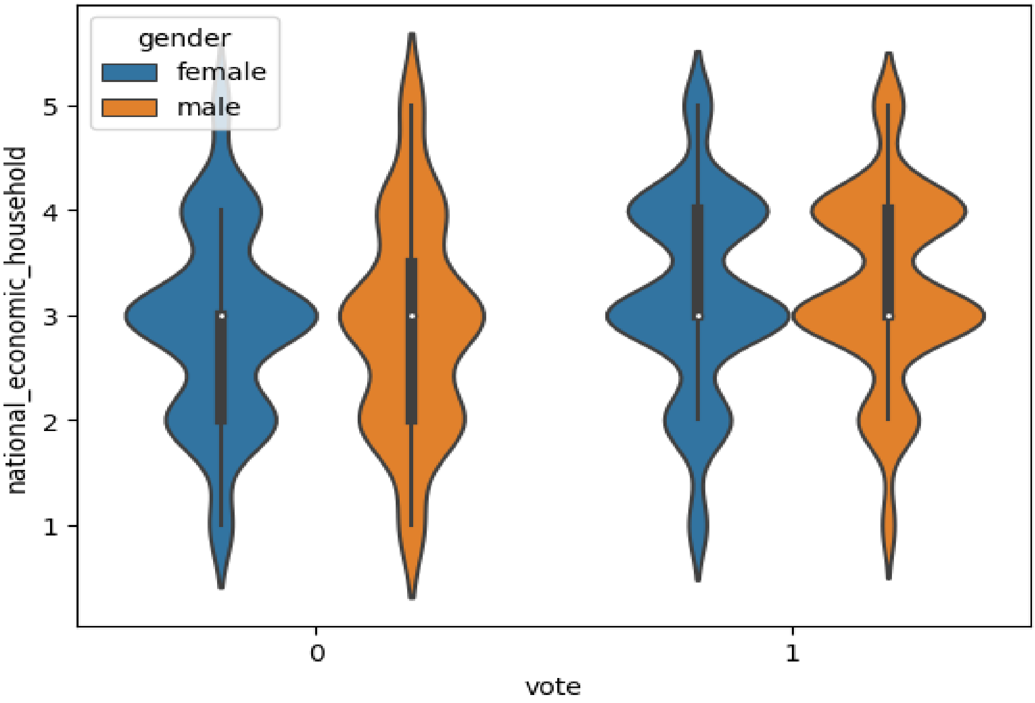
Axes: xlabel='vote', ylabel='national_economic_cond'>



Observation:

The economic national conditional looks almost smae for both the parties. At top 5 the Labour party leader looks good.

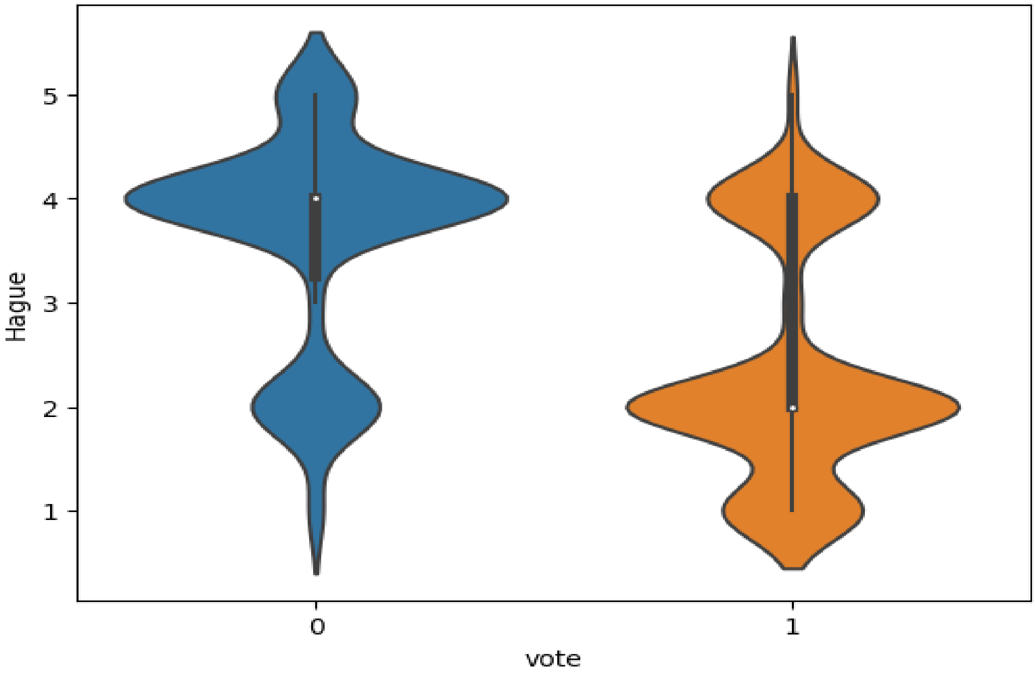
<Axes: xlabel='vote', ylabel='national_economic_household'>



Observation:

econmic household of the nation looks equally distributed. The Median is around 3.

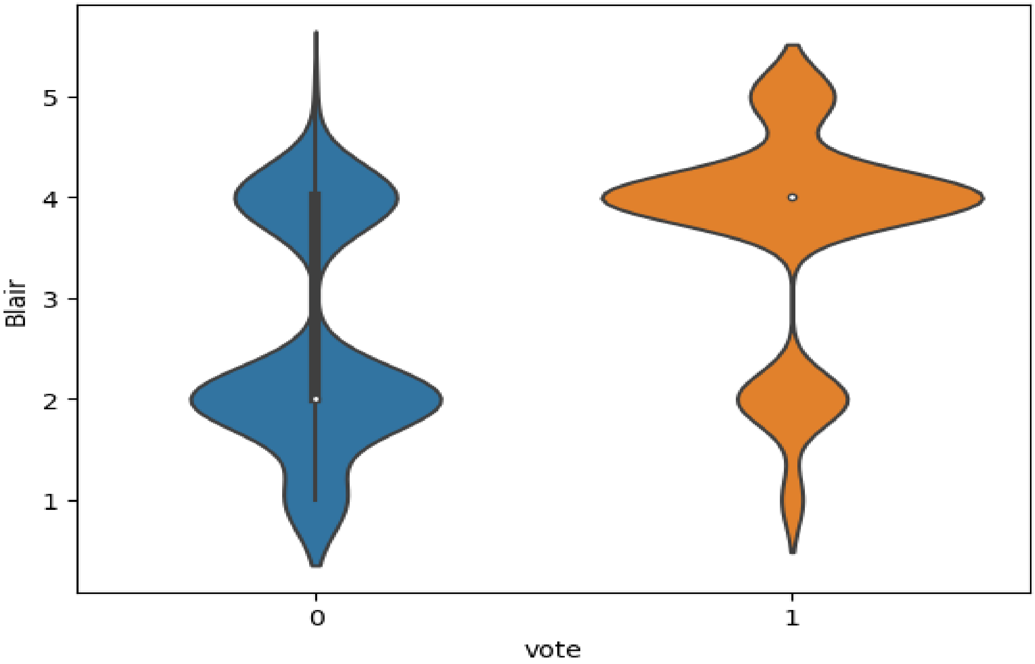
<Axes: xlabel='vote', ylabel='Hague'>



Observation:

The conservative leader has median around 4 and for labour its around 2 that means more positive votes for Hague from conservative party.

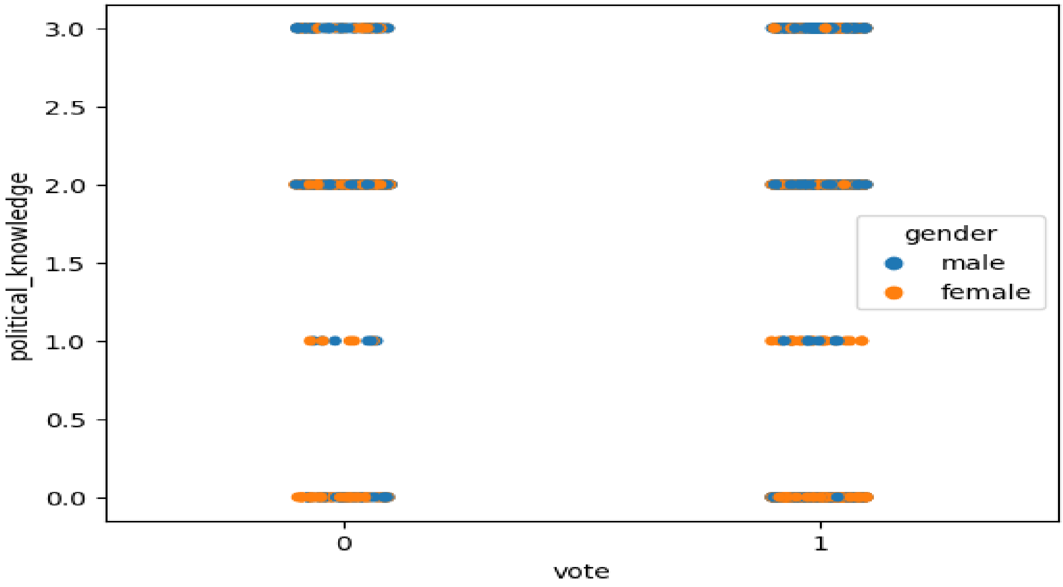
<Axes: xlabel='vote', ylabel='Blair'>



Observation:

The conservative leader is having median around 2 and violin plot looks close to normal. But labour leader is not equally distributed. the number of people opting Labour is more than other.

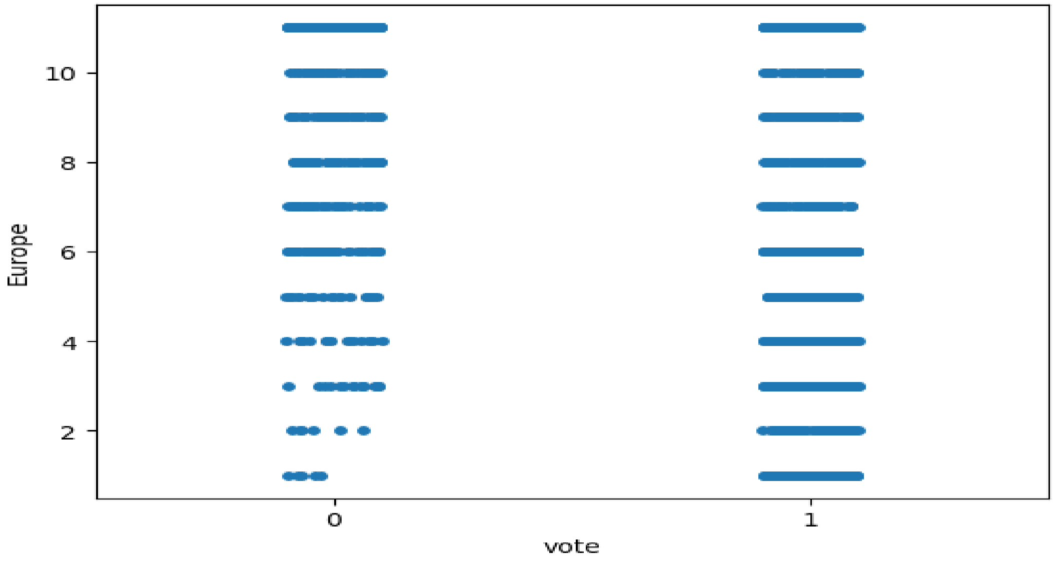
```
<Axes: xlabel='vote', ylabel='political_knowledge'>
```



Observation:

The women have least knowledge on the politics.Knowing about the politics and then voting looks like labour leader party has majority.

```
<Axes: xlabel='vote', ylabel='Europe'>
```

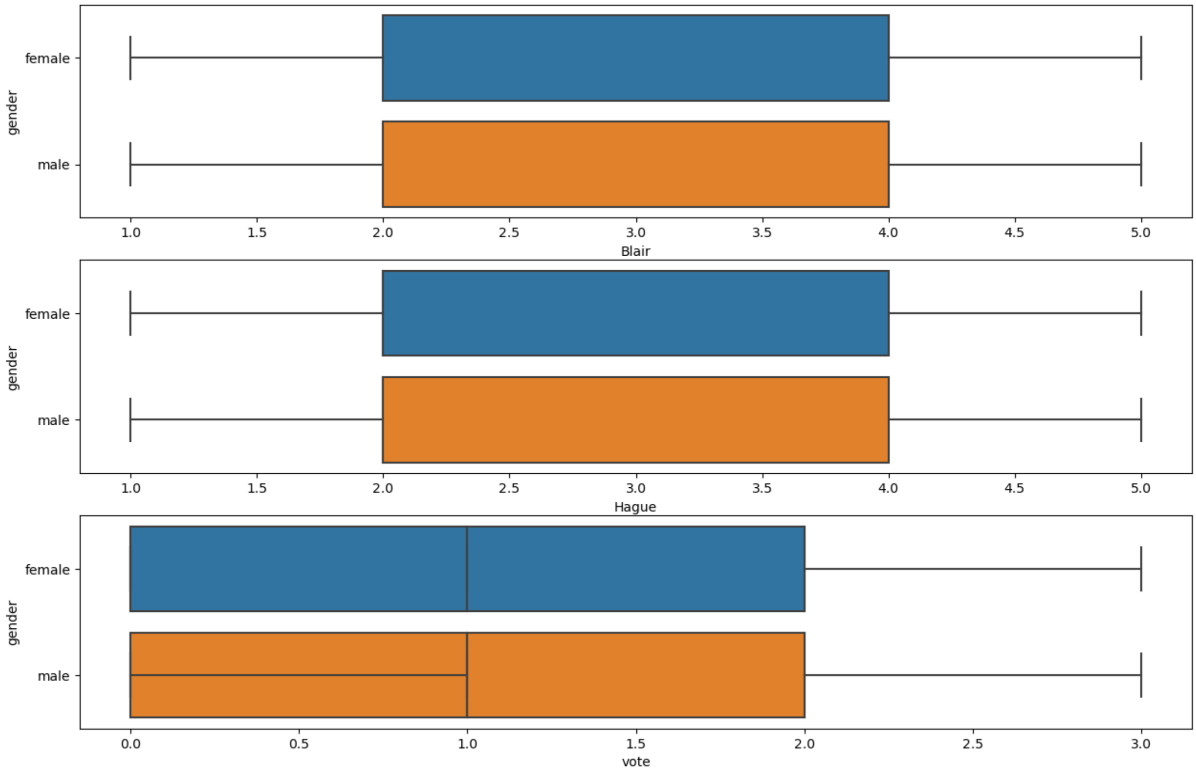


Observation:

Conservative party is bit unstable compared to Labour leader party. The majority of votes are or Labour leader.

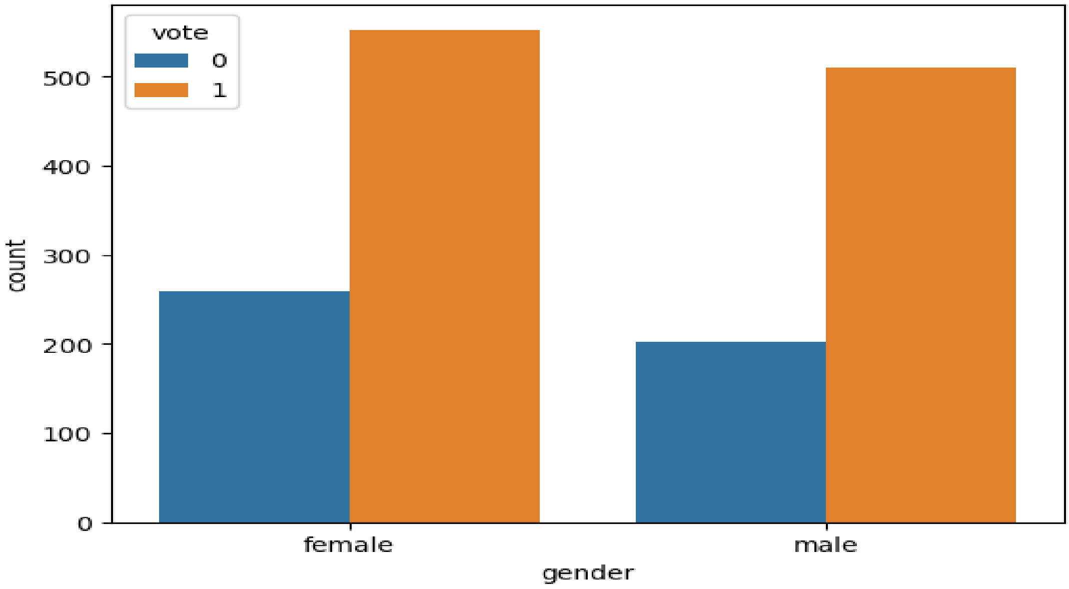
Categorical vs numerical

```
<Axes: xlabel='vote', ylabel='gender'>
```



Multivariant analysis

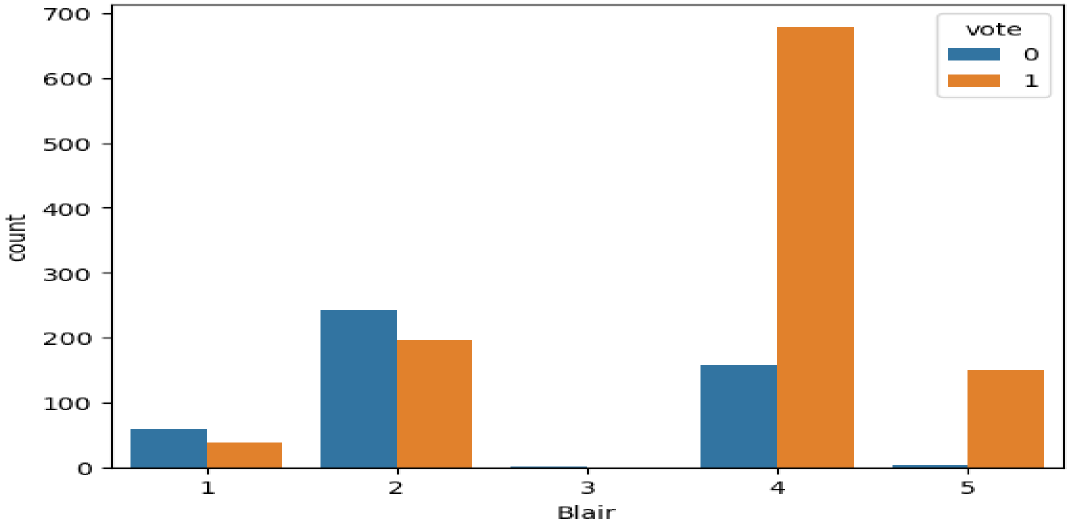
```
<Axes: xlabel='gender', ylabel='count'>
```



Observation:

Irrespective of gender the Labour leader is getting more vote.

