Project Solution : Machine Learning

Submitted by:

Name : V R S Anurag

Email ID : vrsanurag@gmail.com

Date : 15th May,2021

Problem 1 :

You are hired by one of the leading news channels CNBE who wants to analyze 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:

S.No	Variable Name	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.

1.1) Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it.

First, we will be loading the given dataset into the data frame and name it as df1 and will check the top 5 and bottom 5 observations.

- Top 5 Observations:

	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

- Bottom 5 Observations:

	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

From the above data set you can see that the first column 'Unnamed:0' consists of only index numbers, hence that column is not much of use for our modelling. Hence, we will be dropping the column.

- Checking the Data types of the Data Set:

```
<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
```

- After dropping the first column, the data set consist of **9 variables** and **1,525 observations.**
- From the above data set we can that only age is of continuous variable and others are categorical variable but are in integer form in the given data set which we need to convert to categorical.
- From the above data set we can say that **'vote'** is the Target variable with two categories and balance 8 variables are the independent variable or predictor variable which will be used for model building.
- From the given dictionary we can say that there are two parties i.e., Conservative & Labour.

Since the following variables (economic.cond.national, economic.cond.household, Blair, Hague, Europe & political.knowledge) are of integer type but actually they are categorical, hence we will be converting them to categorical variables. After converting to categorical we will be checking the info of the data set again.

❖ Checking the Data types of the Data Set:

```
<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   object
 3   economic.cond.household  1525 non-null   object
 4   Blair                    1525 non-null   object
 5   Hague                    1525 non-null   object
 6   Europe                   1525 non-null   object
 7   political.knowledge      1525 non-null   object
 8   gender                   1525 non-null   object
dtypes: int64(1), object(8)
memory usage: 107.4+ KB
```

❖ Checking the summary of the dataset for Continuous Variable:

age	
count	1525
mean	54.1823
std	15.71121
min	24
25%	41
50%	53
75%	67
max	93

From the above summary we can say the following

- There are about **1,525 voters**
- Minimum age of the voter is 24 and Maximum age is 93
- Mean age of the voter is around 54

❖ Checking the summary of the dataset for Categorical Variable:

	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
count	1525	1525	1525	1525	1525	1525	1525
unique	5	5	5	5	11	4	2
top	3	3	4	2	11	2	female
freq	607	648	836	624	338	782	812

From the above summary we can say the following,

- In National Economic Conditions there are '5' unique values ranging from 1 to 5 where 1 is worst and 5 is best and '3' is frequently repeated that is around 607 times.
- In Household Economic Conditions there are '5' unique values ranging from 1 to 5 where 1 is worst and 5 is best and '3' is frequently repeated that is around 648 times.
- In Blair ('Assessment of Labour Leader') there are '5' unique values ranging from 1 to 5 where 1 is worst and 5 is best and '4' is frequently repeated that is around 836 times.
- In Hague ('Assessment of Conservative Leader') there are '5' unique values ranging from 1 to 5 where 1 is worst and 5 is best & '2' is frequently repeated that is around 624 times.
- In Europe ('A 11-point scale) there are 11 scale values ranging from 1 to 11 and '11' scale is repeated most frequently that is 338 times.
- In Political Knowledge scale ranging from 0 to 3 and '2' is repeated most frequently i.e., around 782 times.
- They are two unique data in gender i.e., Male & Female where female voters are higher i.e., around 812.

❖ Checking for any missing values:

Total	Percent	
gender	0	0
political.knowledge	0	0
Europe	0	0
Hague	0	0
Blair	0	0
economic.cond.household	0	0
economic.cond.national	0	0
age	0	0
vote	0	0

From the table we can say that there are no null values present.

❖ Checking for any duplicate values:

- Number of duplicate rows = 8

We can see that there are **8 duplicate rows**, since the number is less, we can drop the duplicate rows. After dropping the duplicate rows there will be **1,517 observations**.

❖ Checking Unique Values for Categorical Variables

```
VOTE :  2
Conservative     460
Labour          1057
Name: vote, dtype: int64
```

```
ECONOMIC.COND.NATIONAL
:  5
1     37
5     82
2    256
4    538
3    604
Name: economic.cond.na
tional, dtype: int64
```

```
ECONOMIC.COND.HOUSEHOL
D :  5
1     65
5     92
2    280
4    435
3    645
Name: economic.cond.ho
usehold, dtype: int64
```

```
BLAIR :  5
3      1
1     97
5    152
2    434
4    833
Name: Blair, dtype
: int64
```

```
HAGUE :  5
3     37
5     73
1    233
4    557
2    617
Name: Hague, dtype:
int64
```

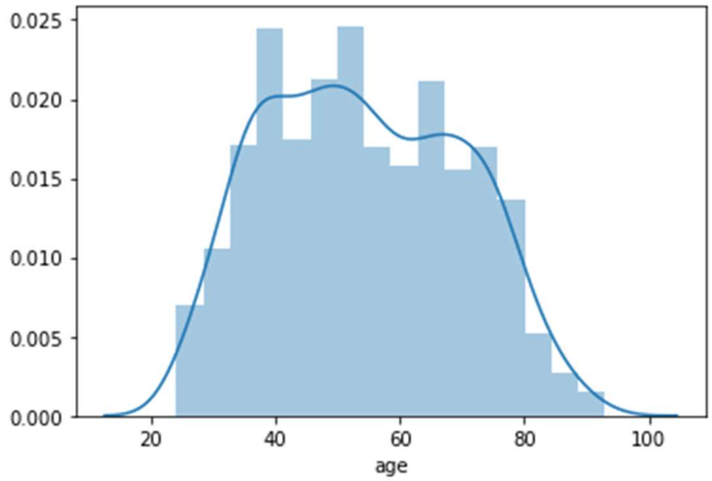
```
EUROPE :  11
2      77
7      86
10    101
1     109
9     111
8     111
5     123
4     126
3     128
6     207
11    338
Name: Europe, dtype:
int64
```

```
POLITICAL.KNOWLEDGE :
4
1     38
3    249
0    454
2    776
Name: political.knowl
edge, dtype: int64
```

```
GENDER :  2
male      709
female    808
Name: gender, dtype: int64
```

1.2) Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers?

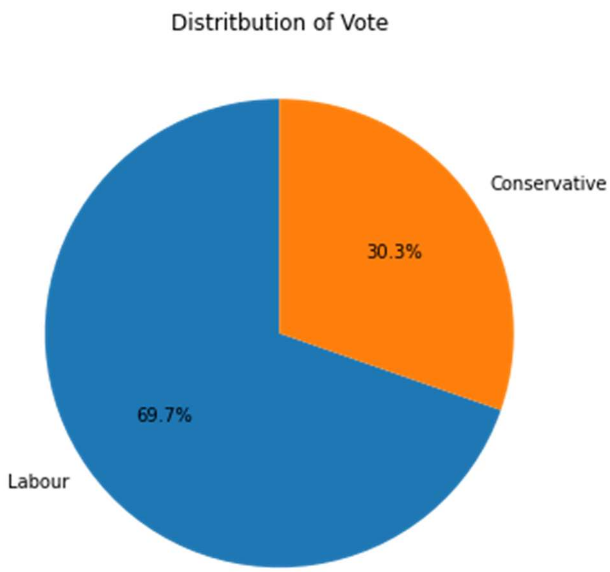
❖ Univariate Analysis : Distplot



From the distribution plot we can say the following,

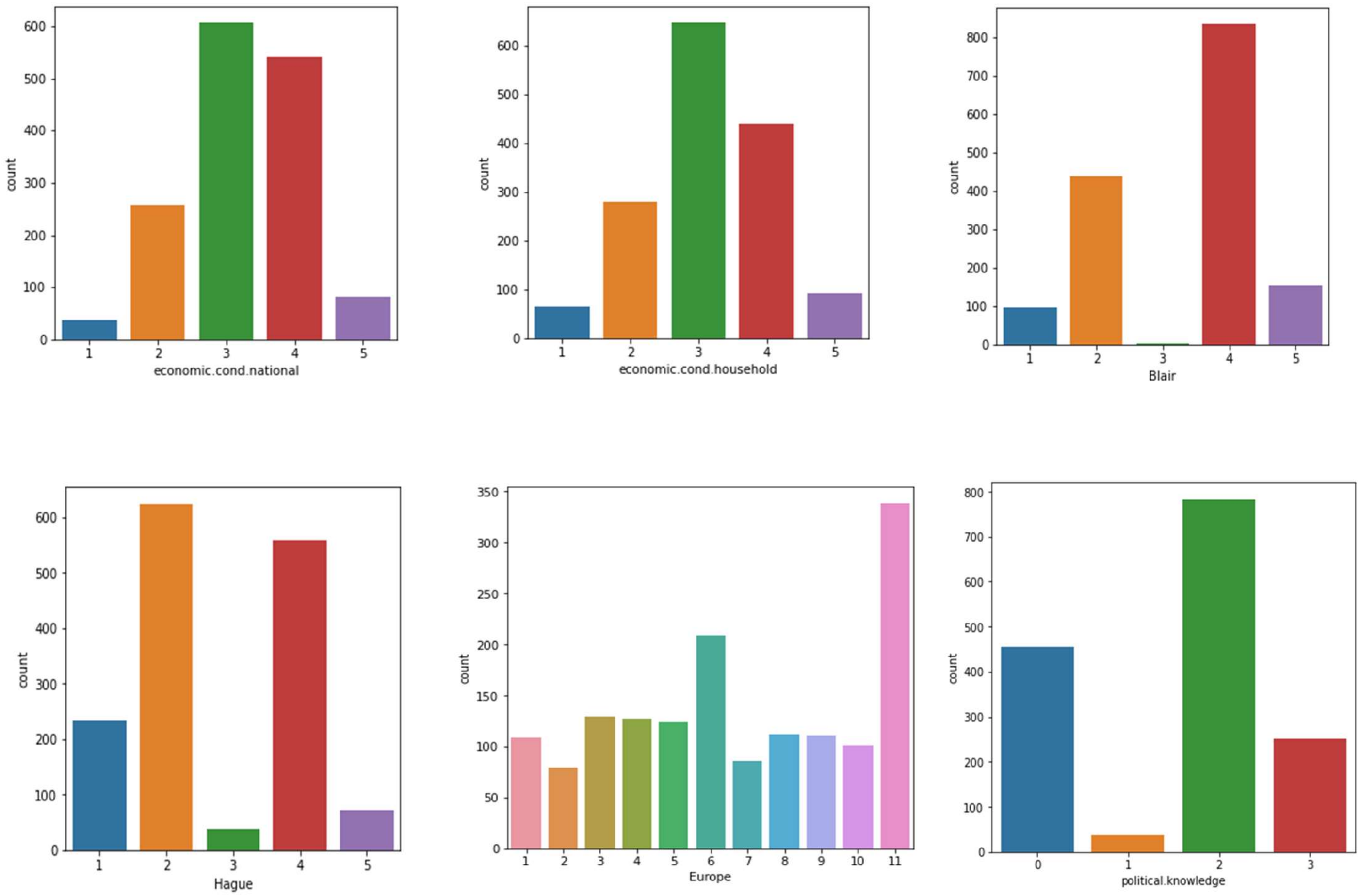
- Age is Normally Distributed with Age ranging from 25 to 90.
- Age Between 40-70 contributed highest number of voters.