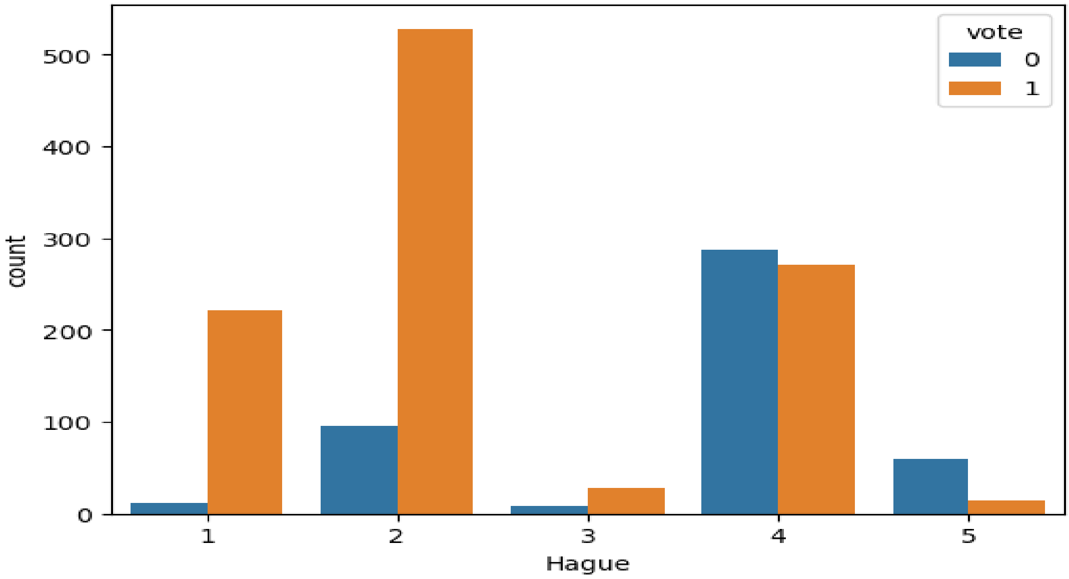
<Axes: xlabel='Blair', ylabel='count'>



Observation:

Blair is getting more vote from labour leader and the way they have rated is 4. So the service to public is good.

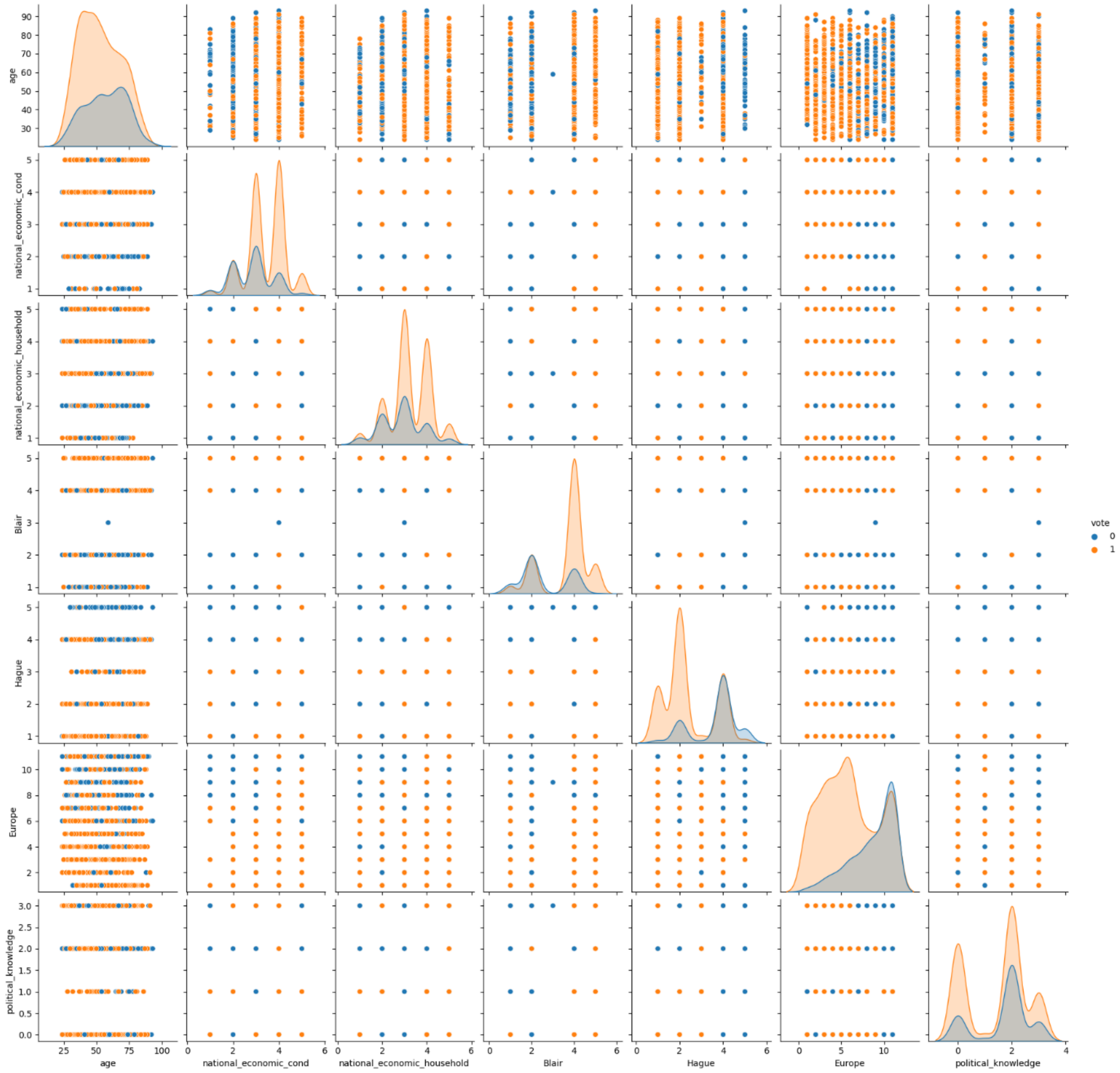
<Axes: xlabel='Hague', ylabel='count'>



Observation:

Hague is getting more vote in labour leader and the way they have rated is 2. So, the service to public is not so good. Only 300 people think he is fine. But 500+ people belive he is not good fit

```
<seaborn.axisgrid.PairGrid at 0x142b2b2b250>
```


age
90
80
70
60
50
40
30
national_economic_cond
national_economic_household
Blair
Hague
5
4
3
2
1
Europe
10
8
6
4
2
political_knowledge
3.0
2.5
2.0
1.5
1.0
0.5
0.0
25
50
75
100
0
2
4
6
0
5
10
0
2
4
vote
0
1

Overview of Naïve Bayes Model

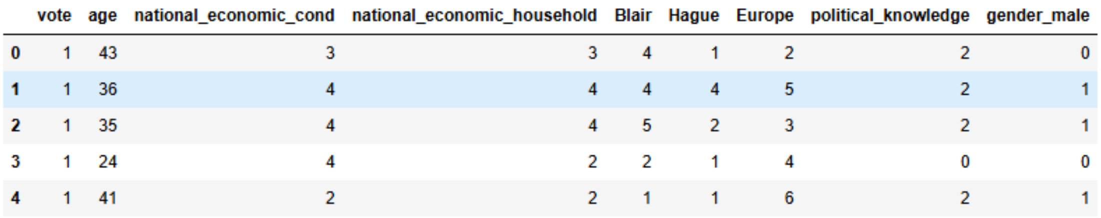
Naive Bayes is a algorithm for classification tasks, when dealing with high-dimensional data. The algorithm is based on Naïve Bayes Theorem with the assumption of independence between every pair of features.

Steps in Naïve Bayes:

1. Model Building: Fit the NB model.
2. Model Evaluation: Precision, Recall, F1 score and accuracy.
3. Model Prediction: Use the model built for predicting on new dataset.

Model Building:

1. Create dummy variables for the categorical variables.

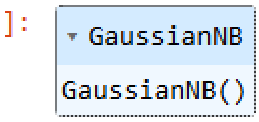


	vote	age	national_economic_cond	national_economic_household	Blair	Hague	Europe	political_knowledge	gender_male
0	1	43	3	3	4	1	2	2	0
1	1	36	4	4	4	4	5	2	1
2	1	35	4	4	5	2	3	2	1
3	1	24	4	2	2	1	4	0	0
4	1	41	2	2	1	1	6	2	1

2. On the dummy data, drop the target variable and save to X and pop the target variable to y.

3. Split the data in train and test. Test data as 30% and train data as 70%. Using train_test_split function.

4. Fit the Gaussian model using fit function passing the Y and X train data.



```
]:  ▾ GaussianNB
    GaussianNB()
```

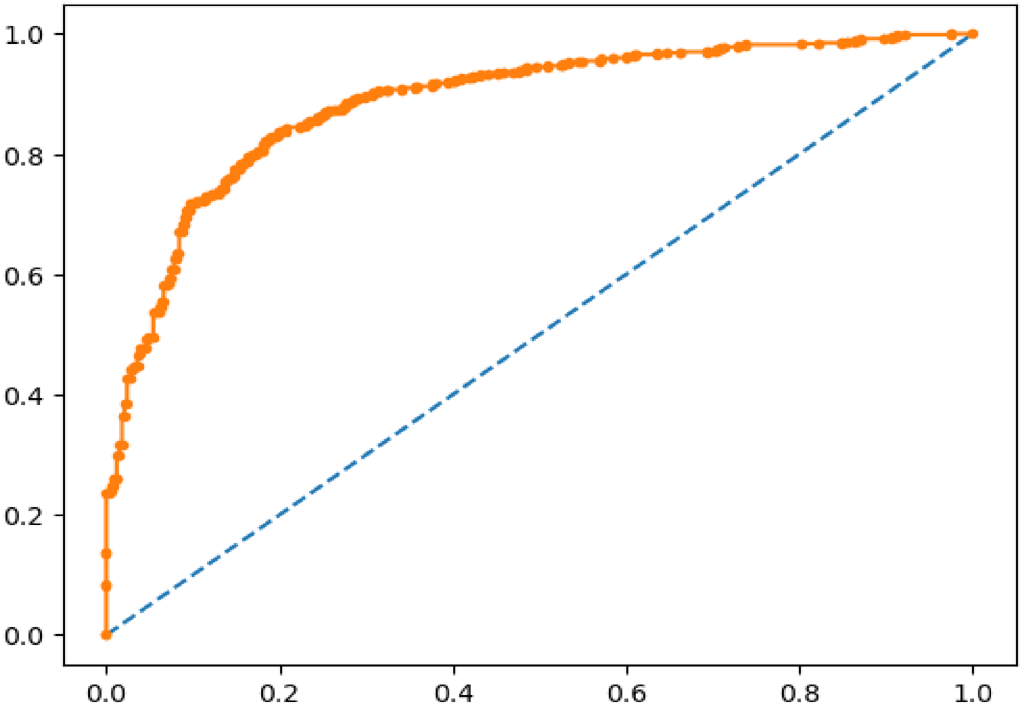
5. Prediction on train and test data.

6. Evaluate the model. Calculate the mode score to get accuracy from both training and test data. For training the accuracy is 83%. and test data accuracy is 83%.

7. Get the confusion matrix and AUC-ROC curve for both the dataset.

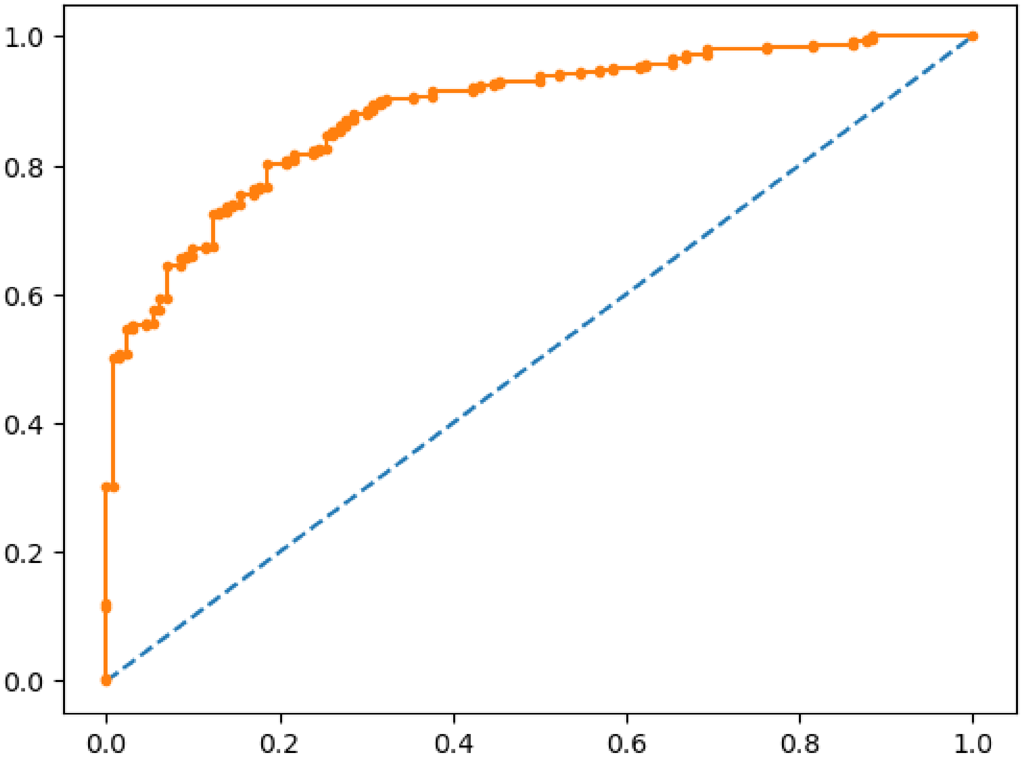
AUC-ROC Curve for train data

AUC: 0.886



AUC-ROC Curve for test data

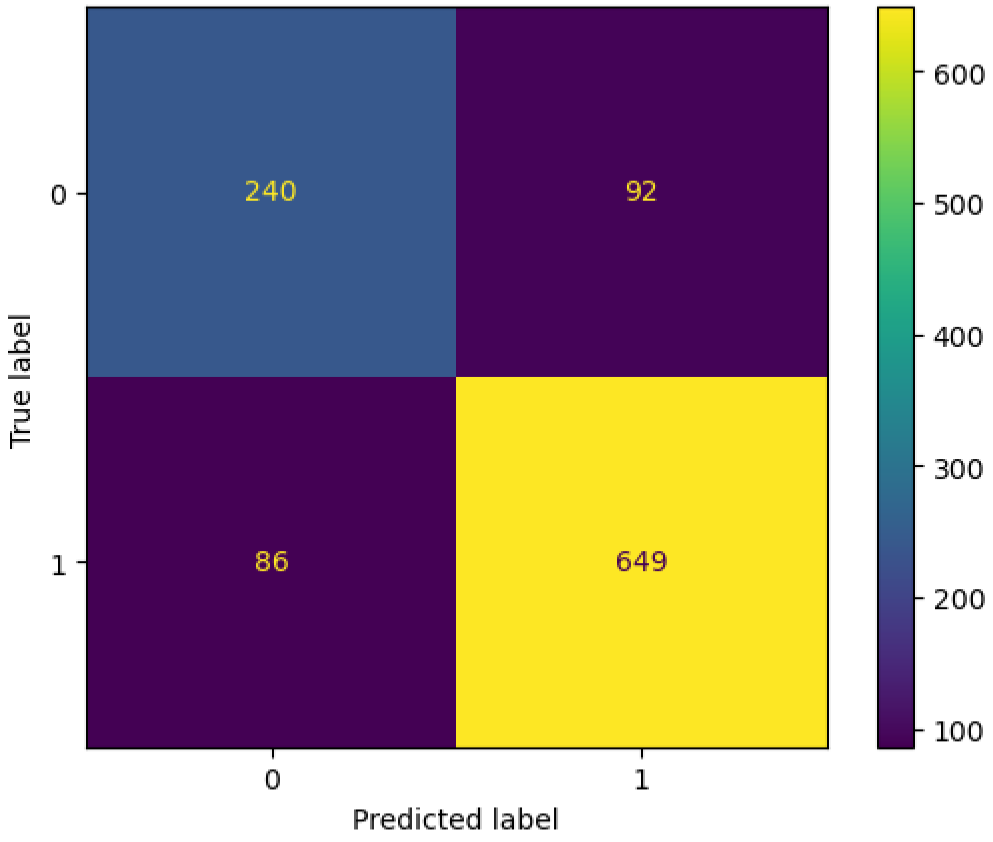
AUC: 0.885



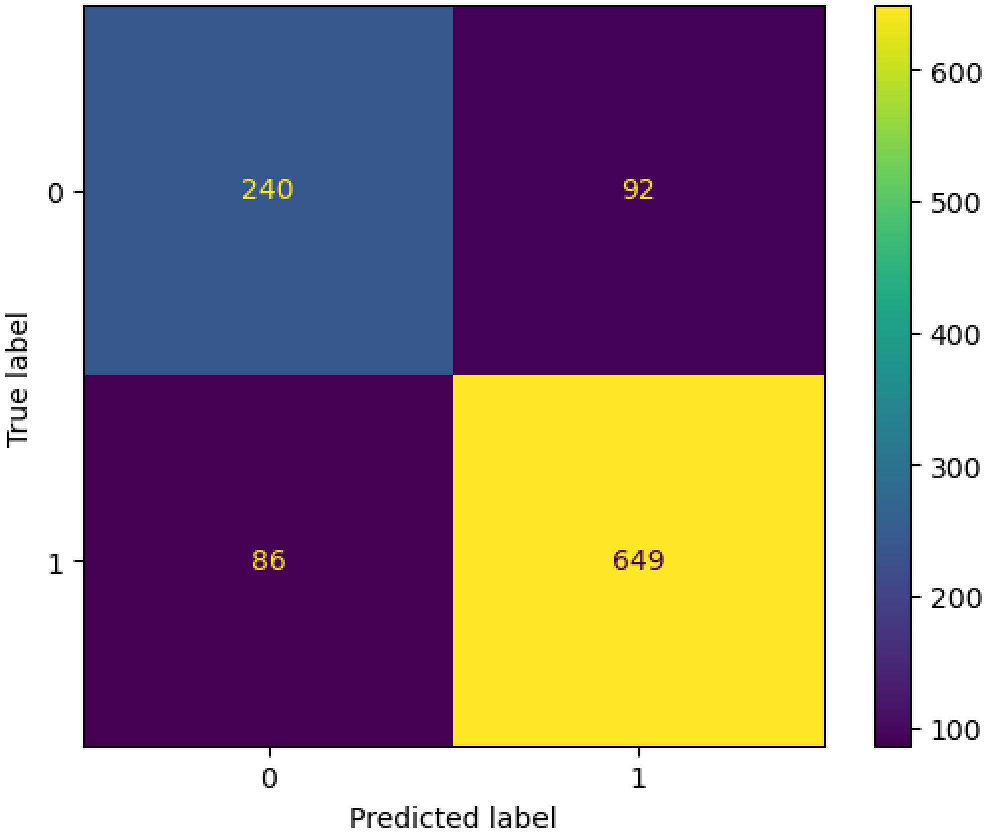
Classification report for train and test data

Train Data	Test data
precision recall f1-score support 0 0.74 0.72 0.73 332 1 0.88 0.88 0.88 735 accuracy 0.83 1067 macro avg 0.81 0.80 0.80 1067 weighted avg 0.83 0.83 0.83 1067	precision recall f1-score support 0 0.68 0.72 0.70 130 1 0.89 0.87 0.88 328 accuracy 0.83 458 macro avg 0.78 0.79 0.79 458 weighted avg 0.83 0.83 0.83 458

Confusion Matrix for Train data



Confusion Matrix for Test data



Actionable Insights:

1) The model has accuracy of 83%. Which is good fit for classification.
2) The model is having pretty high for recall around 87%, which means its able to rectify the actual true positives. The type II error is very minimal.
3) The precision is 0.89 for the class of interest. Again, the type I error is minimal.

Overview of KNN

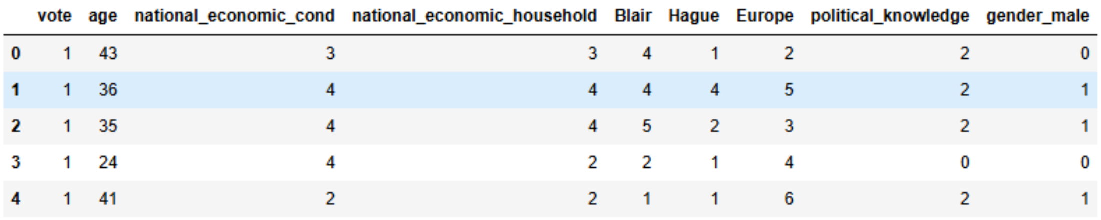
K-Nearest Neighbours (KNN) is a simple algorithm used for classification and regression tasks. It operates by finding the most similar data points (neighbours) in the training set and making predictions based on those neighbours. It uses few distance measure like Euclidean distance, Manhattan distance, and Minkowski distance.

Steps to Build KNN Model:

1. Model Building: Fit the KNN model.
2. Model Evaluation: Precision, Recall, F1 score and accuracy.
3. Model Prediction: Use the model built for predicting on new dataset. AUC and ROC curve.

Model Building:

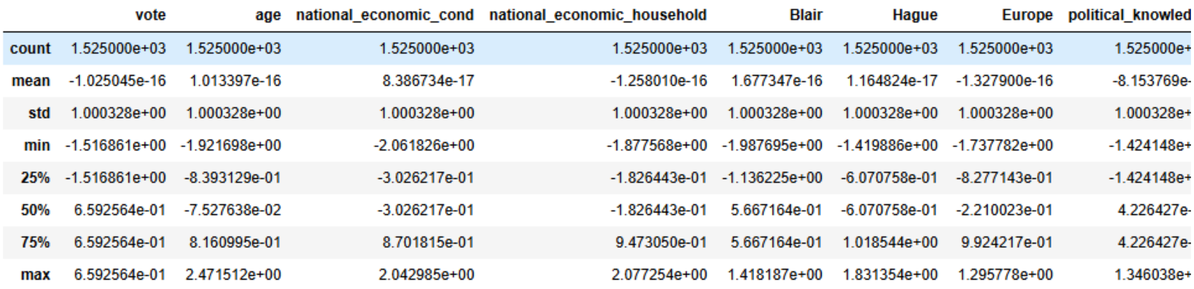
1. Create dummy variables for the categorical variables.



	vote	age	national_economic_cond	national_economic_household	Blair	Hague	Europe	political_knowledge	gender_male
0	1	43	3	3	4	1	2	2	0
1	1	36	4	4	4	4	5	2	1
2	1	35	4	4	5	2	3	2	1
3	1	24	4	2	2	1	4	0	0
4	1	41	2	2	1	1	6	2	1

2. On the dummy data, drop the target variable and save to X and pop the target variable to y.

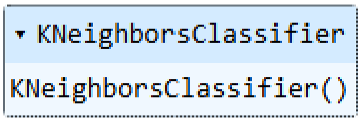
3. Scale the data. Apply the zscore on the feature.



	vote	age	national_economic_cond	national_economic_household	Blair	Hague	Europe	political_knowled
count	1.525000e+03	1.525000e+03	1.525000e+03	1.525000e+03	1.525000e+03	1.525000e+03	1.525000e+03	1.525000e+
mean	-1.025045e-16	1.013397e-16	8.386734e-17	-1.258010e-16	1.677347e-16	1.164824e-17	-1.327900e-16	-8.153769e-
std	1.000328e+00	1.000328e+00	1.000328e+00	1.000328e+00	1.000328e+00	1.000328e+00	1.000328e+00	1.000328e+
min	-1.516861e+00	-1.921698e+00	-2.061826e+00	-1.877568e+00	-1.987695e+00	-1.419886e+00	-1.737782e+00	-1.424148e+
25%	-1.516861e+00	-8.393129e-01	-3.026217e-01	-1.826443e-01	-1.136225e+00	-6.070758e-01	-8.277143e-01	-1.424148e+
50%	6.592564e-01	-7.527638e-02	-3.026217e-01	-1.826443e-01	5.667164e-01	-6.070758e-01	-2.210023e-01	4.226427e-
75%	6.592564e-01	8.160995e-01	8.701815e-01	9.473050e-01	5.667164e-01	1.018544e+00	9.924217e-01	4.226427e-
max	6.592564e-01	2.471512e+00	2.042985e+00	2.077254e+00	1.418187e+00	1.831354e+00	1.295778e+00	1.346038e+

4. Split the data in train and test. Test data as 30% and train data as 70%. Using train_test_split function.

5. Fit the KNN model using fit function passing the Y and X train data.



```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

6. Prediction on train and test data.

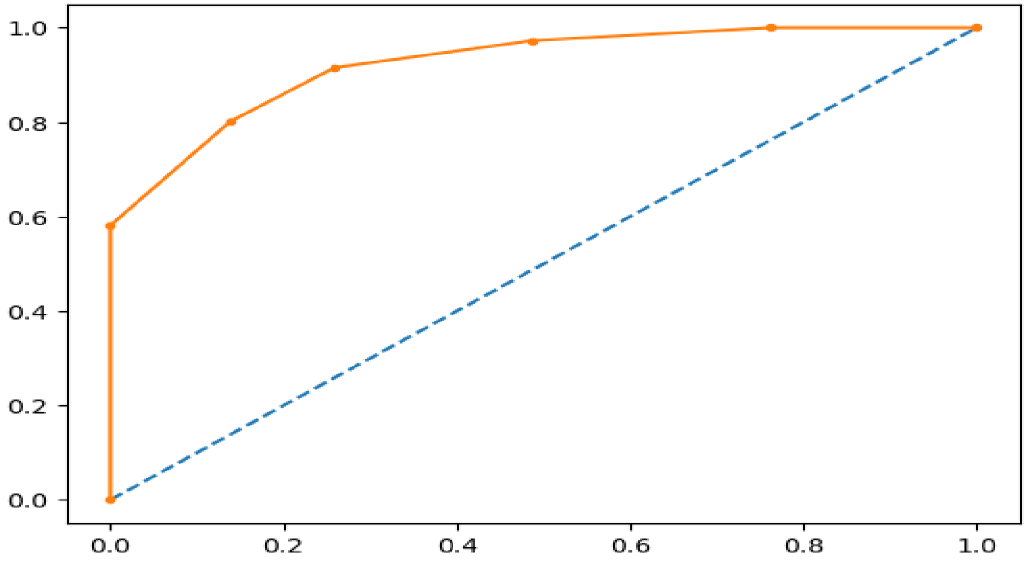
7. Evaluate the model. Calculate the mode score to get accuracy from both training and test data. For training the accuracy is 86%. and test data accuracy is 79%.

8. Get the confusion matrix and AUC-ROC curve for both the dataset.

Model Predictions

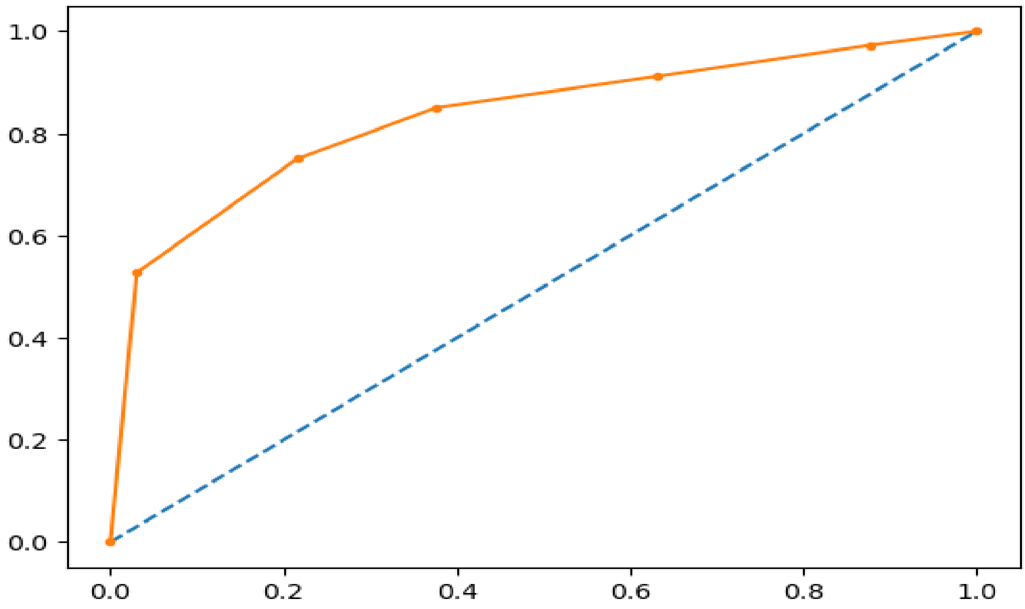
AUC-ROC Curve for train data

AUC: 0.924



AUC-ROC Curve for test data

AUC: 0.832s

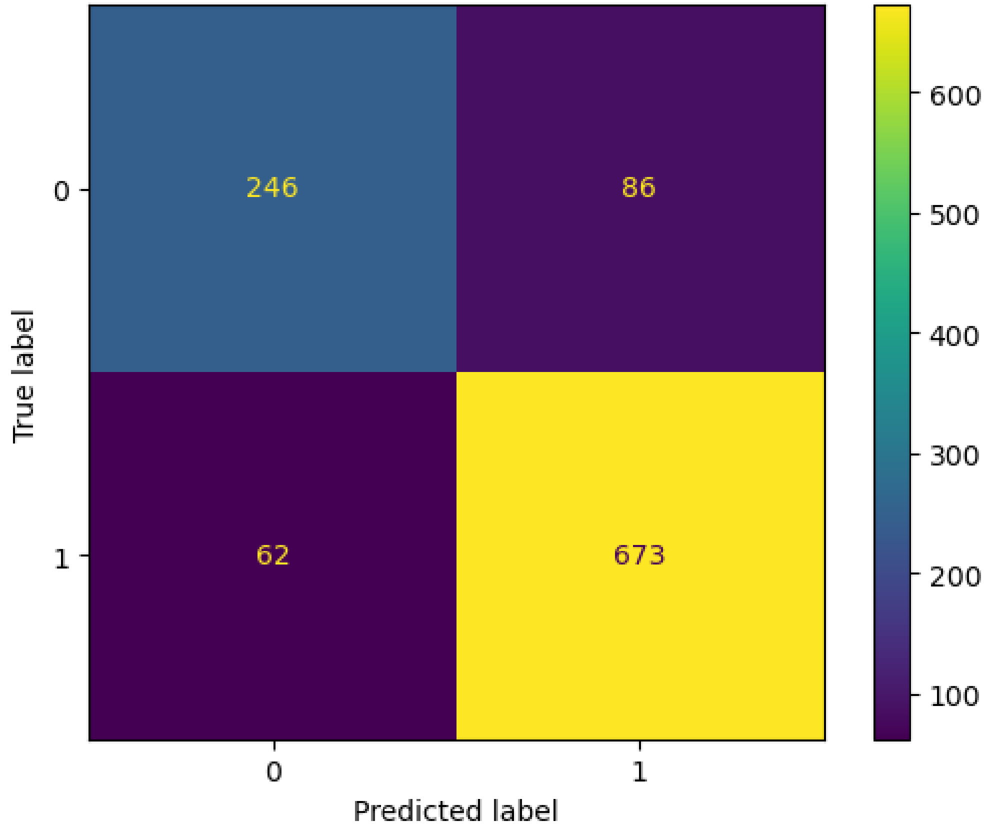


Model Evaluation:

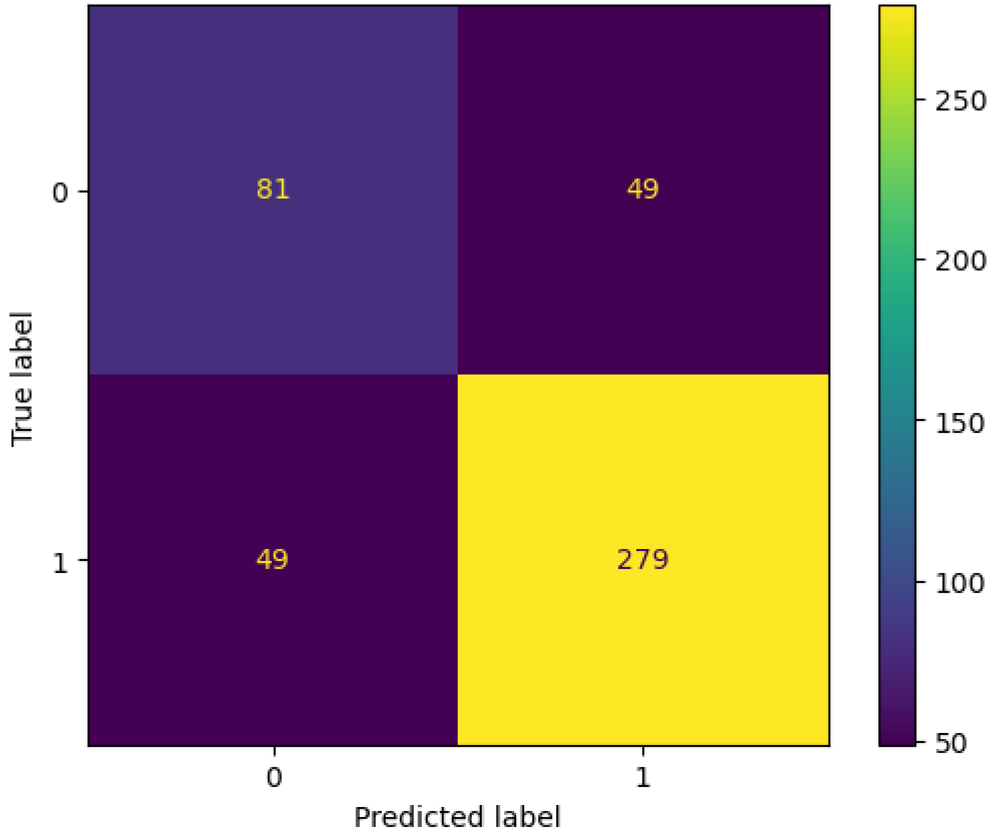
Classification report for train and test data

Train Data	Test data
precision recall f1-score support 0 0.80 0.74 0.77 332 1 0.89 0.92 0.90 735 accuracy 0.86 1067 macro avg 0.84 0.83 0.83 1067 weighted avg 0.86 0.86 0.86 1067	precision recall f1-score support 0 0.62 0.62 0.62 130 1 0.85 0.85 0.85 328 accuracy 0.79 458 macro avg 0.74 0.74 0.74 458 weighted avg 0.79 0.79 0.79 458

Confusion Matrix for Train data

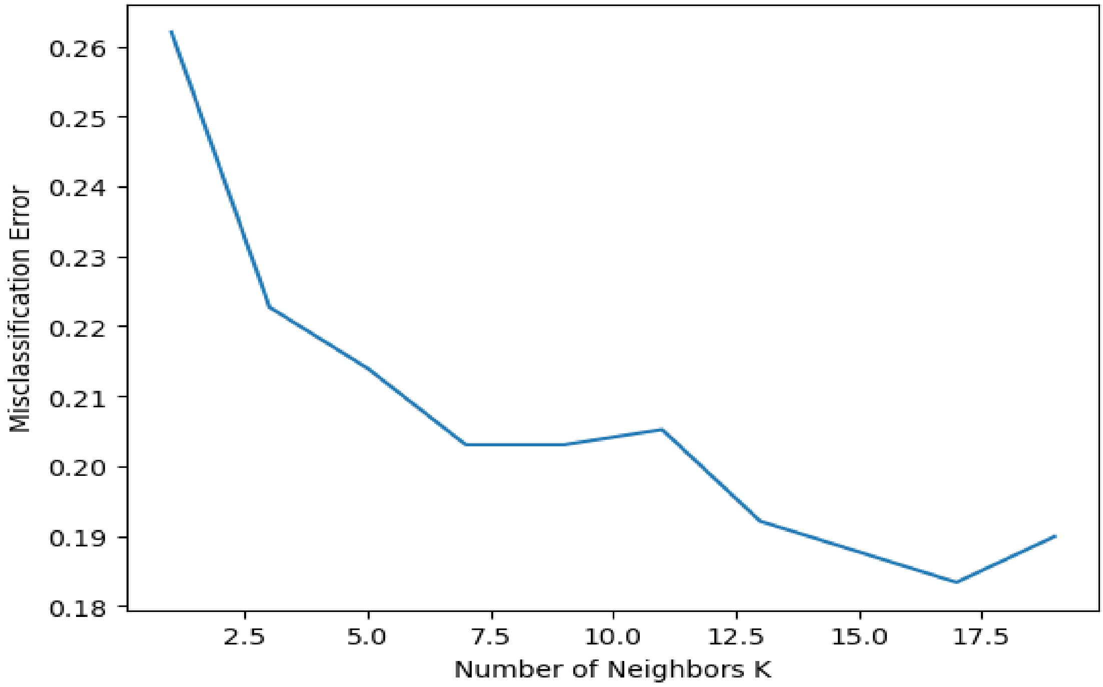


Confusion Matrix for Test data

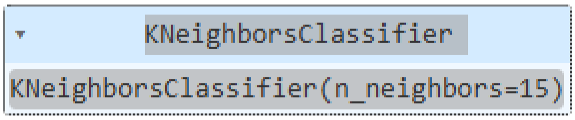


9. The difference is more between train and test data so will find next best K value.

Plot misclassification error vs k (with k value on X-axis) using matplotlib.



10. K=15 fit the KNN model.

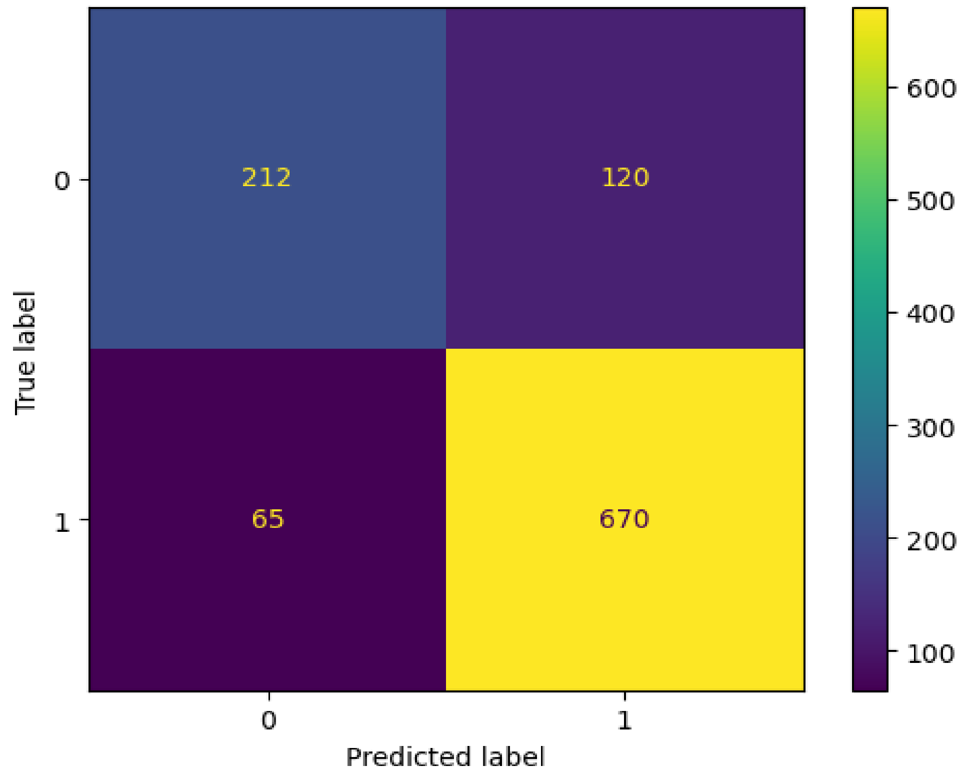

KNeighborsClassifier