❖ Distribution of Vote : Using Pie-Plot



From the pie-plot we can say that Labour Party has 69.7% of votes and Conservative has 30.3% of Votes

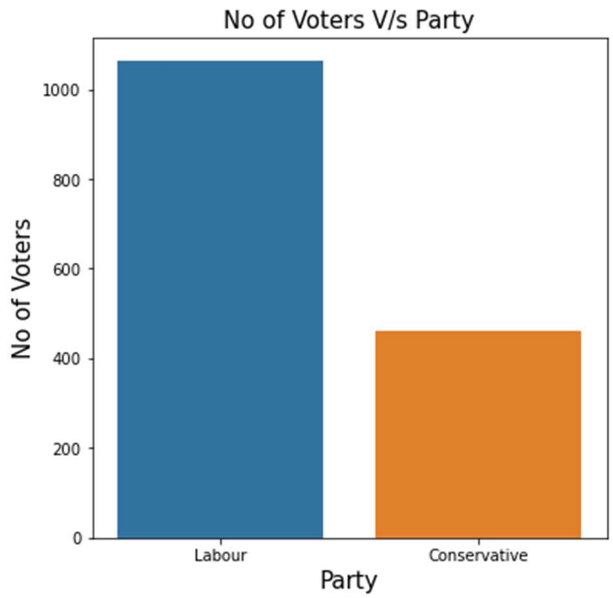
❖ Count Plot for Various Categorical Variables:



From the above plots we can see the following

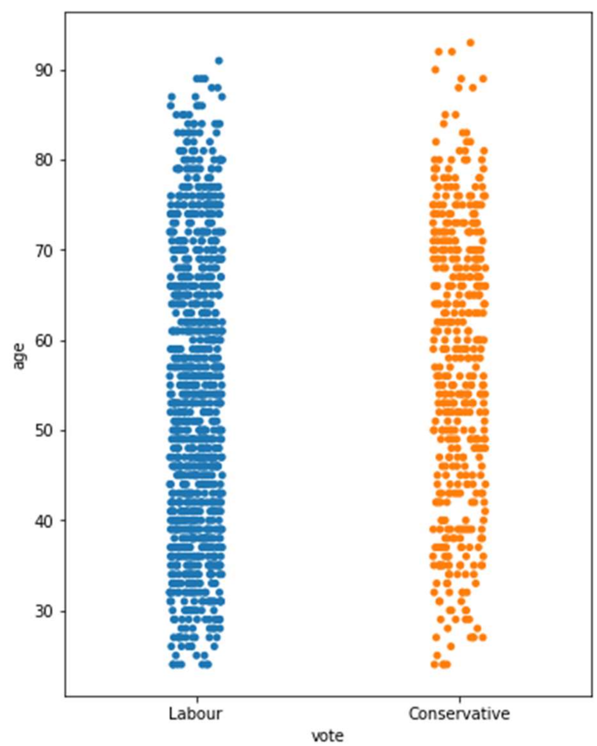
- In Assessment of current national economic conditions, most of the voters have given 3&4.
- In Assessment of current household economic conditions, most of the voters have given 3&4.
- Most of the voters have assessed the Labour leader with 4, i.e., around 800 voters.
- Most of the voters have assessed the Conservative leader with 2 & 4, highest rating being 2 with approximately 600 voters.
- In the voter's respondent towards European integration attitude toward European integration chart, around 350 voters have given scale rating of 11 and around 200 voters have given scale rating of 6.
- In Political Knowledge chart most of the voters given rating of 2 & 3.

❖ <u>Bi-Variate Analysis:</u>



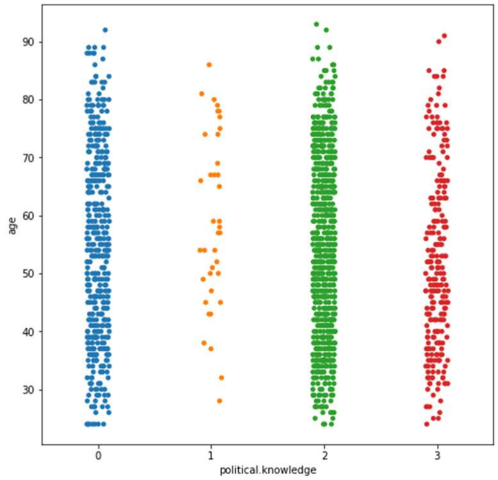
- From the bar plot we can say that more than 1000 people have voted for Labour Party and around 450 people voted for Conservative Party

❖ <u>Age V/s Party Choice:</u>



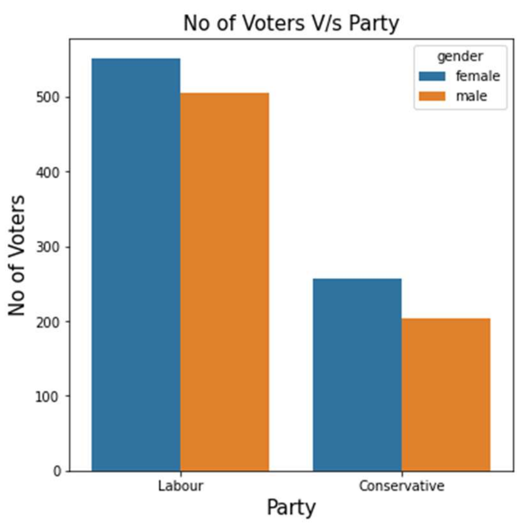
- From the plot we can see that people of all ages voted for both parties
- The density of voters is more for Labour party

❖ Age V/s Political Knowledge:



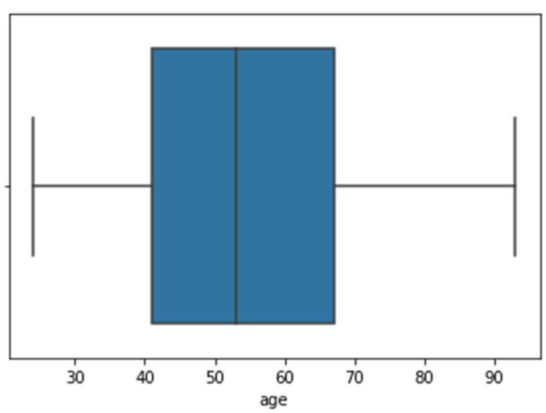
- From the above plot we can see that mostly, voters of all ages are aware of the party's position of European Integration

❖ Party Choice/Gender V/s No of Voters:



- Labour Party gets highest proportion of votes from both Female & Male.

❖ Boxplot for Age:



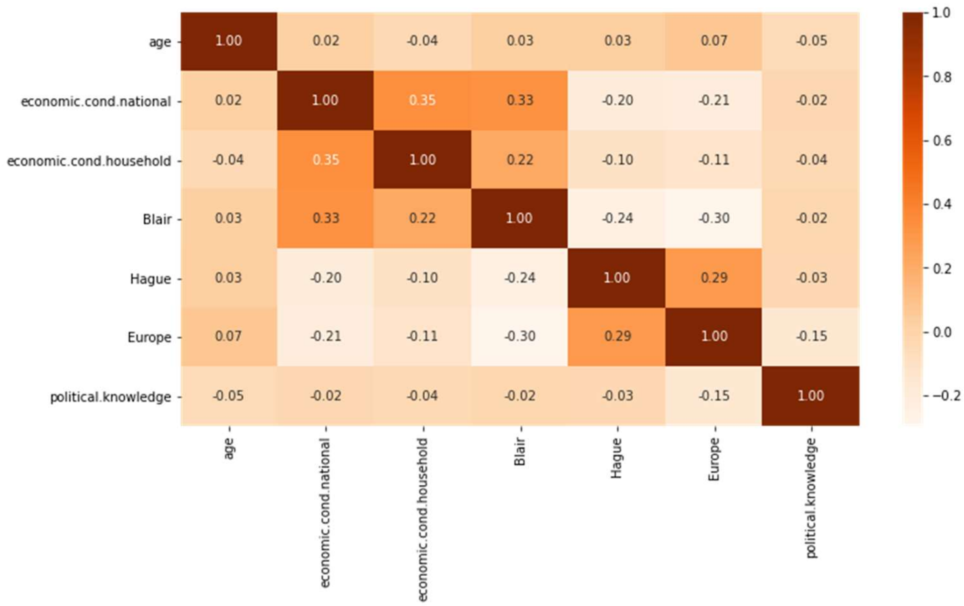
- From the plot age is equally distributed and has no outliers

❖ Multi-Variate Analysis: (Pair-Plot)

From the above Pair-Plot we can say the following

- Age is Normally distributed
- National economic condition looks normal with multiple peaks
- Household economic condition looks normal with multiple peaks
- Blair looks normal with two peaks
- Hague looks normal with multiple peaks
- Political Knowledge looks normal with multiple peaks

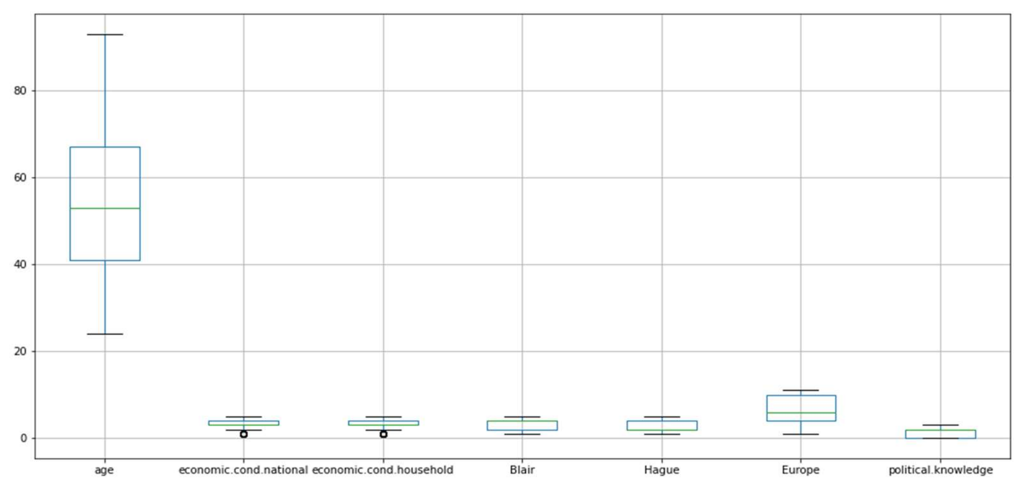
❖ Heat Map:



From the above heat map, we can say the following

- There are no multi-collinearity amount variables.
- Ratings of House-hold Economic conditions is marginally having positive correlation with National Economic conditions.
- Similarly, there is marginal positive correlation between Ratings of National Economic conditions with Blair.