```
KNeighborsClassifier(n_neighbors=15)
```

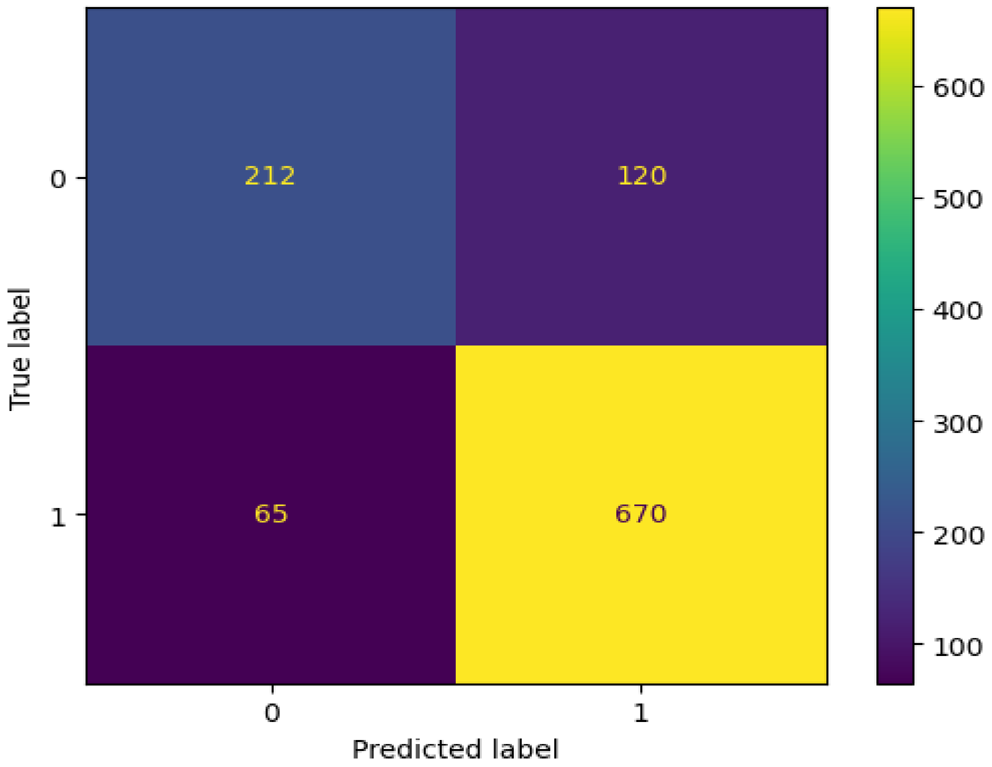
Classification report for train and test data with K=15

Train Data	Test data
precision recall f1-score support 0 0.77 0.64 0.70 332 1 0.85 0.91 0.88 735 accuracy 0.83 1067 macro avg 0.81 0.78 0.79 1067 weighted avg 0.82 0.83 0.82 1067	precision recall f1-score support 0 0.68 0.65 0.66 130 1 0.86 0.88 0.87 328 accuracy 0.81 458 macro avg 0.77 0.76 0.77 458 weighted avg 0.81 0.81 0.81 458

Confusion Matrix for Train data

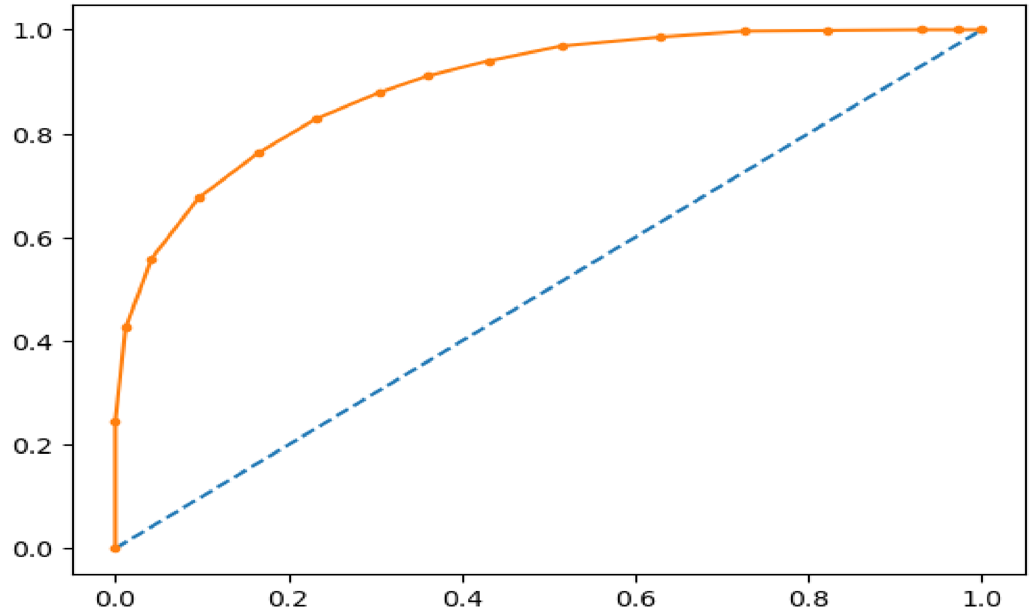


Confusion Matrix for Test data



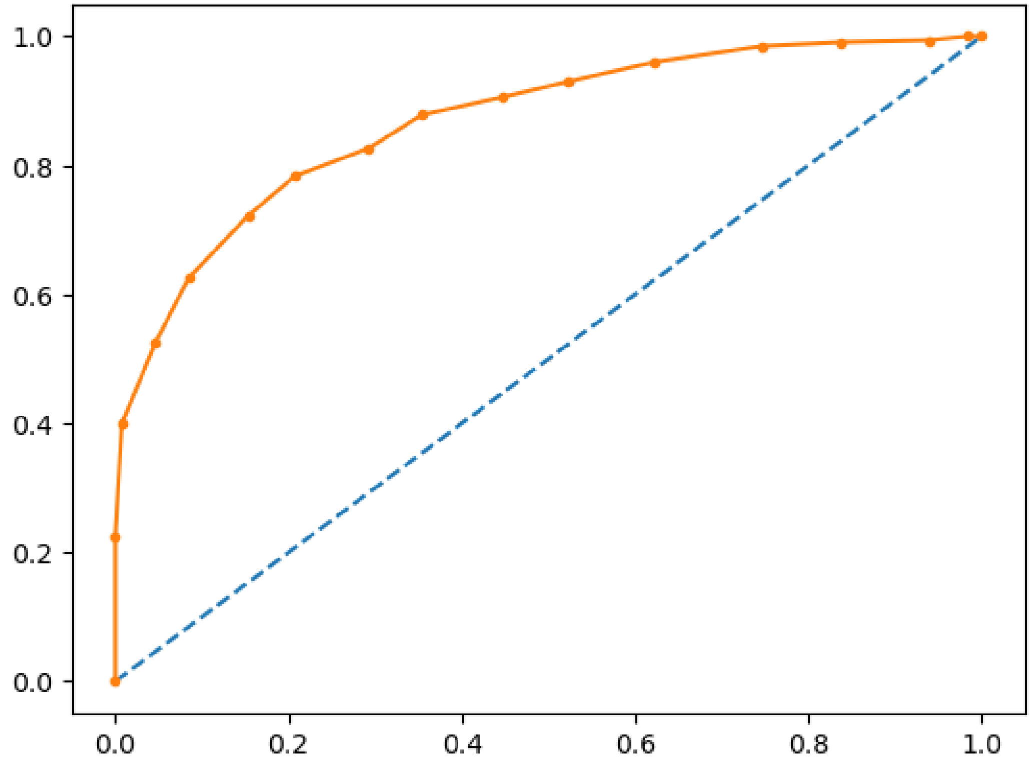
AUC-ROC Curve for train data

AUC: 0.894



AUC-ROC Curve for test data

AUC: 0.869



As the difference between train and test accuracies is less, it is a valid model. With K=15 the model is stable.

Actionable Insights:

1) The model has accuracy of 81%. Which is good fit for classification.
2) The model is having pretty high for recall around 88%, which means its able to rectify the actual true positives. The type II error is very minimal.
3) The precision is 0.86 for the class of interest. Again, the type I error is minimal.
4) The model predicts good win for Labour leader.

Overview of Bagging

Bagging is ensemble technique to improve the stability and accuracy of machine learning algorithms. Its parallel model building and evaluation. The result of one doesn’t affect the other or source dataset. It uses a technique of sampling with replacement.

Steps in Bagging:

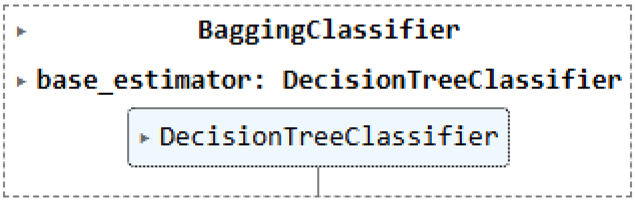
1. Model Building: Fit the Bagging model.
2. Model Evaluation: Precision, Recall, F1 score and accuracy.
3. Model Prediction: Use the model built for predicting on new dataset AUC and ROC.

Model Building:

1. Drop the target variable and save to X and pop the target variable to y.

2. Split the data in train and test. Test data as 30% and train data as 70%. Using train_test_split function.

3. Fit the Bagging model using fit function passing the Y and X train data. With parameters as base_estimator=cart,n_estimators=100,random_state=1