❖ Boxplot for checking any outliers present in the data set:



- From the above box plot we can see that there are no major outliers present in the data set.

1.3) Encode the data (having string values) for Modelling. Is Scaling necessary here or not? Data Split: Split the data into train and test (70:30)

❖ **Data Encoding:**

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

From the above data set we can see that variable 'vote' and 'gender' are of categorical having string values. In order to build the model, we need to do categorical encoding.

- **Categorical Encoding:** Categorical encoding is a process of converting categories to numbers. Different approaches for Categorical Encoding,

1. **Label Encoding:** Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering or ordinal.
2. **One Hot Encoding:** One-Hot Encoding is another popular technique for treating categorical variables. It simply creates additional features based on the number of unique values in the categorical feature. Every unique value in the category will be added as a feature. In our case we will be proceeding with one hot encoding for Gender & Vote. For balance categorical variables we will be doing label encoding.

❖ Dataset after completing Encoding:

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	vote_Labour	gender_male
0	43	3	3	4	1	2	2	1	0
1	36	4	4	4	4	5	2	1	1
2	35	4	4	5	2	3	2	1	1
3	24	4	2	2	1	4	0	1	0
4	41	2	2	1	1	6	2	1	1

From the above data set we can see that now all the variables have been converted into numerical data.

❖ Scaling the 'Age' variable:

From the above data we can see that age is of double digits and rest of other variables are of single digits. However, scaling does not have impact on Logistic Regression/LDA. Since KNN model is distance-based algorithm, scaling need to be done. Hence, we will be doing scaling for the age variable.

❖ Dataset after scaling:

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	vote_Labour	gender_male
0	-0.71616	3	3	4	1	2	2	1	0
1	-1.16212	4	4	4	4	5	2	1	1
2	-1.22583	4	4	5	2	3	2	1	1
3	-1.92662	4	2	2	1	4	0	1	0
4	-0.84358	2	2	1	1	6	2	1	1

❖ Data Split: Train Test Split

- First step is to separate the Target Variable , we will be dropping the Target from the dataset and assigning it to variable 'X' and pop in another variable 'y'.

```
X = df1.drop('vote_Labour',axis=1)     # Copy all predictor variables into X data frame

y = df1[['vote_Labour']]               # Copy target variable into the y data frame
```

- Now we will split the data into train and test. The training data consists of 70% of the data and testing data consists of 30% with random state = 1

```
X_train, X_test,y_train,y_test = train_test_split(X, y, test_size = 0.3, random_state = 1)
```

❖ Checking the Top 5 Records of the Training Set

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
991	-1.289535	2	4	1	4	11	2	0
1274	-0.907286	4	3	4	4	6	0	1
649	0.430587	4	3	4	4	7	2	0
677	-0.461328	3	3	4	2	11	0	1
538	-0.652453	5	3	4	2	8	0	1

❖ Checking the Top 5 Records of the Test Set:

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
504	1.067669	3	3	2	2	8	2	0
369	-0.716161	3	2	4	2	8	3	1
1075	2.214417	5	5	5	2	1	2	1
1031	-0.461328	2	3	2	4	8	2	0
1329	-1.353243	5	4	4	4	8	0	1

1.4) Apply Logistic Regression and LDA (linear discriminant analysis)

❖ **<u>Logistic Regression:</u>**

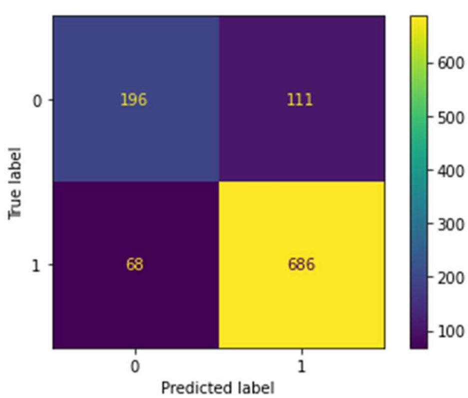
- Importing Logistic Regression from Sklearn library
- Creating Logistic Regression Model with parameters (solver = 'newton-cg', max_iter = 1000, penalty = 'none', verbose = True, n_jobs = 2)
  - ✓ Solver = newton-sg is applied as the computes faster
  - ✓ max_iter = 1000 # Means it will stop after 1000 iteration if it is unable to find optimal values
  - ✓ penalty = 'none' # means no regularization is required
  - ✓ Verbose = True # it the prints out the parameters
  - ✓ n_jobs = 2 # no of cores used parallely

- Fitting the model into the Training Data Set
- Checking the Feature Importance in Logistic Regression,

```
The coeff of age is -0.2354175928898455
The coeff of economic.cond.national is 0.6375859258584184
The coeff of economic.cond.household is 0.06123036423892158
The coeff of Blair is 0.6045934756735092
The coeff of Hague is -0.8294485069901592
The coeff of Europe is -0.21178547460952699
The coeff of political.knowledge is -0.3252373426867469
The coeff of gender_male is 0.19912296601084142
```

❖ **<u>Performance Metrics on Training Data Set</u>**

- Accuracy Score for Training Set is : 83.12912346842602
- Confusion Matrix for Training Set



- Classification Report for Training Set

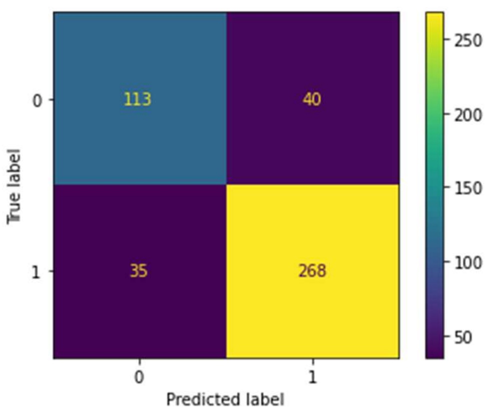
	precision	recall	f1-score	support
0	0.74	0.64	0.69	307
1	0.86	0.91	0.88	754
accuracy			0.83	1061
macro avg	0.80	0.77	0.79	1061
weighted avg	0.83	0.83	0.83	1061

From the above we can say the following for Training,

- Accuracy/Model score for Training Set is **83.12** which is shows that our model is performing good.
- Precision for Labour (Class = 1) is **0.86** & Recall for Labour (Class = 1) is **0.91** which is good.
- From the confusion matrix we can see that our model predicted Labour (Class = 1) 686 times correctly which is good prediction.

❖ **<u>Performance Metrics on Test Data Set</u>**

- Accuracy Score for Test Set is : 83.55263157894737
- Confusion Matrix for Test Set



- Classification Report for Test Set

	precision	recall	f1-score	support
0	0.76	0.74	0.75	153
1	0.87	0.88	0.88	303
accuracy			0.84	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.84	0.83	456

From the above we can say the following for Test,

- Accuracy/Model score for Test Set is **83.55** which is shows that our model is performing good.
- Precision for Labour (Class = 1) is **0.87** & Recall for Labour (Class = 1) is **0.88** which shows that our model is predicting great in our test set.
- From the confusion matrix we can see that our model predicted Labour (Class = 1) 268 times correctly which is good prediction

❖ **<u>Comparison Between Training & Test data set: Logistic Regression</u>**

	Log Regr Train	**Log Regr Test**
Accuracy	83.13	83.55
Precision	86	87
Recall	91	88

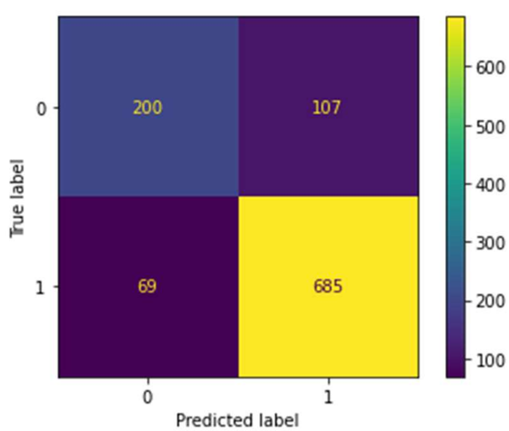
From the table we can see that there is no case of underfitting/overfitting and the values are within industrial limits (~10%)

❖ **<u>LDA (linear discriminant analysis)</u>**

- Importing LDA from Sklearn library
- Creating LDA Model with parameters (solver = 'svd', shrinkage = None, priors = None, n_components = None)
- Fitting the model into the Training Data Set

❖ **<u>Performance Metrics on Training Data Set</u>**

- Accuracy Score for Training Set is : 83.41187558906692
- Confusion Matrix for Training Set



- Classification Report for Training Set

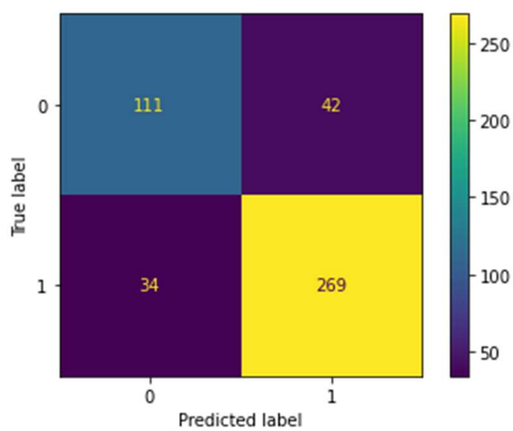
	precision	recall	f1-score	support
0	0.74	0.65	0.69	307
1	0.86	0.91	0.89	754
accuracy			0.83	1061
macro avg	0.80	0.78	0.79	1061
weighted avg	0.83	0.83	0.83	1061