```
▸ BaggingClassifier
▸ base_estimator: DecisionTreeClassifier
▸ DecisionTreeClassifier
```

4. Prediction on train and test data.

5. Evaluate the model. Calculate the mode score to get accuracy from both training and test data

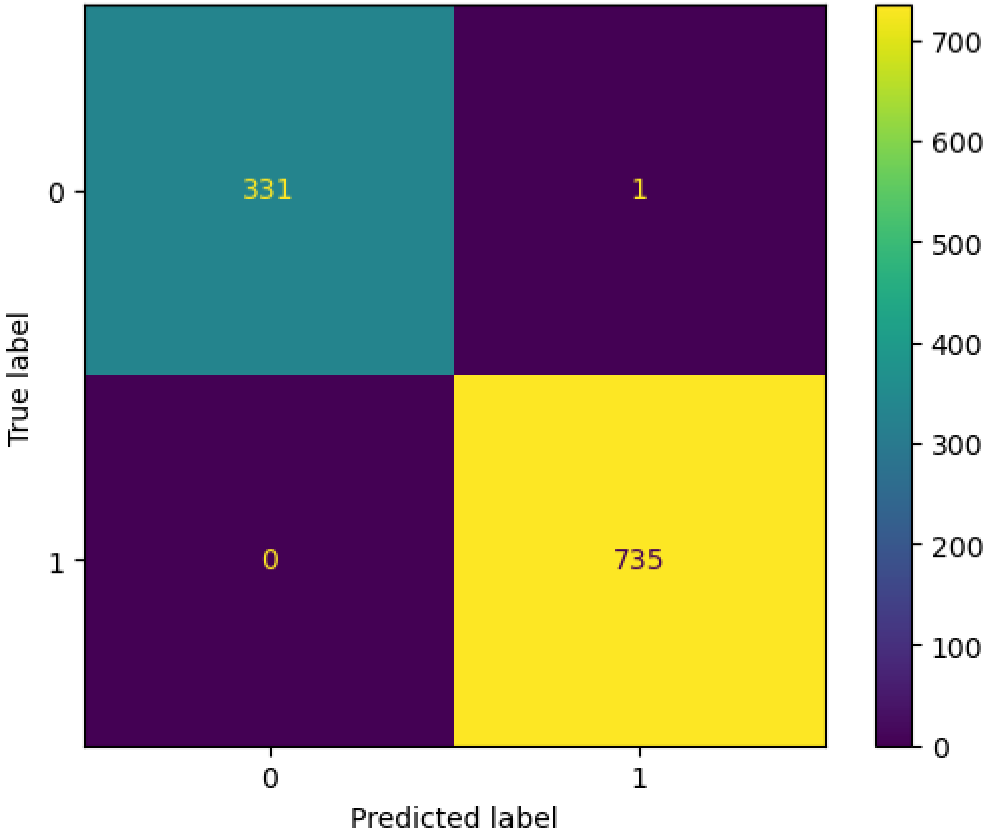
6. Get the confusion matrix and AUC-ROC curve for both the dataset.

Model Evaluation

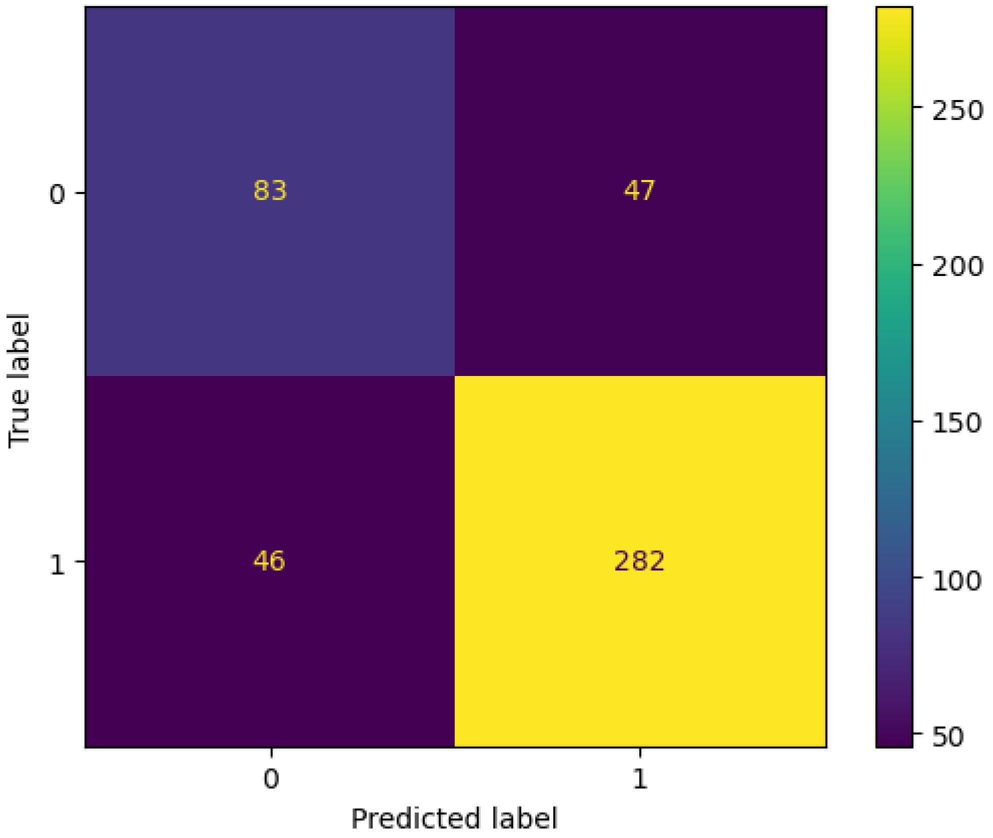
Classification report for train and test data

Train Data	Test data
precision recall f1-score support 0 1.00 1.00 1.00 332 1 1.00 1.00 1.00 735 accuracy 1.00 1067 macro avg 1.00 1.00 1.00 1067 weighted avg 1.00 1.00 1.00 1067	precision recall f1-score support 0 0.64 0.64 0.64 130 1 0.86 0.86 0.86 328 accuracy 0.80 458 macro avg 0.75 0.75 0.75 458 weighted avg 0.80 0.80 0.80 458

Confusion Matrix for Train data



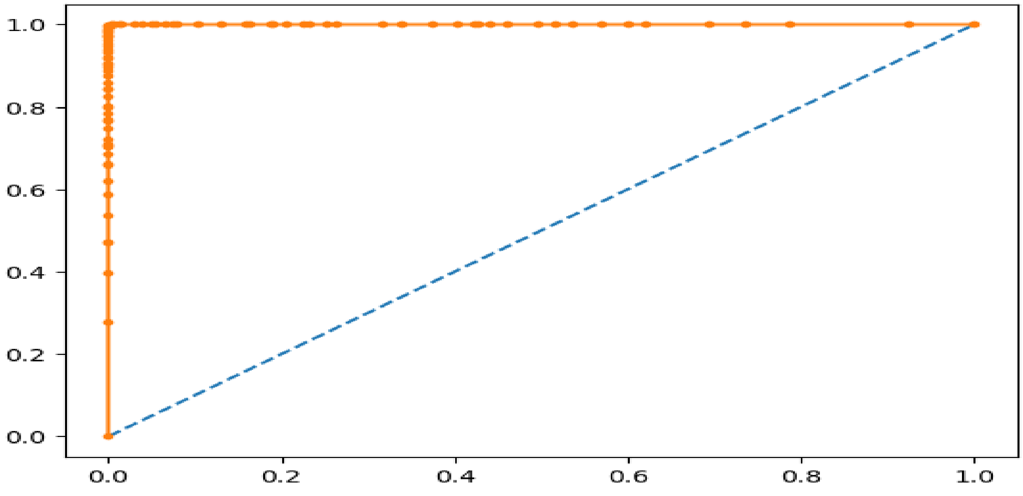
Confusion Matrix for Test data



Model Prediction

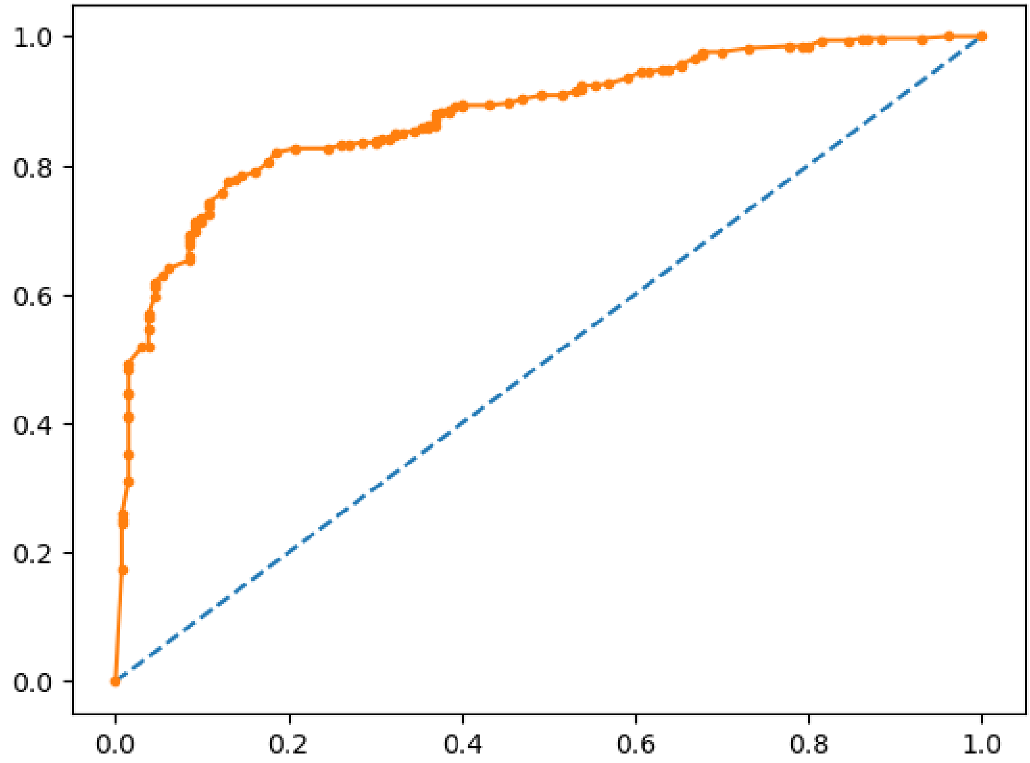
AUC-ROC Curve for train data

AUC: 1.000



AUC-ROC Curve for test data

AUC: 0.877



Actionable Insights and recommendation:

1) The model has accuracy of 80%.
2) The model is having pretty high for recall around 86%, which means its able to rectify the actual true positives. The type II error is very minimal.
3) The precision is 0.86 for the class of interest. Again, the type I error is minimal.
4) Still the model can be fine tuned since difference in train and test performance.

Overview of Boosting

Boosting is an ensemble technique. It combines the predictions of several different weak learners to produce strong result. It is a sequential process. The sample is taken from previous model to improve the errors found in previous one.

Steps in Boosting:

1. Model Building: Fit the Boosting model.
2. Model Evaluation: Precision, Recall, F1 score and accuracy.
3. Model Prediction: Use the model built for predicting on new dataset AUC and ROC.

Two Types of Boosting Model:

a. Ada Boosting: Works on weight adjustment of incorrectly classified instance. Predictions are combined through a weighted majority vote. Fit the model with AdaBoostClassifier from sklearn with parameters n_estimators ,random_state.

b. Gradient Boosting: Gradient Boosting builds models sequentially, each trying to minimize the errors. Each model is trained to predict the residuals of combined previous models. The final prediction is the sum of all previous predictions. Fit the GradientBoostingClassifier model.

Model Building:

1. Drop the target variable and save to X and pop the target variable to y.