From the above we can say the following for Training,

- Accuracy/Model score for Training Set is **83.41** which is shows that our model is performing good.
- Precision for Labour (Class = 1) is **0.86** & Recall for Labour (Class = 1) is **0.91** which is good.
- From the confusion matrix we can see that our model predicted Labour (Class = 1) **685 times** correctly which is good prediction

❖ **Performance Metrics on Test Data Set**

- Accuracy Score for Test Set is : 83.33
- Confusion Matrix for Training Set



- Classification Report for Test Set

	precision	recall	f1-score	support
0	0.77	0.73	0.74	153
1	0.86	0.89	0.88	303
accuracy			0.83	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.83	0.83	456

From the above we can say the following for Test dataset,

- Accuracy/Model score for Test Set is **83.33** which is shows that our model is performing good.
- Precision for Labour (Class = 1) is **0.86** & Recall for Labour (Class = 1) is **0.89** which shows that our model is predicting great in our test set.
- From the confusion matrix we can see that our model predicted Labour (Class = 1) 269 times correctly which is good prediction

❖ **<u>Comparison Between Training & Test data set: LDA</u>**

	LDA Train	LDA Test
Accuracy	83.41	83.33
Precision	86	86
Recall	91	89

From the table we can see that there is no case of underfitting/overfitting and the values are within industrial limits (~10%)

❖ **<u>Comparing Performance Matrix of Logistic Regression & Linear Discriminant Analysis</u>**

	Log Regr Train	Log Regr Test	LDA Train	LDA Test
Accuracy	83.13	83.55	83.41	83.33
Precision	86	87	86	86
Recall	91	88	91	89

From the above table we can say the following

- Accuracy score for both Logistic Regression & Linear Discriminant Analysis is almost same with marginal difference.
- Precision for both Logistic Regression & Linear Discriminant Analysis is almost same with marginal difference.
- Recall for both Logistic Regression & Linear Discriminant Analysis is almost same with marginal difference.
- There is overfit/underfit issue
- Both model performance is good and same.

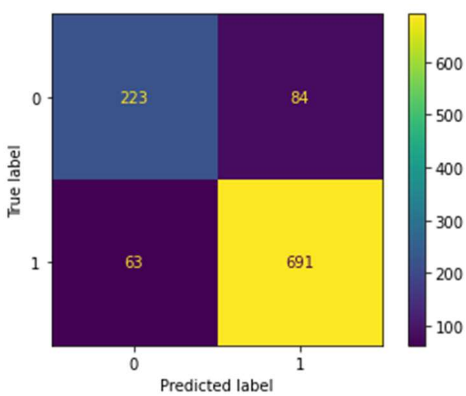
1.5) Apply KNN Model and Naïve Bayes Model. Interpret the results

❖ **KNN Model:**

- Importing KNN Model from sklearn library
- Creating KNN Model with parameters(n_neighbors = 5, weights = 'uniform', algorithm = 'auto')
  - ✓ n_neighbors = 5, means that the k value = 5
  - ✓ weights = uniform, means the model will distribute weights uniformly
  - ✓ algorithm = auto, means systems automatically select which suits better
- Fitting the model into the Training Data Set
- Predicting on Train and Test data set

❖ **Performance Metric on Training Data**

- Accuracy Score for Test Set is : 86.14
- Confusion Matrix for Training Set



- Classification Report for Training Set

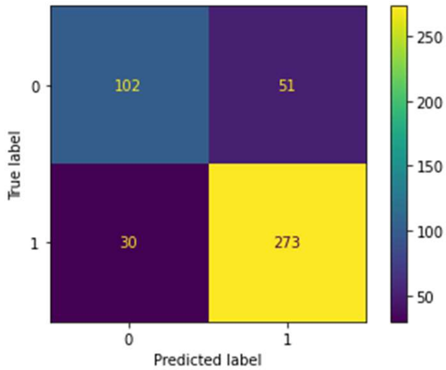
	precision	recall	f1-score	support
0	0.78	0.73	0.75	307
1	0.89	0.92	0.90	754
accuracy			0.86	1061
macro avg	0.84	0.82	0.83	1061
weighted avg	0.86	0.86	0.86	1061

From the above we can say the following for Training,

- Accuracy/Model score for Training Set is 86.14 which is shows that our model is performing good.
- Precision for Labour (Class = 1) is 0.89 & Recall for Labour (Class = 1) is 0.92 which is good.
- From the confusion matrix we can see that our model predicted Labour (Class = 1) 691 times correctly which is good prediction

❖ **<u>Performance Metric on Test Data</u>**

- Accuracy Score for Test Set is : 82.23
- Confusion Matrix for Test Set



- Classification Report for Test Set

	precision	recall	f1-score	support
0	0.77	0.67	0.72	153
1	0.84	0.90	0.87	303
accuracy			0.82	456
macro avg	0.81	0.78	0.79	456
weighted avg	0.82	0.82	0.82	456

From the above we can say the following for Test dataset,

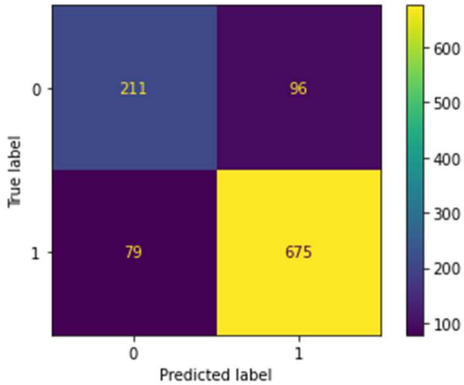
- Accuracy/Model score for Test Set is 82.23 which is shows that our model is performing good.
- Precision for Labour (Class = 1) is 0.84 & Recall for Labour (Class = 1) is 0.90 which shows that our model is predicting good in our test set.
- From the confusion matrix we can see that our model predicted Labour (Class = 1) 273 times correctly which is good prediction

❖ **<u>Naïve Bayes Model:</u>**

- Importing Gaussian NB from sklearn library
- Creating NB Model with default parameters
- Fitting the model into the Training Data Set
- Predicting on Train and Test data set

❖ **<u>Performance Metric on Training Data</u>**

- Accuracy Score for Test Set is : 83.50
- Confusion Matrix for Training Set



- Classification Report for Training Set

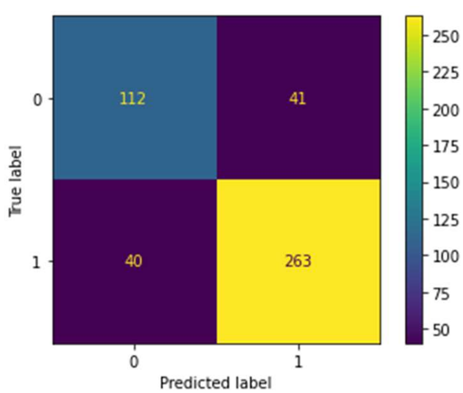
	precision	recall	f1-score	support
0	0.73	0.69	0.71	307
1	0.88	0.90	0.89	754
accuracy			0.84	1061
macro avg	0.80	0.79	0.80	1061
weighted avg	0.83	0.84	0.83	1061

From the above we can say the following for Training,

- Accuracy/Model score for Training Set is 83.50 which is shows that our model is performing good.
- Precision for Labour (Class = 1) is 0.88 & Recall for Labour (Class = 1) is 0.90 which is good.
- From the confusion matrix we can see that our model predicted Labour (Class = 1) 675 times correctly which is good prediction.

❖ **<u>Performance Metric on Test Data</u>**

- Accuracy Score for Test Set is : 82.23
- Confusion Matrix for Test Set



- Classification Report for Test Set

```
  precision    recall  f1-score   support

           0       0.74      0.73      0.73       153
           1       0.87      0.87      0.87       303

    accuracy                           0.82       456
   macro avg       0.80      0.80      0.80       456
weighted avg       0.82      0.82      0.82       456
```

From the above we can say the following for Test dataset,

- Accuracy/Model score for Test Set is 82.23 which is shows that our model is performing good.
- Precision for Labour (Class = 1) is 0.87 & Recall for Labour (Class = 1) is 0.87 which shows that our model is predicting marginally good in our test set.
- From the confusion matrix we can see that our model predicted Labour (Class = 1) 263 times correctly which is marginally good prediction.

❖ **Comparing Performance Metrics of all four models : Logistic Regression, LDA, KNN & Naive Bayes**

	Log Regr Train	**Log Regr Test**	**LDA Train**	**LDA Test**	**KNN Train**	**KNN Test**	**NB Train**	**NB Test**
Accuracy	83.13	83.55	83.41	83.33	86.15	82.24	83.51	82.24
Precision	86	87	86	86	89	84	88	87
Recall	91	88	91	89	92	90	90	87

From the above table we can say the following,

- Accuracy for Training set is high for KNN model i.e., 86.15
- KNN Model has good recall value on Train & Test set when compared to other models

Hence based on these values we can say that KNN model is performing good on our Target variable before model tuning

1.6) Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting

a) Logistic Regression (Using GridSearchCV)

- Importing Grid Search CV Library from sklearn
- Entering the parameters into the grid so as to find optimal parameters using GridSearchCV.

```
grid = {'penalty':['l2','none'],
       'solver':['sag','saga','newton-cg','lbfgs'],
        'max_iter': [500,1000,1500],
        'tol':[0.001,0.0001]}
```

- Building a GridSearchCV model in order to find best parameters
- Getting Best Parameters for Logistic Regression

```
{'max_iter': 500, 'penalty': 'none', 'solver': 'sag', 'tol': 0.001}
```

- Fitting best parameters into Logistic Regression Model
  - ✓ Max_iteration = 500, means the model will stop after 500 iteration if it is unable to find optimal values
  - ✓ Penalty = None, means regularization is not required
  - ✓ Solver = 'sag' performs fast on huge data set
  - ✓ n_jobs = -1 uses all the processors available
  - ✓ verbose = True, means displays all the parameters used for building the model
- Fitting our Logistic Regression Model to our Train set
- Predicting on Train & Test Set
- Coefficients of the features

```
The coeff of age is -0.23551138802037014
The coeff of economic.cond.national is 0.6413131463029488
The coeff of economic.cond.household is 0.06313777828116227
The coeff of Blair is 0.6063884880236599
The coeff of Hague is -0.8273921867222934
The coeff of Europe is -0.21088859545256108
The coeff of political.knowledge is -0.32293045861421676
The coeff of gender_male is 0.20069348934293163
```

From the above we can say that following are important features

- Hague (Assessment of the Conservative Leader)
- Assessment of Current National Economic
- Blair (Assessment of Labour Leader)

b) Linear Discriminant Analysis (Using GridSearchCV)

- Entering the parameters into the grid1 so as to find optimal parameters using GridSearchCV

```
grid1 = {'solver':['svd','lsqr','eigen'],
         'tol':[0.001,0.0001],
         'shrinkage': [None],
          'n_components': [None]}
```

- Building a GridSearchCV model and will fit it into our Train set in order to find best parameters
- Getting Best Parameters for LDA
  (n_components': None, 'shrinkage': None, 'solver': 'svd', 'tol': 0.001)
- Fitting best parameters into Linear Discriminant Analysis Model
  - ✓ solver = 'svd', does not calculate covariance matrix
  - ✓ n_components = None, its generally used for dimensional reduction in this case it is not required
  - ✓ priors = None, prior probabilities is none
- Fitting our Linear Discriminant Analysis Model to our Train set
- Predicting on Train & Test Set

c) KNN Model (Using GridSearchCV)

- Entering the parameters into the grid2 so as to find optimal parameters using GridSearchCV

```
grid2 = {'n_neighbors':[5,7,9,11,13,15,17,19],
         'weights':['uniform'],
          'leaf_size' : [30,50,70],
          'algorithm' : ['auto']}
```

- Building a GridSearchCV model and will fit it into our Train set in order to find best parameters
- Getting Best Parameters for KNN Model
  ({'algorithm': 'auto', 'leaf_size': 30, 'n_neighbors': 9, 'weights': 'uniform'})
- Fitting best parameters into KNN Model
- Predicting on Train & Test Set

d) Naïve Bayes Model (Using GridSearchCV)

- Entering the parameters into the grid2 so as to find optimal parameters using GridSearchCV

```
grid3 = {'var_smoothing':[1e-9,1e-8, 1e-7, 1e-6, 1e-5, 1e-4, 1e-3]}
```

- Building a GridSearchCV model and will fit it into our Train set in order to find best parameters
- Getting best parameters for NB Model ({'var_smoothing': 0.001})
- Building our Naive Bayes Model using best parameters from GridSearchCV
- Fitting our Naive Bayes Model to our Train set
- Predicting on Train and Test Set

e) Bagging (Using Random Forest)

- Importing Bagging Classifier from sklearn library
- Importing Random Forest Classifier from sklearn library
- Using GridSearchCV for getting best parameters for building Random Forest Classifier

```
param_grid = {
    'max_depth': [6,7,8],
    'max_features': [6,7,8],
    'min_samples_leaf': [10,20,30],
    'min_samples_split': [30, 60,90],
    'n_estimators': [301, 501]
```

- Getting Best Parameters {'max_depth': 7, 'max_features': 6, 'min_samples_leaf': 10, 'min_samples_split': 30, 'n_estimators': 501}
- First, we will build random forest classifier model with best parameters
- Keeping the random forest model as base estimator for our bagging classifier
- Fitting the model into our Training set
- Predicting on Train & Test Data

f) Gradient Boosting

- Importing Gradient Boosting Classifier from sklearn
- Building Gradient Boosting Classifier model with Random State = 1
- Fitting our model to the training set
- Predicting on Training & Test Set

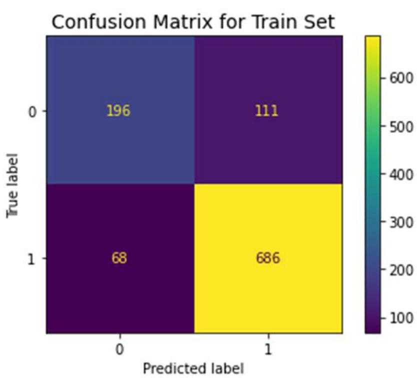
g) **Adaptive Boosting**

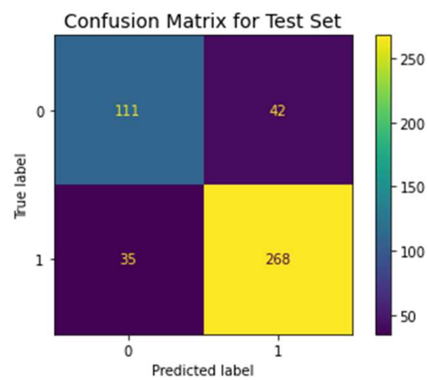
- Importing Adaptive Boosting Classifier from sklearn
- Building Adaptive Boosting Classifier model with n_estimators = 100 & random state = 1
- Fitting our model to the training set
- Predicting on Training & Test Set

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. Final Model: Compare the models and write inference which model is best/optimized

a) Logistic Regression (Tuned Model)

- Accuracy Score for Logistic Regression Training Set is : 83.12
- Accuracy Score for Logistic Regression Test Set is : 83.11
- Confusion Matrix for Train and Test Set





From the confusion matrix we can see the following,

- ✓ Model predicted Labour (Class = 1) 686 times for Train Set
- ✓ Model predicted Conservative (Class = 0) 196 times for Train Set
- ✓ Model predicted Labour (Class = 1) 268 times for Test Set
- ✓ Model predicted Conservative (Class = 0) 111 times for Test Set

- Classification Report for Train and Test Set

❖ **Train Set:**

	precision	recall	f1-score	support
0	0.74	0.64	0.69	307
1	0.86	0.91	0.88	754
accuracy			0.83	1061
macro avg	0.80	0.77	0.79	1061
weighted avg	0.83	0.83	0.83	1061

From the above report we can say the following for Training set,

- ✓ Precision for Class = 0 (Conservative) is 0.74
- ✓ Recall for Class = 0 (Conservative) is 0.64
- ✓ Precision for Class = 1 (Labour) is 0.86
- ✓ Recall for Class = 1 (Labour) is 0.91

❖ **Test Set**

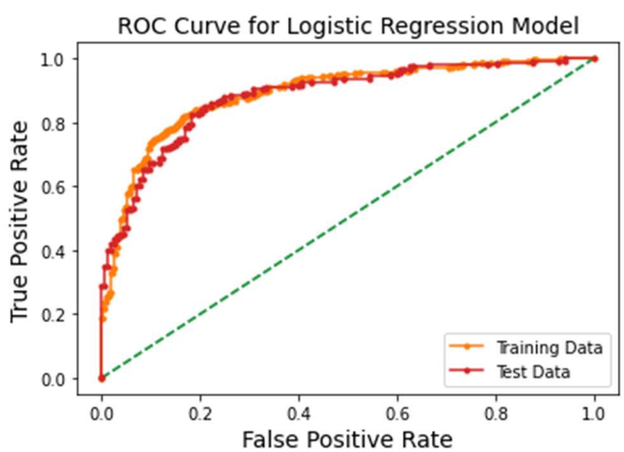
	precision	recall	f1-score	support
0	0.76	0.73	0.74	153
1	0.86	0.88	0.87	303
accuracy			0.83	456
macro avg	0.81	0.80	0.81	456
weighted avg	0.83	0.83	0.83	456

From the above report we can say the following for Test set,

- ✓ Precision for Class = 0 (Conservative) is 0.76
- ✓ Recall for Class = 0 (Conservative) is 0.73
- ✓ Precision for Class = 1 (Labour) is 0.86
- ✓ Recall for Class = 1 (Labour) is 0.88

By Comparing both the Training & Test set we can see that our model performs well. But there is no improvement when compared to earlier model i.e., not tuned model

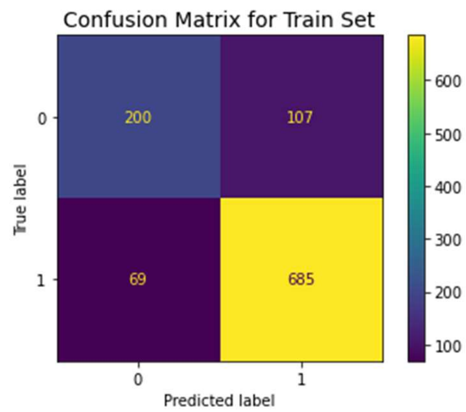
- AUC Score & ROC Curve for Training & Test Set

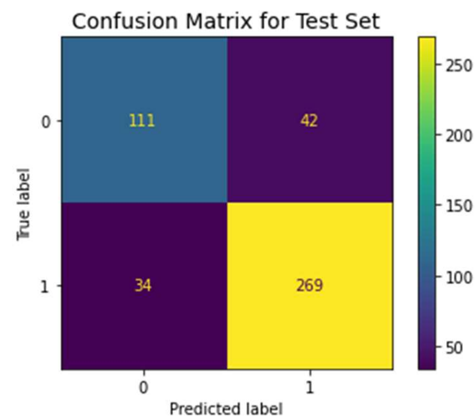


- ✓ AUC for Training 89.00
- ✓ AUC for Test 88.26

b) Linear Discriminant Analysis (Tuned Model)

- Accuracy Score for Linear Discriminant Analysis Training Set is : 83.41
- Accuracy Score for Linear Discriminant Analysis Test Set is : 83.33
- Confusion Matrix for Training and Test Set





From the confusion matrix we can see the following,

- ✓ Model predicted Labour (Class = 1) 685 times for Train set
- ✓ Model predicted Conservative (Class = 0) 200 times for Train set
- ✓ Model predicted Labour (Class = 1) 269 times for Test set
- ✓ Model predicted Conservative (Class = 0) 111 times for Test set

- Classification Report for Train and Test Set

❖ **<u>Train Set:</u>**

	precision	recall	f1-score	support
0	0.74	0.65	0.69	307
1	0.86	0.91	0.89	754
accuracy			0.83	1061
macro avg	0.80	0.78	0.79	1061
weighted avg	0.83	0.83	0.83	1061

From the above report we can say the following for Training set,

- ✓ Precision for Class = 0 (Conservative) is 0.74
- ✓ Recall for Class = 0 (Conservative) is 0.65
- ✓ Precision for Class = 1 (Labour) is 0.86
- ✓ Recall for Class = 1 (Labour) is 0.91

❖ **<u>Test Set:</u>**

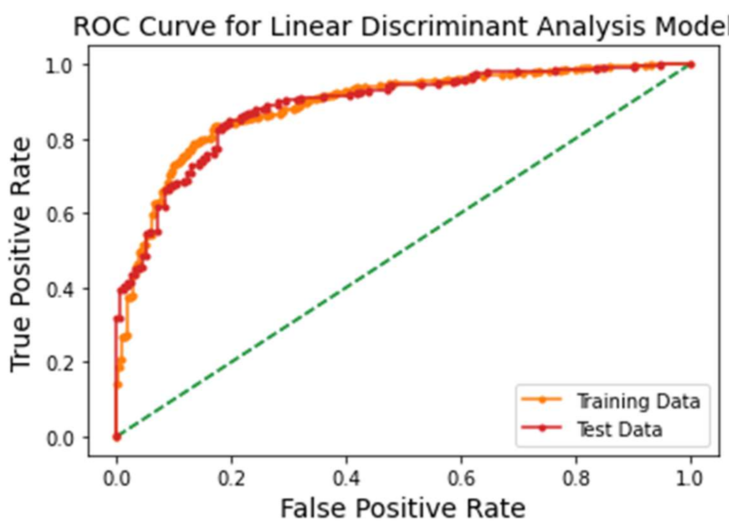
	precision	recall	f1-score	support
0	0.77	0.73	0.74	153
1	0.86	0.89	0.88	303
accuracy			0.83	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.83	0.83	456