2. Split the data in train and test. Test data as 30% and train data as 70%. Using train_test_split function.

3. Fit the Bagging model using fit function passing the Y and X train data.

4. Prediction on train and test data.

5. Evaluate the model. Calculate the mode score to get accuracy from both training and test data.

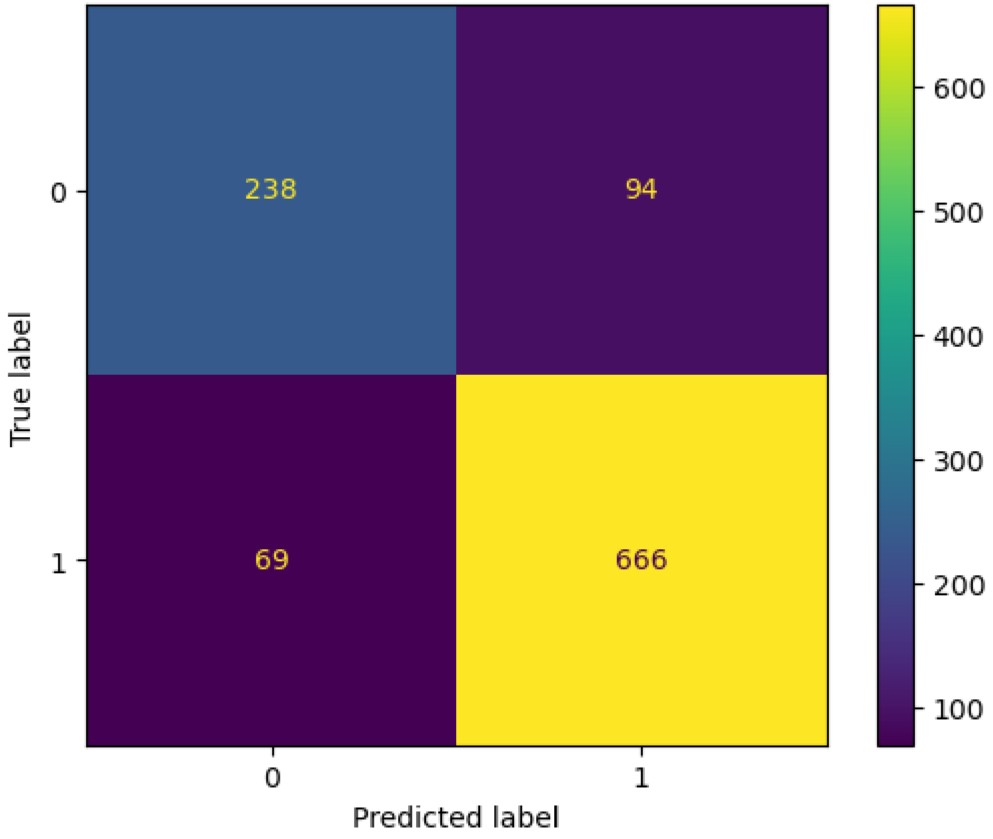
6. Get the confusion matrix and AUC-ROC curve for both the dataset

Model Evaluation for Ada Boosting

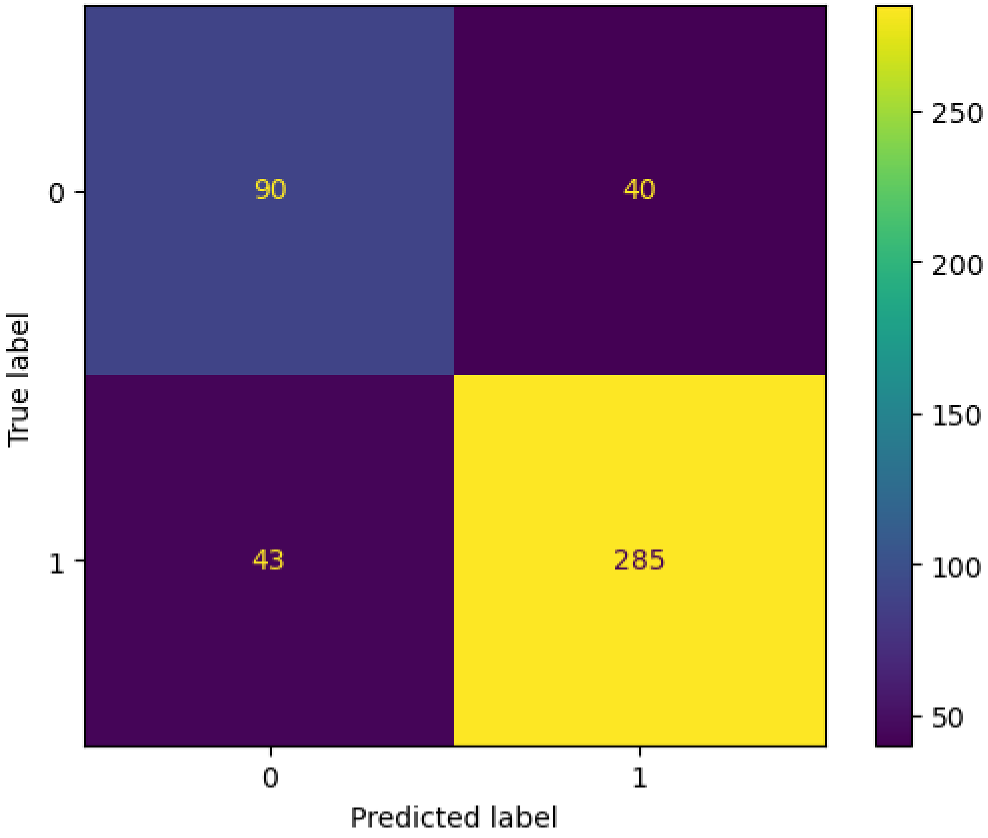
Classification report for train and test data

Train Data	Test data
precision recall f1-score support	precision recall f1-score support
0 0.78 0.72 0.74 332	0 0.68 0.69 0.68 130
1 0.88 0.91 0.89 735	1 0.88 0.87 0.87 328
accuracy 0.85 1067	accuracy 0.82 458
macro avg 0.83 0.81 0.82 1067	macro avg 0.78 0.78 0.78 458
weighted avg 0.84 0.85 0.85 1067	weighted avg 0.82 0.82 0.82 458

Confusion Matrix for Train data



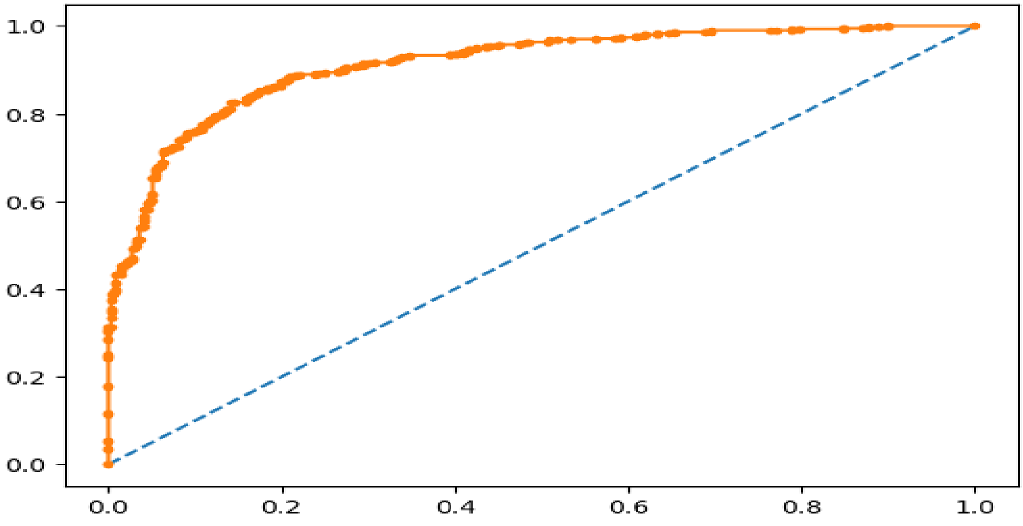
Confusion Matrix for Test data



Model Prediction

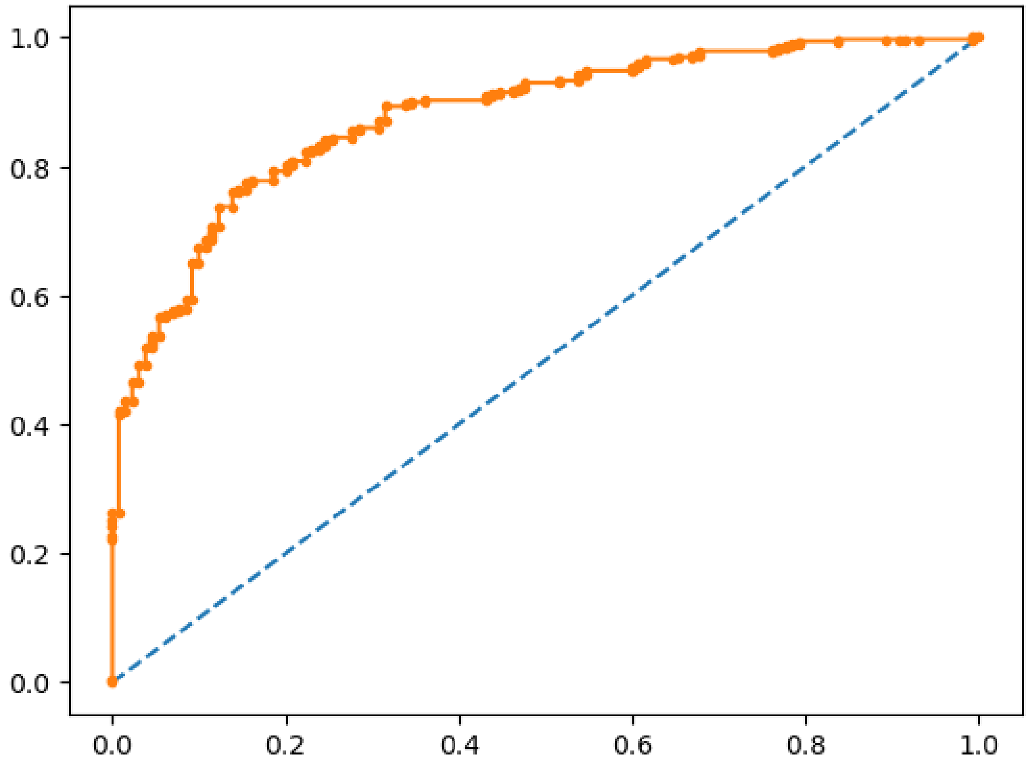
AUC-ROC Curve for train data

AUC: 0.913



AUC-ROC Curve for test data

AUC: 0.879

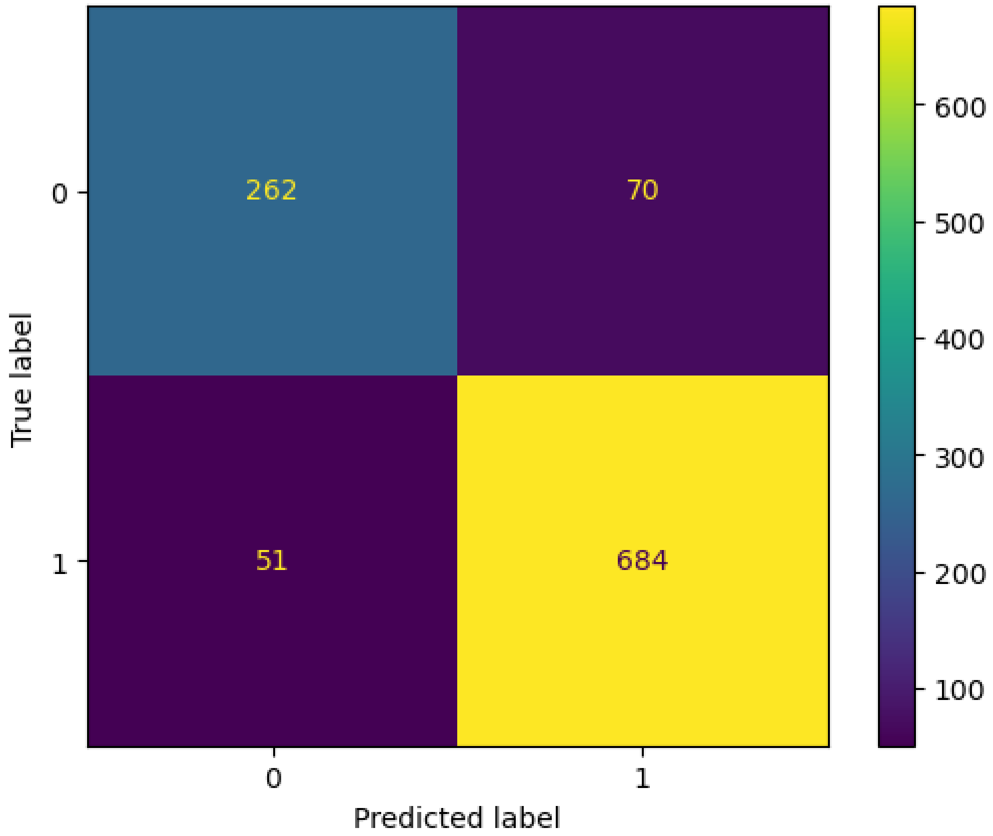


Model Evaluation for Gradient Boosting

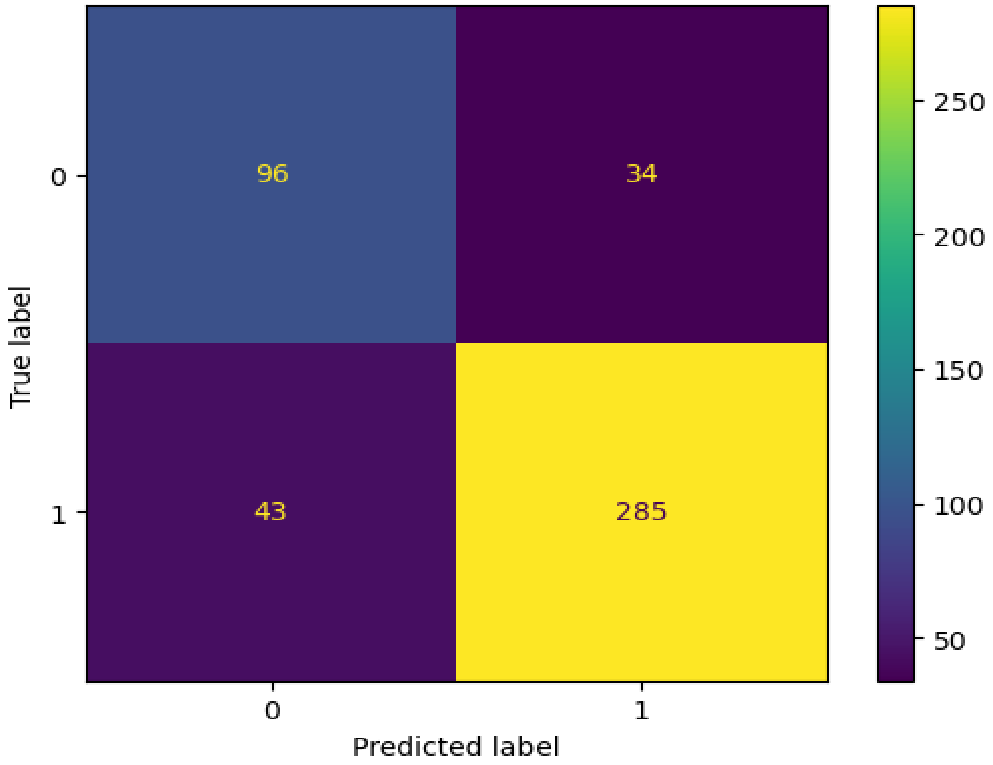
Classification report for train and test data

Train Data	Test data
precision recall f1-score support 0 0.84 0.79 0.81 332 1 0.91 0.93 0.92 735 accuracy 0.89 1067 macro avg 0.87 0.86 0.87 1067 weighted avg 0.89 0.89 0.89 1067	precision recall f1-score support 0 0.69 0.74 0.71 130 1 0.89 0.87 0.88 328 accuracy 0.83 458 macro avg 0.79 0.80 0.80 458 weighted avg 0.84 0.83 0.83 458

Confusion Matrix for Train data



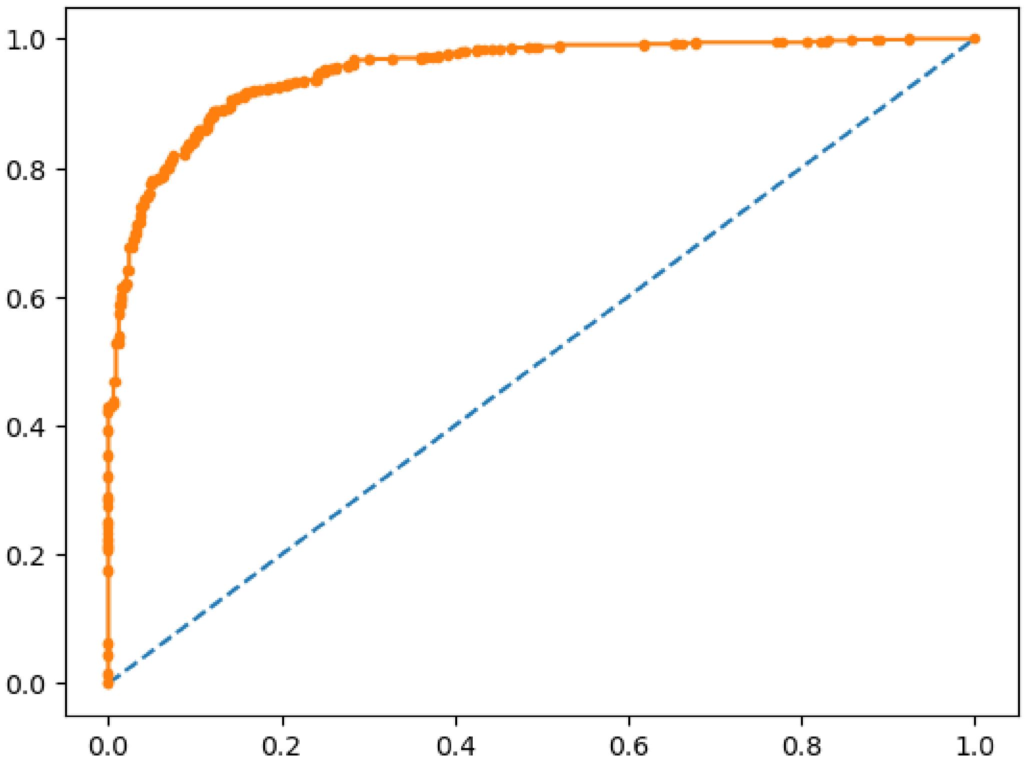
Confusion Matrix for Test data



Model Prediction

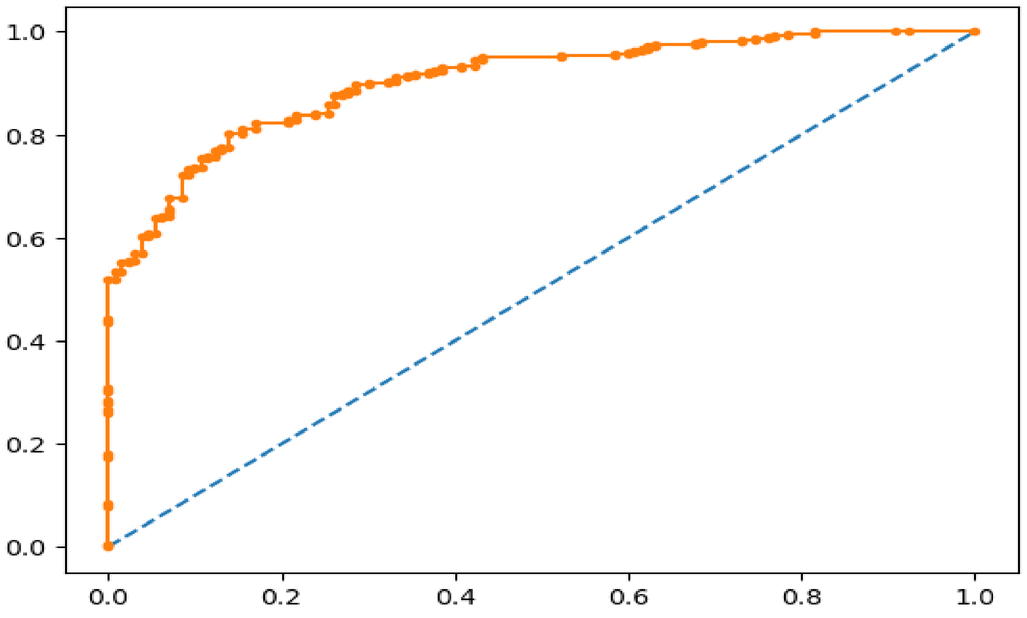
AUC-ROC Curve for train data

AUC: 0.950



AUC-ROC Curve for test data

AUC: 0.904



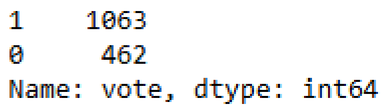
Actionable Insights and recommendation:

1) The model has accuracy of 82% and 83%. Which is good fit for classification.
2) The model is having pretty high for recall %, which means its able to rectify the actual true positives. The type II error is very minimal.
3) The precision for the class of interest is high. Again, the type I error is minimal.
4) The Gradient and Ada boosting can be give little better performance after tuning.

Smote

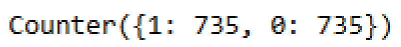
It’s a technique of handling the imbalance of the dataset. Where the minority class gets equal number of datasets like majority class, by creating synthetic samples.

Currently, in our dataset we have imbalance in 0 class.



```
1    1063
0     462
Name: vote, dtype: int64
```

- We will import Smote from imblearn.over_sampling
- Initialize the Smote method
- Fit and resample the smote function with X_train and y_train dataset and store in separate variables and use these variables on the algorithms to check the performance.
- After smote, the datasets are equal in proportion.



```
Counter({1: 735, 0: 735})
```

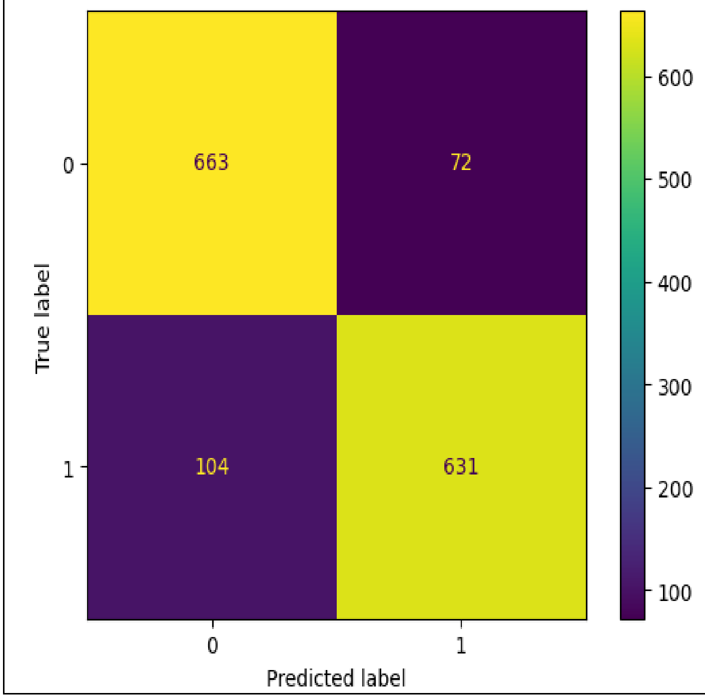
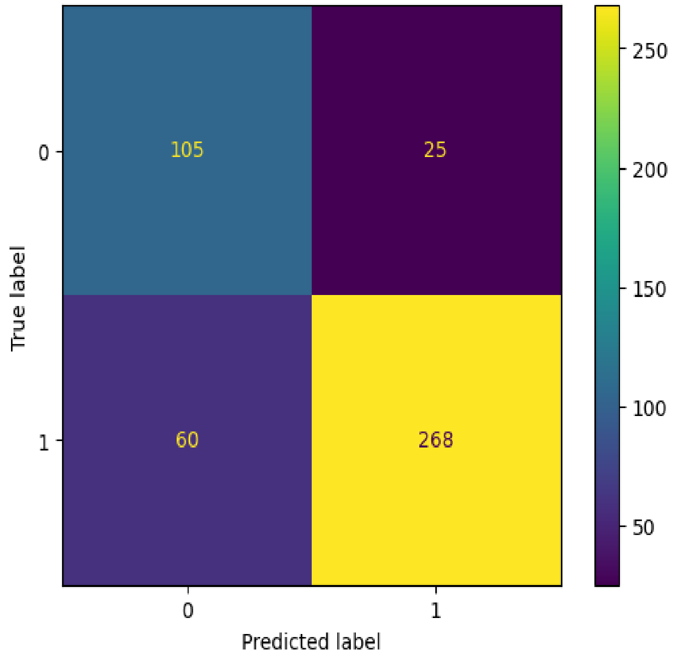
Bagging:

Adding the parameters for tuning the model.

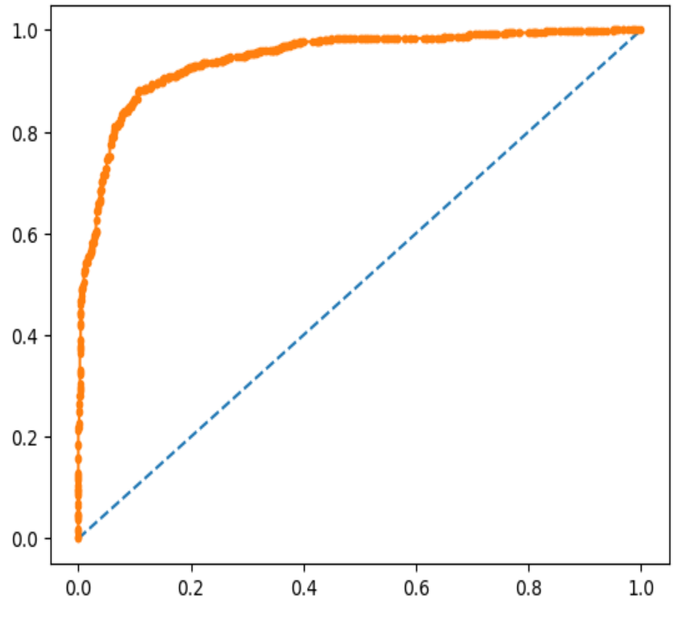
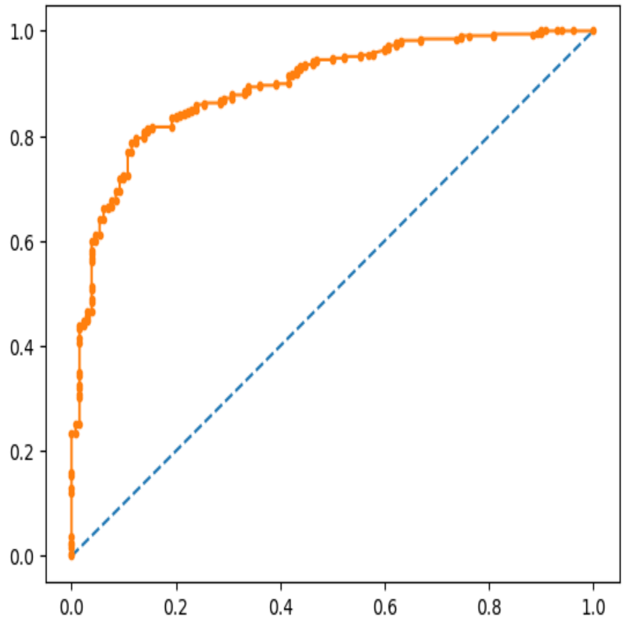
Classification report for train and test data

Train Data	Test data
precision recall f1-score support 0 0.86 0.90 0.88 735 1 0.90 0.86 0.88 735 accuracy 0.88 1470 macro avg 0.88 0.88 0.88 1470 weighted avg 0.88 0.88 0.88 1470	precision recall f1-score support 0 0.64 0.81 0.71 130 1 0.91 0.82 0.86 328 accuracy 0.81 458 macro avg 0.78 0.81 0.79 458 weighted avg 0.84 0.81 0.82 458

Confusion Matrix for train and test data

Train Data	Test data
	

AUC_ROC for Train and Test data

Train Data	Test data
AUC: 0.944 	AUC: 0.894 

Boosting

Ada Boosting classification report for train and test data.

Train Data

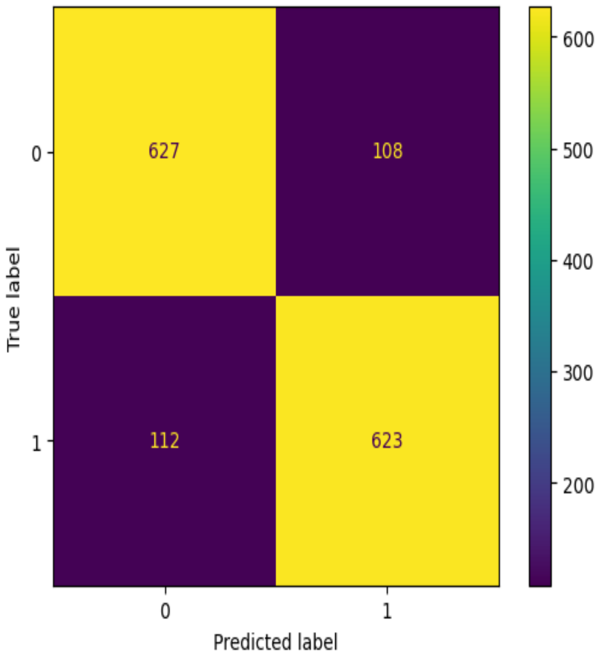
```
        precision    recall  f1-score   support

        0       0.85      0.85      0.85       735
        1       0.85      0.85      0.85       735

    accuracy                           0.85      1470
   macro avg       0.85      0.85      0.85      1470
weighted avg       0.85      0.85      0.85      1470
```

Test data

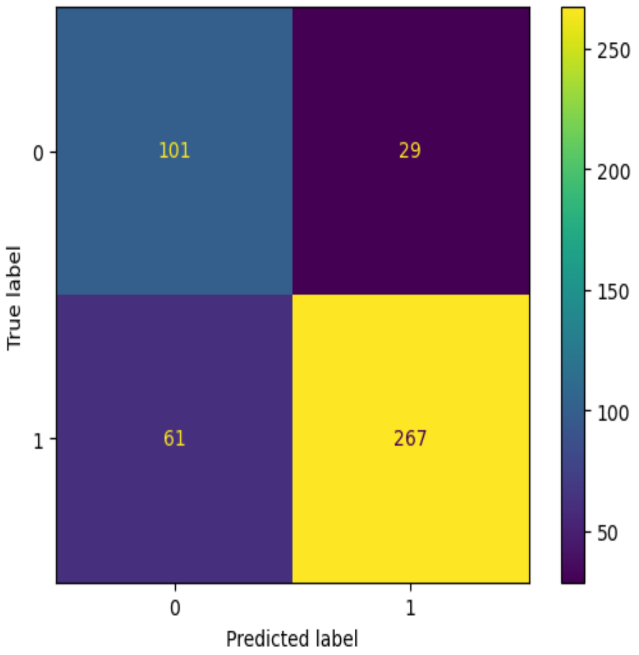
```
     precision    recall  f1-score   support

      0       0.62      0.78      0.69       130

      1       0.90      0.81      0.86       328

   accuracy                           0.80       458

   macro avg       0.76      0.80      0.77       458

weighted avg       0.82      0.80      0.81
458
```

Ada Boosting confusion Matrix for train and test data

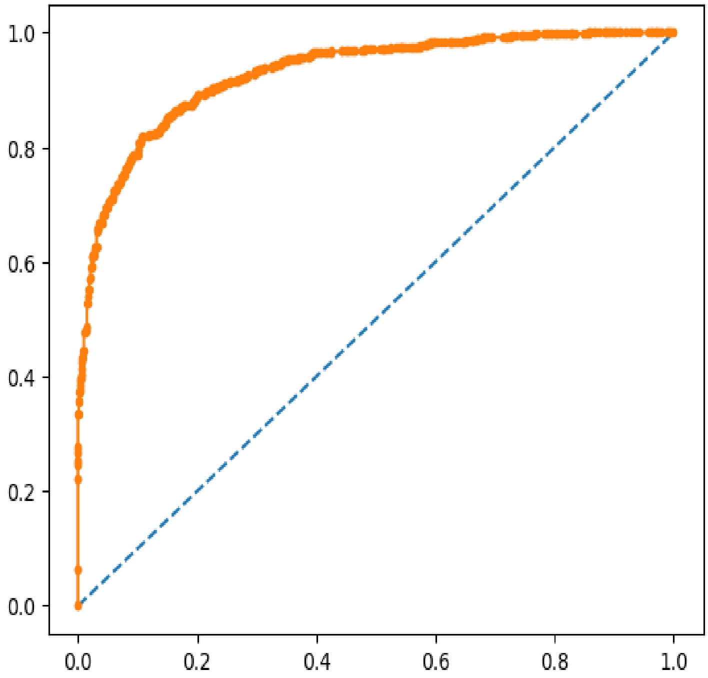
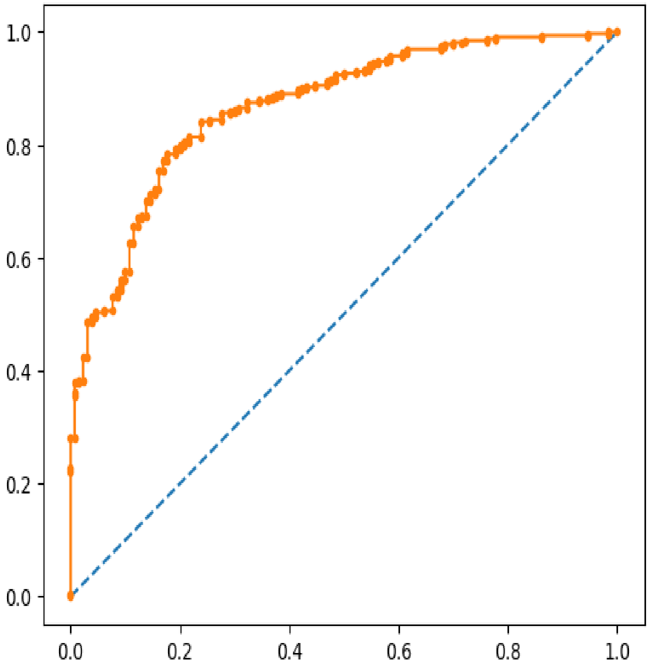
Train Data



Test data



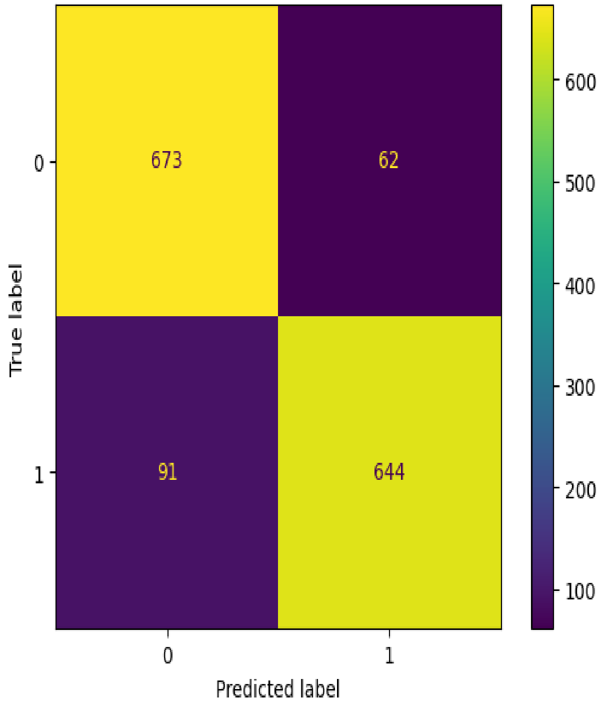
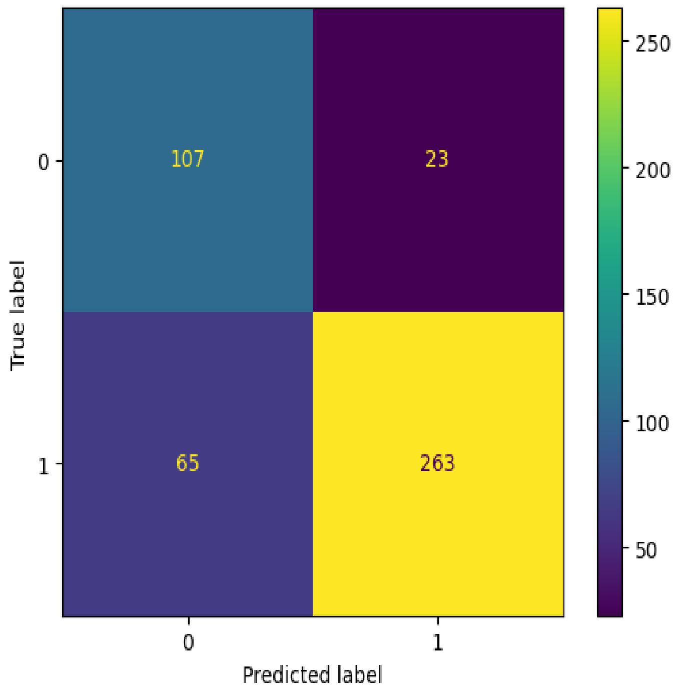
Ada Boosting AUC_ROC for Train and Test data

Train Data	Test data
AUC: 0.929	AUC: 0.868
	

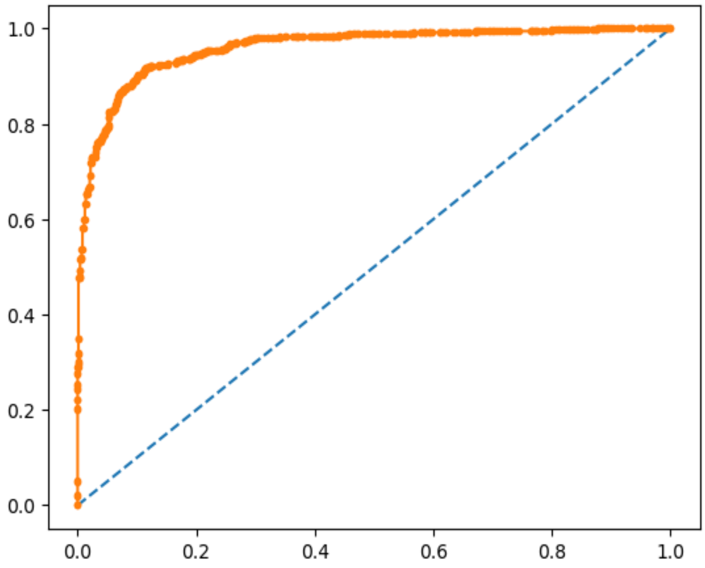
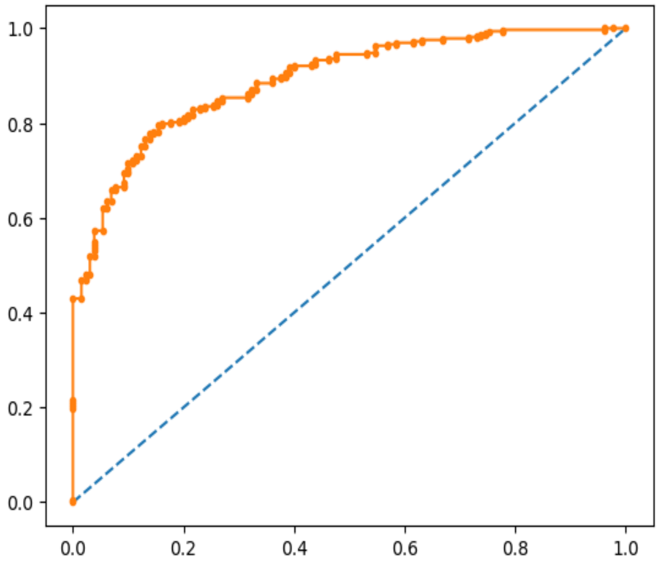
Gradient boosting: Classification report for train and test data.

Train Data	Test data
precision recall f1-score support	precision recall f1-score support
0 0.88 0.92 0.90 735	0 0.62 0.82 0.71 130
1 0.91 0.88 0.89 735	1 0.92 0.80 0.86 328
accuracy 0.90 1470	accuracy 0.81 458
macro avg 0.90 0.90 0.90 1470	macro avg 0.77 0.81 0.78 458
weighted avg 0.90 0.90 0.90 1470	weighted avg 0.84 0.81 0.81 458

Gradient boosting: Confusion Matrix for train and test data

Train Data	Test data
	

Gradient boosting: AUC_ROC for Train and Test data

Train Data	Test data
AUC: 0.960 	AUC: 0.891 

Observation:

From all the inferences above, we see that mostly all the models have similar performance. However, the Gradient boosting after Smote model is giving the best results on accuracy.

Comparison of Different Models and Final model selection

Training performance comparison:

	Naïve Bayes	KNN(k=15)	Bagging	Ada Boosting	Gradient Boosting	After Smote Bagging	After Smote Ada Boosting	After Smote Gradient Boosting
Accuracy	0.83	0.83	1	0.85	0.85	0.88	0.85	0.90
Recall	0.88	0.91	1	0.91	0.93	0.90	0.85	0.91
Precision	0.88	0.85	1	0.88	0.88	0.90	0.85	0.91
F1	0.88	0.88	1	0.89	0.89	1	0.85	0.89

Testing performance comparison:

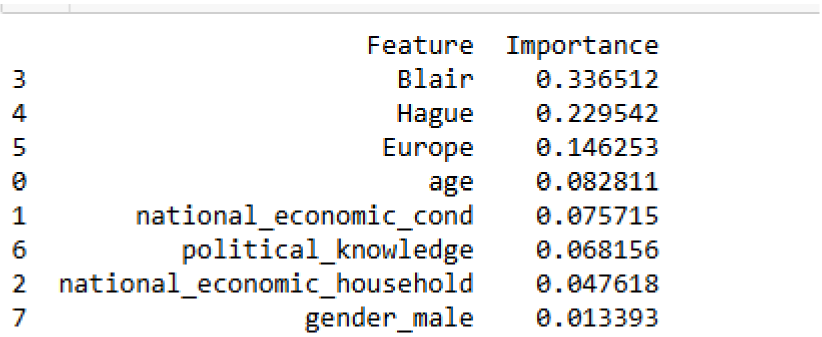
	Naïve Bayes	KNN(k=15)	Bagging	Ada Boosting	Gradient Boosting	After Smote Bagging	After Smote Ada Boosting	After Smote Gradient Boosting
Accuracy	0.83	0.81	0.80	0.82	0.85	0.81	0.80	0.81
Recall	0.87	0.88	0.86	0.91	0.87	0.82	0.90	0.92
Precision	0.89	0.86	0.86	0.88	0.88	0.91	0.90	0.92
F1	0.88	0.87	0.86	0.88	0.87	0.86	0.86	0.86

The Gradient boosting after Smote model is giving the best results on accuracy.

Best model selection:

With this, it is also very clear that the Gradient Boosting model has performed above all the rest of the models. With an Accuracy value of 90%, it is predicting the highest percentage of both our classes of interest. If we still look at the Recall value, the model is able to identify 91% of the true positives correctly. Similarly, we see that the Area Under the Curve (AUC) captured is 96% for train data and 89% for the test data. It is not the best however; it still supersedes all the other models. Therefore, it is safe to say that this model can be used for making predictions on any unseen data that is fed to the model.

Important feature selection on final model.