From the above report we can say the following for Test set,

- ✓ Precision for Class = 0 (Conservative) is 0.77
- ✓ Recall for Class = 0 (Conservative) is 0.73
- ✓ Precision for Class = 1 (Labour) is 0.86
- ✓ Recall for Class = 1 (Labour) is 0.89

By Comparing both the Training & Test set we can see that our model performs well. But there is no improvement when compared to earlier model i.e. not tuned model

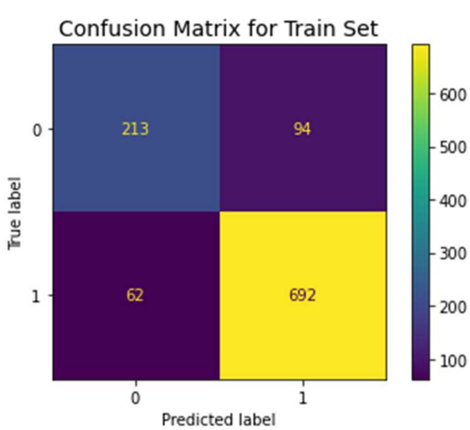
- AUC Score & ROC Curve for Training & Test Set

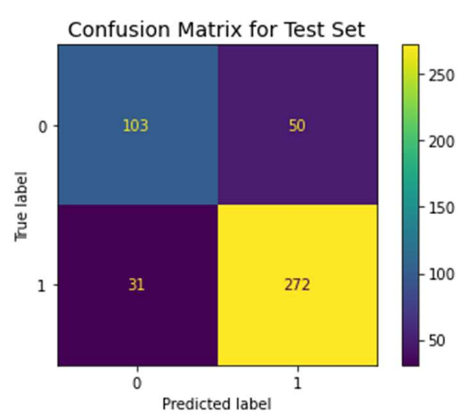


- ✓ AUC for Training 88.93
- ✓ AUC for Test 88.76

c) KNN Model (Tuned Model)

- Accuracy Score for KNN Model Training Set is : 85.29
- Accuracy Score for KNN Model Test Set is : 82.23
- Confusion Matrix for Training and Test Set





From the confusion matrix we can see the following,

- ✓ Model predicted Labour (Class = 1) 692 times for Train set
- ✓ Model predicted Conservative (Class = 0) 213 times for Train set
- ✓ Model predicted Labour (Class = 1) 272 times for Test set
- ✓ Model predicted Conservative (Class = 0) 103 times for Test set

- Classification Report for Train and Test Set

❖ **Train Set:**

	precision	recall	f1-score	support
0	0.77	0.69	0.73	307
1	0.88	0.92	0.90	754
accuracy			0.85	1061
macro avg	0.83	0.81	0.82	1061
weighted avg	0.85	0.85	0.85	1061

From the above report we can say the following for Training set,

- ✓ Precision for Class = 0 (Conservative) is 0.77
- ✓ Recall for Class = 0 (Conservative) is 0.69
- ✓ Precision for Class = 1 (Labour) is 0.88
- ✓ Recall for Class = 1 (Labour) is 0.92

❖ **Test Set:**

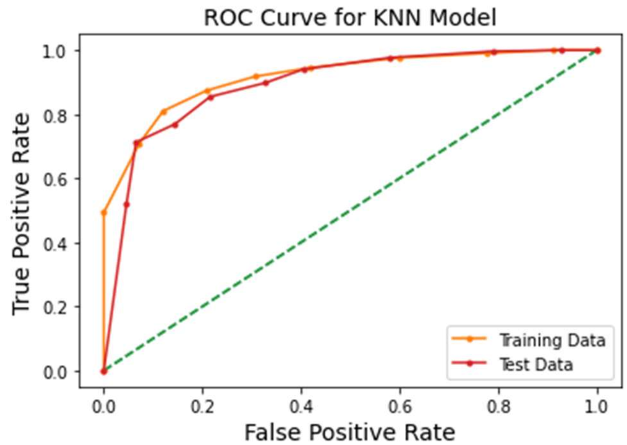
	precision	recall	f1-score	support
0	0.77	0.67	0.72	153
1	0.84	0.90	0.87	303
accuracy			0.82	456
macro avg	0.81	0.79	0.79	456
weighted avg	0.82	0.82	0.82	456

From the above report we can say the following for Test set,

- ✓ Precision for Class = 0 (Conservative) is 0.77
- ✓ Recall for Class = 0 (Conservative) is 0.67
- ✓ Precision for Class = 1 (Labour) is 0.84
- ✓ Recall for Class = 1 (Labour) is 0.90

By Comparing both the Training & Test set we can see that our model performs well. But there is no improvement when compared to earlier model i.e., not tuned model

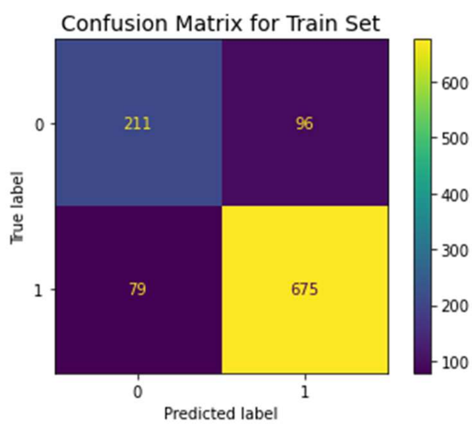
- AUC Score & ROC Curve for Training & Test Set

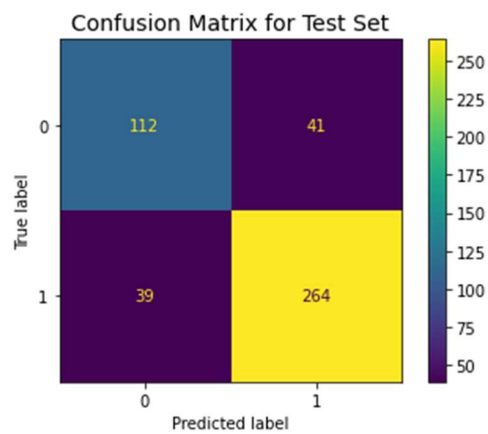


✓ AUC for Training 91.71
✓ AUC for Test 89.43

d) Naïve Bayes (Tuned Model)

- Accuracy Score for NB Model Training Set is : 83.50
- Accuracy Score for NB Model Test Set is : 82.45
- Confusion Matrix for Training and Test Set





From the confusion matrix we can see the following,

✓ Model predicted Labour (Class = 1) 675 times for Train set
✓ Model predicted Conservative (Class = 0) 211 times for Train set
✓ Model predicted Labour (Class = 1) 264 times for Test set
✓ Model predicted Conservative (Class = 0) 112 times for Test set

- Classification Report for Train and Test Set

❖ **Train Set:**

	precision	recall	f1-score	support
0	0.73	0.69	0.71	307
1	0.88	0.90	0.89	754
accuracy			0.84	1061
macro avg	0.80	0.79	0.80	1061
weighted avg	0.83	0.84	0.83	1061

From the above report we can say the following for Training set,

- ✓ Precision for Class = 0 (Conservative) is 0.73
- ✓ Recall for Class = 0 (Conservative) is 0.69
- ✓ Precision for Class = 1 (Labour) is 0.88
- ✓ Recall for Class = 1 (Labour) is 0.90

❖ **Test Set:**

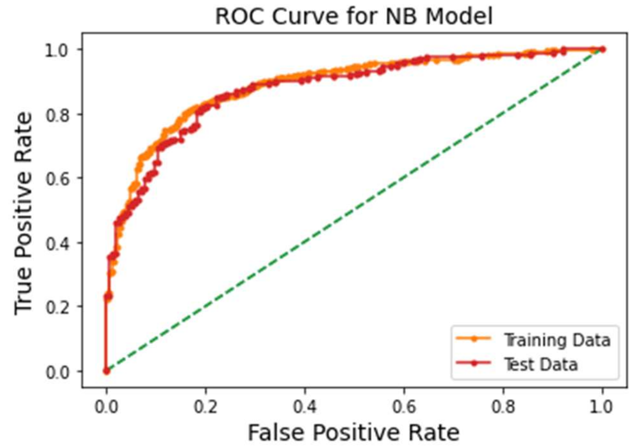
	precision	recall	f1-score	support
0	0.74	0.73	0.74	153
1	0.87	0.87	0.87	303
accuracy			0.82	456
macro avg	0.80	0.80	0.80	456
weighted avg	0.82	0.82	0.82	456

From the above report we can say the following for Test set,

- ✓ Precision for Class = 0 (Conservative) is 0.74
- ✓ Recall for Class = 0 (Conservative) is 0.73
- ✓ Precision for Class = 1 (Labour) is 0.87
- ✓ Recall for Class = 1 (Labour) is 0.87

By Comparing both the Training & Test set we can see that our model performs well. But there is no improvement when compared to earlier model i.e., not tuned model

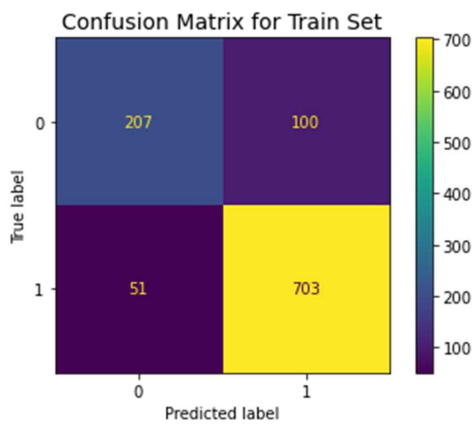
- AUC Score & ROC Curve for Training & Test Set