```
                     Feature  Importance
3                      Blair    0.336512
4                      Hague    0.229542
5                     Europe    0.146253
0                        age    0.082811
1      national_economic_cond    0.075715
6        political_knowledge    0.068156
2  national_economic_household    0.047618
7                gender_male    0.013393
```

Conclude with the key takeaways for the business

1) Age Influence: After the analysis on age factor, age ranging from youth to old they tend to follow the pattern of voting to particular party. From the model it shows Labour leader shows majority.

2) Economic Perception:

a) National Economic Conditions: The economic condition like GDP, Inflation rate etc how these are affected by choosing the particular part based on that we could see voting behaviour.

b) Household Economic Conditions: Insights into how people perceive their household economic conditions can affect consumer behaviour and spending patterns. How Good the middle class or below poverty people can survive with the fluctuation is countries economy.

3) Voting Trends: Distribution of Votes: By analysis the model predict that the Conservative is less preferred over Labour by the majority. By this we could understand the public sentiment over the government.

4)Election Candidate Ratings

Blair is getting more vote from labour leader and the way they have rated is 4. So the service to public is good and this leadership can be trusted by public.

Hague is getting more vote in labour leader and the way they have rated is 2. So, the service to public is not so good. Only 300 people think he is fine. But 500+ people believe he is not good fit.

5) Gender Distribution:

Voting by Gender: Female is more compared to male. The voting trend goes more in Labour by female. The party can target more female for campaigns and products which focuses more on female.

6)Political Knowledge Levels: The average level of political knowledge among citizens can affect how informed the public is about policy changes, and other decision changes which in turn influences business regulations and compliance.

7)Opinions on Europe: Europe Opinion on how the integration into their culture, business will have huge impact on business decisions related to international trade and market expansion. The people are diverse and few of them are sceptic about this integration and few are okay with trend.

With the model choose we could see that the model is able to capture 89% of the data correctly.

Executive Summary

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:

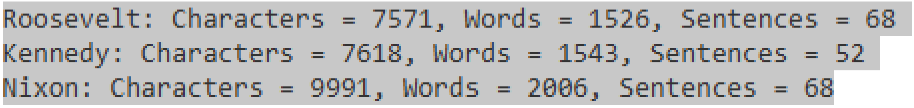
1. President Franklin D. Roosevelt in 1941
2. President John F. Kennedy in 1961
3. President Richard Nixon in 1973

Problem 2 - Define the problem and Perform Exploratory Data Analysis

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

Steps:

1. Import all the required package. Like inaugural, stopwords, stemmer etc.
2. Find the length of text using len function to find the number of characters in the speech.
3. To find number of words use the package from nltp word_tokenize and apply len function to the same.
4. To find number of words use the package from nltp sent_tokenize and apply len function to the same.



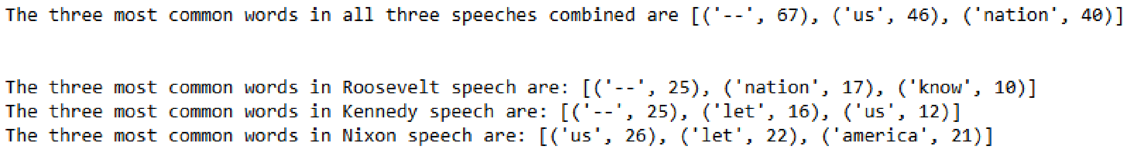
```
Roosevelt: Characters = 7571, Words = 1526, Sentences = 68
Kennedy: Characters = 7618, Words = 1543, Sentences = 52
Nixon: Characters = 9991, Words = 2006, Sentences = 68
```

Problem 2 - Text cleaning

Stopword removal - Stemming - find the 3 most common words used in all three speeches

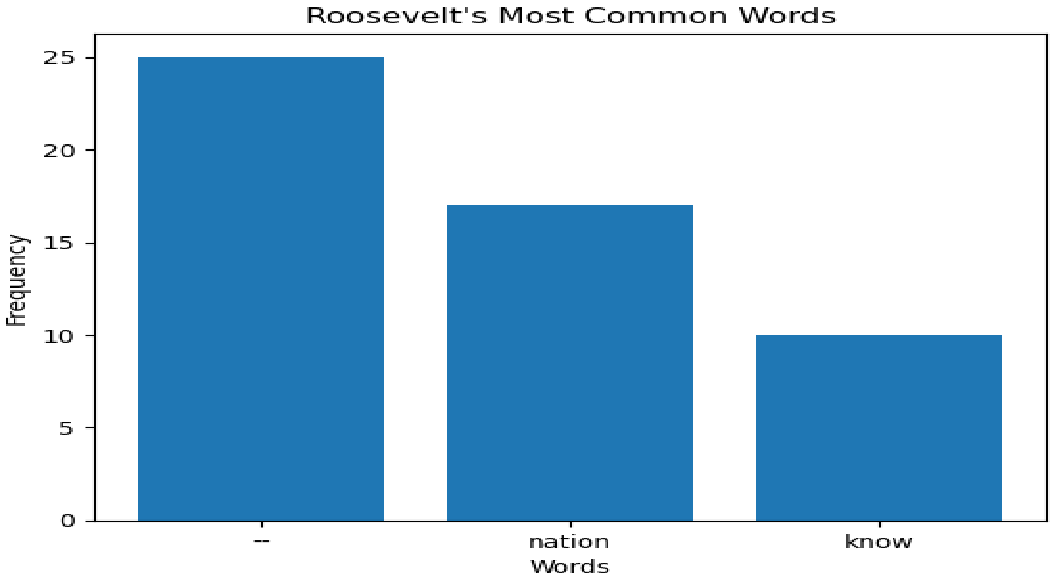
Steps:

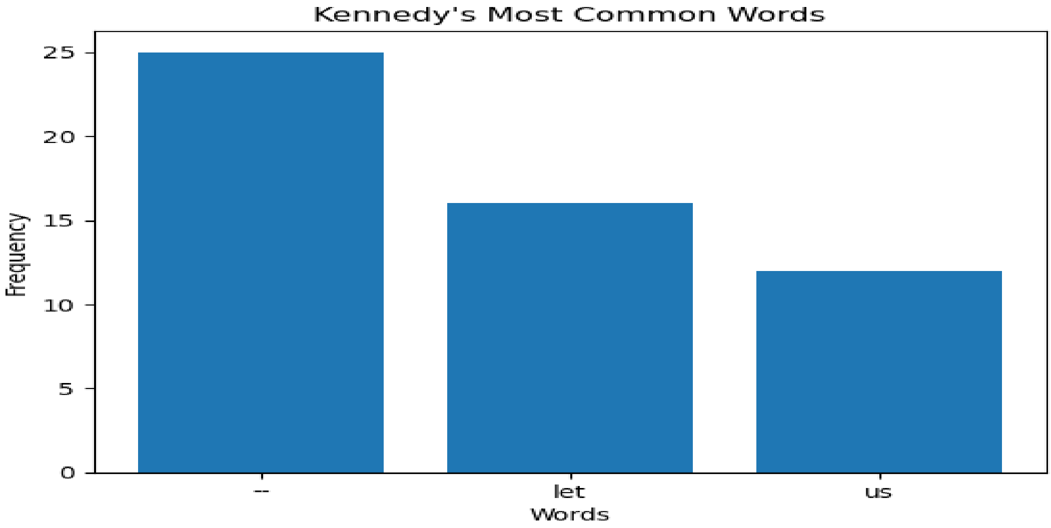
1. Remove all the stopwords from English and list of punctuations.
2. Perform the stemming operations. Like converting wedding to wed.
3. Convert all the text to lower.
4. Find the FreqDist of all the filtered words.
5. Use the most_common function to get the commonly used words.

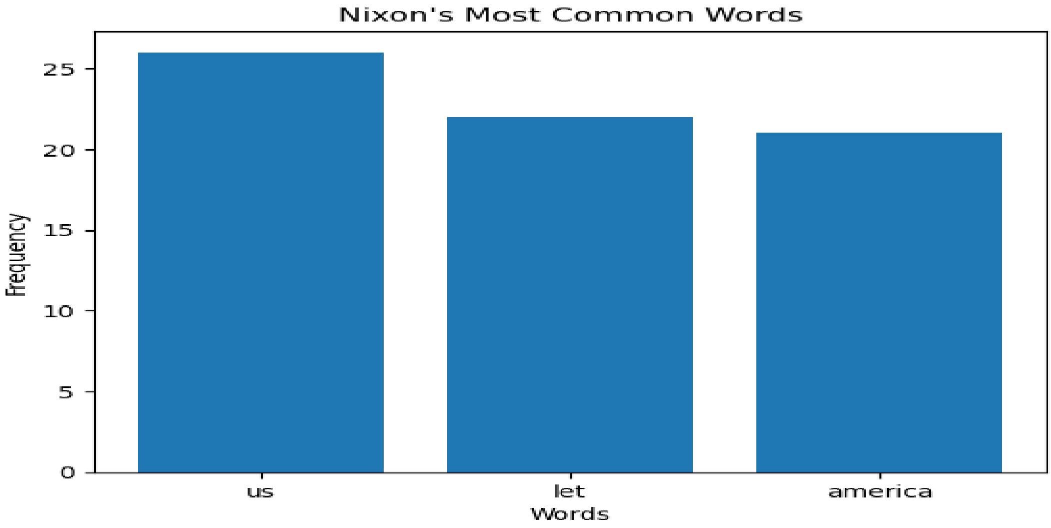


```
The three most common words in all three speeches combined are [('--', 67), ('us', 46), ('nation', 40)]


The three most common words in Roosevelt speech are: [('--', 25), ('nation', 17), ('know', 10)]
The three most common words in Kennedy speech are: [('--', 25), ('let', 16), ('us', 12)]
The three most common words in Nixon speech are: [('us', 26), ('let', 22), ('america', 21)]
```

Plot word frequencies





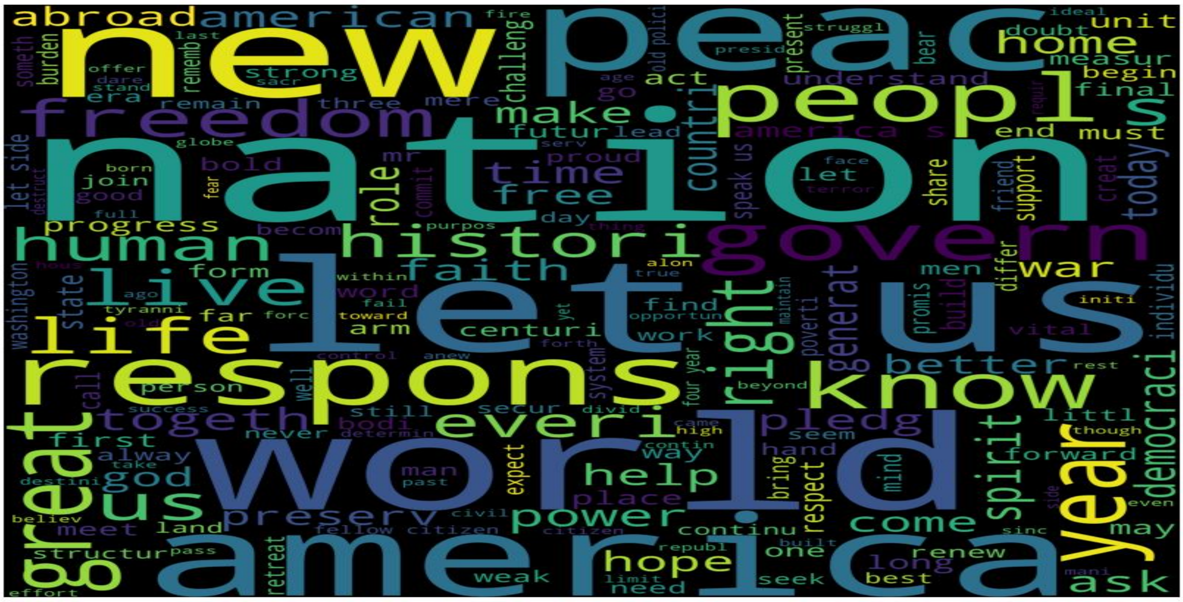


Problem 2 - Plot Word cloud of all three speeches

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

Generated and displayed word clouds for each of the three specified inaugural speeches.

Word Cloud for speeches (after cleaning)!!


nation
let us
world
america
new
peac
respons
peopl
freedom
govern
histori
human
know
great
life
live
right
year
everi
togeth
pledg
spirit
democraci
better
help
power
come
hope
go
us
god
faith
make
time
free
countri
role
today
support
share
home
unit
abroad
american
strong
war
ask
may
must
end
final
begin
measur
doubt
understand
act
challeng
futur
lead
proud
bold
progress
join
good
first
alway
preserv
land
meet
structur
weak
place
way
centuri
find
work
arm
toward
far
state
washington
generat
promis
build
vital
individu
initi
men
differ
friend
creat
speak us
let
face
terror
fire
ideal
struggl
bear
present
presid
old polici
age
go
remain
three
mere
era
offer
dare
stand
sacr
someth
burden
rememb
last
globe
born
mr
serv
commit
purpos
becom
day
form
within
word
fail
forc
ago
tyranni
control
anew
forth
system
well
call
person
success
still
secur
divid
never
determin
high
contin
came
seem
hand
forward
littl
though
destini
take
believ
man
past
civil
fellow
citizen
expect
bring
respect
mind
even
sinc
side
republ
built
one
renew
long
seek
best
limit
need
retreat
effort
pass
mani
rest
beyond
four year
yet
opportun
poverti
maintain
alon
true
thing
let side
destruct
full
fear