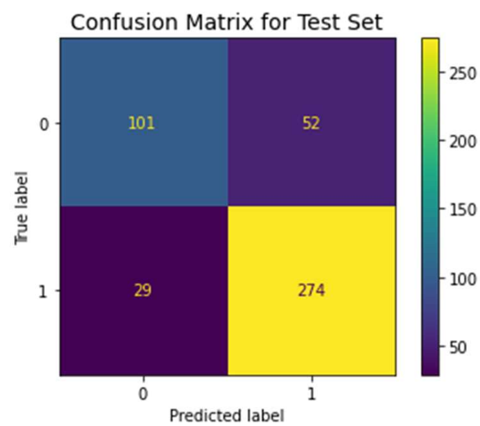


✓ AUC for Training 88.79
✓ AUC for Test 87.66

e) Bagging (Random Forest Classifier)

- Accuracy Score for Bagging Model Training Set is : 85.76
- Accuracy Score for Bagging Model Test Set is : 82.23
- Confusion Matrix for Training and Test Set





From the confusion matrix we can see the following,

✓ Model predicted Labour (Class = 1) 703 times for Train set
✓ Model predicted Conservative (Class = 0) 207 times for Train set
✓ Model predicted Labour (Class = 1) 274 times for Test set
✓ Model predicted Conservative (Class = 0) 101 times for Test set

- Classification Report for Training and Test Set

❖ **Train Set:**

	precision	recall	f1-score	support
0	0.80	0.67	0.73	307
1	0.88	0.93	0.90	754
accuracy			0.86	1061
macro avg	0.84	0.80	0.82	1061
weighted avg	0.85	0.86	0.85	1061

From the above report we can say the following for Training set,

✓ Precision for Class = 0 (Conservative) is 0.80
✓ Recall for Class = 0 (Conservative) is 0.67
✓ Precision for Class = 1 (Labour) is 0.88
✓ Recall for Class = 1 (Labour) is 0.93

❖ **Test Set:**

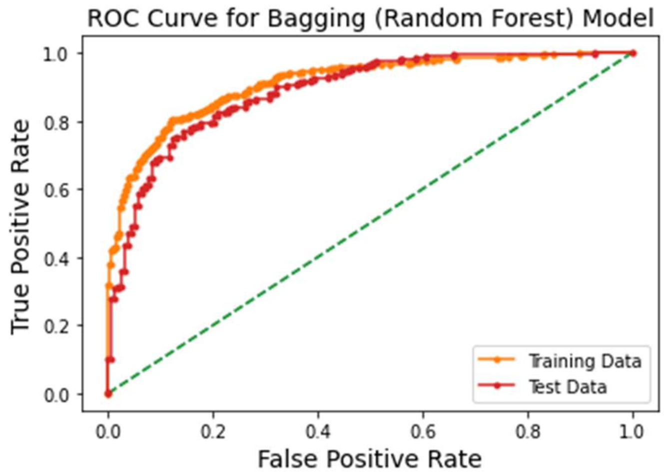
	precision	recall	f1-score	support
0	0.78	0.66	0.71	153
1	0.84	0.90	0.87	303
accuracy			0.82	456
macro avg	0.81	0.78	0.79	456
weighted avg	0.82	0.82	0.82	456

From the above report we can say the following for Test set,

✓ Precision for Class = 0 (Conservative) is 0.78
✓ Recall for Class = 0 (Conservative) is 0.66
✓ Precision for Class = 1 (Labour) is 0.84
✓ Recall for Class = 1 (Labour) is 0.90

By Comparing both the Training & Test set we can see that our model performs well

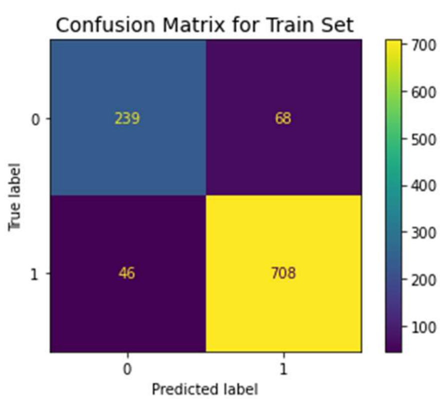
- AUC Score & ROC Curve for Training & Test Set

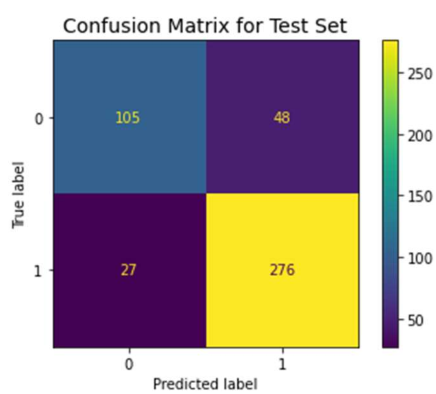


- ✓ AUC for Training 91.24
- ✓ AUC for Test 88.86

f) Gradient Boosting

- Accuracy Score for Gradient Boosting Training Set is : 89.25
- Accuracy Score for Gradient Boosting Test Set is : 83.55
- Confusion Matrix for Training and Test Set





From the confusion matrix we can see the following,

- ✓ Model predicted Labour (Class = 1) 708 times for Train set
- ✓ Model predicted Conservative (Class = 0) 239 times for Train set
- ✓ Model predicted Labour (Class = 1) 276 times for Test set
- ✓ Model predicted Conservative (Class = 0) 105 times for Test set

- Classification Report for Training and Test Set

❖ **Train Set:**

	precision	recall	f1-score	support
0	0.84	0.78	0.81	307
1	0.91	0.94	0.93	754
accuracy			0.89	1061
macro avg	0.88	0.86	0.87	1061
weighted avg	0.89	0.89	0.89	1061

From the above report we can say the following for Training set,

- ✓ Precision for Class = 0 (Conservative) is 0.84
- ✓ Recall for Class = 0 (Conservative) is 0.78
- ✓ Precision for Class = 1 (Labour) is 0.91
- ✓ Recall for Class = 1 (Labour) is 0.94

❖ **Test Set:**

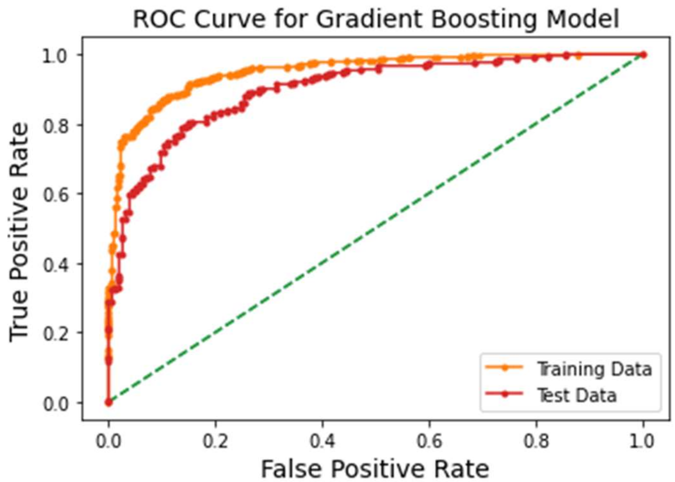
	precision	recall	f1-score	support
0	0.80	0.69	0.74	153
1	0.85	0.91	0.88	303
accuracy			0.84	456
macro avg	0.82	0.80	0.81	456
weighted avg	0.83	0.84	0.83	456

From the above report we can say the following for Test set,

- ✓ Precision for Class = 0 (Conservative) is 0.80
- ✓ Recall for Class = 0 (Conservative) is 0.69
- ✓ Precision for Class = 1 (Labour) is 0.95
- ✓ Recall for Class = 1 (Labour) is 0.91

By Comparing both the Training & Test set we can see that our model performs well

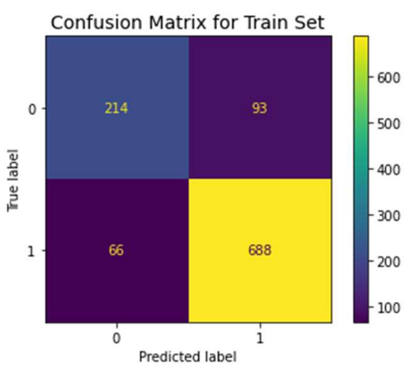
- AUC Score & ROC Curve for Training & Test Set

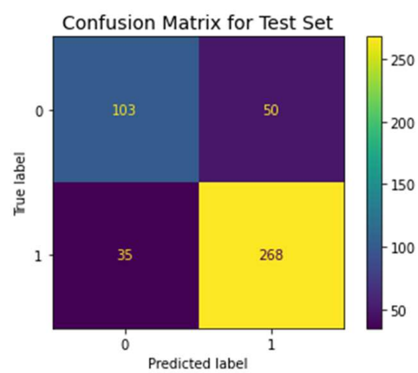


- ✓ AUC for Training 95.11
- ✓ AUC for Test 89.92

g) Adaptive Boosting

- Accuracy Score for Adaptive Boosting Training Set is : 85.01
- Accuracy Score for Adaptive Boosting Test Set is : 81.35
- Confusion Matrix for Training and Test Set





From the confusion matrix we can see the following,

- ✓ Model predicted Labour (Class = 1) 688 times for Train set
- ✓ Model predicted Conservative (Class = 0) 214 times for Train set
- ✓ Model predicted Labour (Class = 1) 268 times for Test set
- ✓ Model predicted Conservative (Class = 0) 103 times for Test set

- ✓ Classification Report for Training and Test Set

❖ **Train Set:**

	precision	recall	f1-score	support
0	0.76	0.70	0.73	307
1	0.88	0.91	0.90	754
accuracy			0.85	1061
macro avg	0.82	0.80	0.81	1061
weighted avg	0.85	0.85	0.85	1061

From the above report we can say the following for Training set,

- ✓ Precision for Class = 0 (Conservative) is 0.76
- ✓ Recall for Class = 0 (Conservative) is 0.70
- ✓ Precision for Class = 1 (Labour) is 0.88
- ✓ Recall for Class = 1 (Labour) is 0.91

❖ **Test Set:**

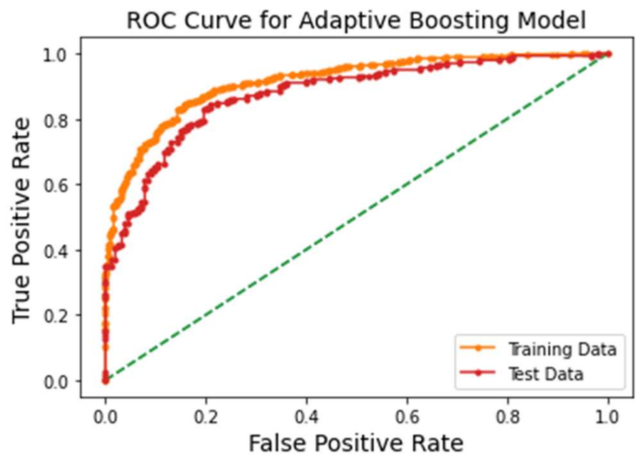
	precision	recall	f1-score	support
0	0.75	0.67	0.71	153
1	0.84	0.88	0.86	303
accuracy			0.81	456
macro avg	0.79	0.78	0.79	456
weighted avg	0.81	0.81	0.81	456

From the above report we can say the following for Test set,

- ✓ Precision for Class = 0 (Conservative) is 0.75
- ✓ Recall for Class = 0 (Conservative) is 0.67
- ✓ Precision for Class = 1 (Labour) is 0.84
- ✓ Recall for Class = 1 (Labour) is 0.88

By Comparing both the Training & Test set we can see that our model performs well

✓ AUC Score & ROC Curve for Training & Test Set



✓ AUC for Training 91.48
✓ AUC for Test 87.73

- **Comparison for Prediction of Class = 1 (Labour) of All the models**

S.No	Model Name	Data Set	Accuracy	Precision	Recall	f1 Score	AUC Score
1	Logistic Regression	Train Set	83.12	86	91	88	89
		Test Set	83.11	86	88	87	88.26
2	Linear Discriminant Analysis	Train Set	83.41	86	91	89	88.93
		Test Set	83.33	86	89	88	88.76
3	K-Nearest Neighbors	Train Set	85.29	88	92	90	91.71
		Test Set	82.23	84	90	87	89.43
4	Naïve Bayes	Train Set	83.5	88	90	89	88.79
		Test Set	82.45	87	87	87	87.66
5	Bagging (Random Forest)	Train Set	85.76	88	93	90	91.24
		Test Set	82.23	84	90	87	88.86
6	Gradient Boosting	Train Set	89.25	91	94	93	95.11
		Test Set	83.55	85	91	88	89.92
7	Adaptive Boositing	Train Set	85.01	88	91	90	91.48
		Test Set	81.35	84	88	86	87.73

From the above table we can see comparison of Performance Metrics of all the models, based on the above table we can say the following,

✓ Gradient Boosting (Train Set) is having Highest Accuracy score i.e. 89.25
✓ Gradient Boosting (Test Set) is having Highest Accuracy score i.e. 83.55
✓ Gradient Boosting (Train Set) is having Highest Precision value i.e. 91
✓ Naive Bayes (Test Set) is having Highest Precision value i.e. 87

- ✓ Gradient Boosting (Train Set) is having Highest Recall value i.e. 94
- ✓ Gradient Boosting (Test Set) is having Highest Recall value i.e. 91
- ✓ Gradient Boosting (Train Set) is having Highest AUC Score value i.e. 95.11
- ✓ Gradient Boosting (Test Set) is having Highest AUC Score value i.e. 89.92

Based on the precision, recall and accuracy scores we can say that **Gradient Boosting** model is performing well on the Train and Test data sets. Hence Gradient Boosting is best suitable model

1.8) Based on these predictions, what are the insights?

We had a business problem where we need to predict which party a voter will vote for on basis of given information. The given data consists of **9 variables** and **1525 observations**. Data set consists of no null values but has 8 duplicate values which has been removed from the data set.

Brief about the variables present in the data set

- Party Choice : Conservative or Labour (Target Variable)
- Age
- Assessment of Current National Economic conditions
- Assessment of Current Household Economic conditions
- Assessment of Labour Leader (Blair)
- Assessment of Conservative Leader (Hague)
- Voters attitude towards European Integration
- Parties' positions on European Integration
- Gender

The target variable consists of two parties i.e., Conservative & Labour

From EDA Analysis we can understand the following,

- Age of the voters is normally distributed and ranging from 25 to 90. Maximum age of the voter is 93.
- Using the pie-plot we can see that around 69.7% of voters opted for Labour Party and 30.3% of voters opted for Conservative Party.
- Voters mostly voted based on the current national economic conditions. Around 1100 voters given rating of 3 & 4
- Voters mostly voted based on the current household economic conditions. Around 1000 voters given rating of 3 & 4
- Around 900~950 voters given rating 4 and above for Labour party leader i.e., Blair. Which tells that most of the voters opting for Labour leader.

- Voters rating for Conservative party leader is 2 & 4, where around 600 voters rated 2 and around 500 voters given 4. Most of the voters rated below 3 which shows the voters opting for conservative party leaders is less.
- Around 900 voters inclined towards the Eurpean Integration i.e. scale above 6.
- Around 1000 voters are fully aware of the respective parties' position on European Integration.
- We can see that most of the female and male voters opting for Labour party leader. Around 550 female voters opted for Labour party leader.
- Voters of all ages have voted for both Labour and Conservative party.
- Voters of all ages are aware of the respective party stand on European Integration

After completing EDA, we have built four models initially i.e. Logistic Regression, LDA, KNN and Naive Bayes models. After building the models we have compared all four models and can say that most of them performed well. Based on the performance metrics we can say that KNN model performed better with 86.15 train accuracy score and 82.24 test accuracy score with 63.54 vote share for Labour Party.

In continuation we have tuned the models using GridSearchCV and also built model based on Bagging and Boosting (Gradient & Adaptive). After building the models, we have compared all the seven models and can say that most of them performed well. Based on the performance metrics, we can say that Gradient Boosting performed better with 89.25 train accuracy score and 83.55 test accuracy score with 64.86 vote share for Labour Party.

Important Features (Top 4) for predicting exit polls based on the coefficients in Logistic Regression,

- Assessment of Conservative Leader (Hague)
- Assessment of Current National Economic Conditions
- Assessment of Labour Leader (Blair)
- Parties' positions on European Integration

<u>Recommendations:</u>

Based on our model predictions and insights we can recommend following,

- The Labour party supporters across all the ages are highly influenced to a certain extent by high positive perception about strong National Economic Condition and Household Economic Condition.
- Voters are strongly influenced with European Integration, around 900 voters given scale rating above 6 and by seeing the pair plot we can see that the stand of Labour Party on European Integration takes away voters to Conservative Party to certain extent. This point to rechecked again.
- Out of the total voters who have voted to Labour Party most of them are Female around 52% of them are female. Labour Party can try ways to attract Male voters to increase the vote bank.

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:

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

In this problem we will be using nltk library and load the speeches of above three Presidents mentioned. We will be applying basic text analytics on the speech to find the speech, stop words, counts & word cloud etc.

- Initially we will be using inaugural.raw function to load the speeches and will be storing into respective dataset.

2.1 Find the number of characters, words, and sentences for the mentioned documents

- Here we will be using length function to find out the number of characters in a text file

Number of Characters in Roosevelt Speech :	7571
Number of Characters in Kennedy Speech :	7618
Number of Characters in Nixon Speech :	9991

- By using split and length function we can find out the number of words in a text file

Number of Words in Roosevelt Speech :	1360
Number of Words in Kennedy Speech :	1390
Number of Words in Nixon Speech :	1819

- Importing sent_tokenize function from nltk tokenize library for getting the number of Sentences

Number of Sentences in Roosevelt Speech :	68
Number of Sentences in Kennedy Speech :	52
Number of Sentences in Nixon Speech :	68

2.2 Remove all the stop-words from all three speeches

- Importing Libraries which are required to remove stop words
- Defining a variable 'stop-words' which contains the list of punctuations from the string library & the English stop-words
- Converting all the words to lower case as stop words defined will be in lowercase
- Tokenize function would split the text into individual words
- Looping the text into the stop words and returning the words which are not in stop words
- Joining the words which are not in stop words and storing in new data
- Below is list of Stop words in respective President Speech,

Number of Stop Words in Roosevelt Speech :	903
Number of Stop Words in Kennedy Speech :	875
Number of Stop Words in Nixon Speech :	1206

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 stop words)

- Number of Words Occurring Frequently in Roosevelt Speech:

```
nation        12
know          10
spirit         9
democracy      9
life           9
people         7
america        7
freedom        6
years          6
's             5
```

We see the most occurring words in Roosevelt Speech. The top three words are

1. nation : 12 times
2. know : 10 times
3. spirit : 9 times

- Number of Words Occurring Frequently in Kennedy Speech:

```
sides       8
world       8
new         7
pledge      7
ask         5
shall       5
power       5
free        5
citizens    5
nations     5
dtype: int64
```

We see the most occurring words in Kennedy Speech. The top three words are

1. sides : 8 times
2. world : 8 times
3. new : 7 times

- <u>Number of Words Occurring Frequently in Nixon Speech:</u>

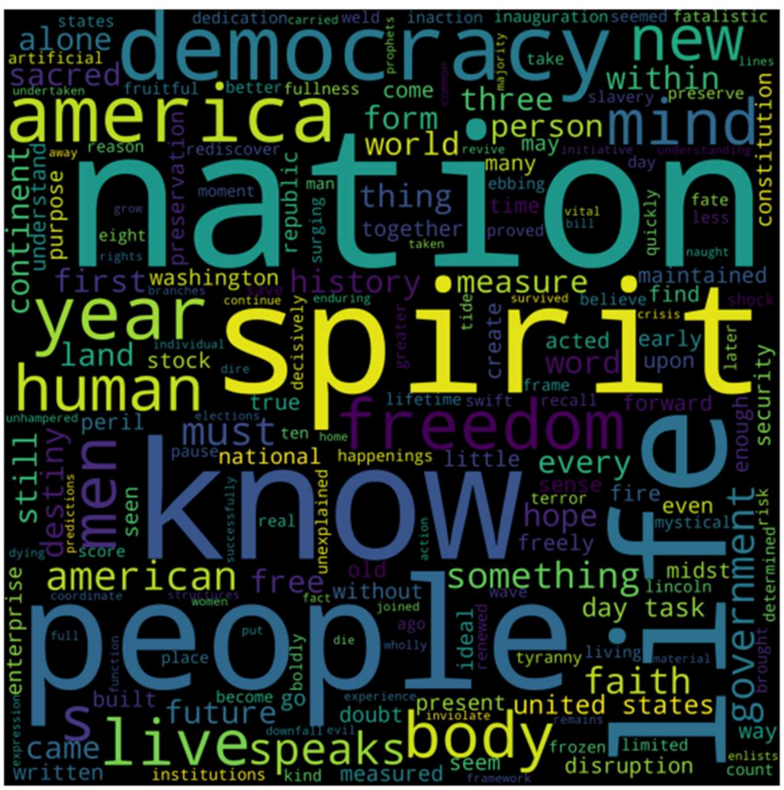
```
america           21
peace             19
world             18
new               15
's                14
nation            11
responsibility    11
government        10
home               9
great              9
dtype: int64
```

We see the most occurring words in Nixon Speech. The top three words are

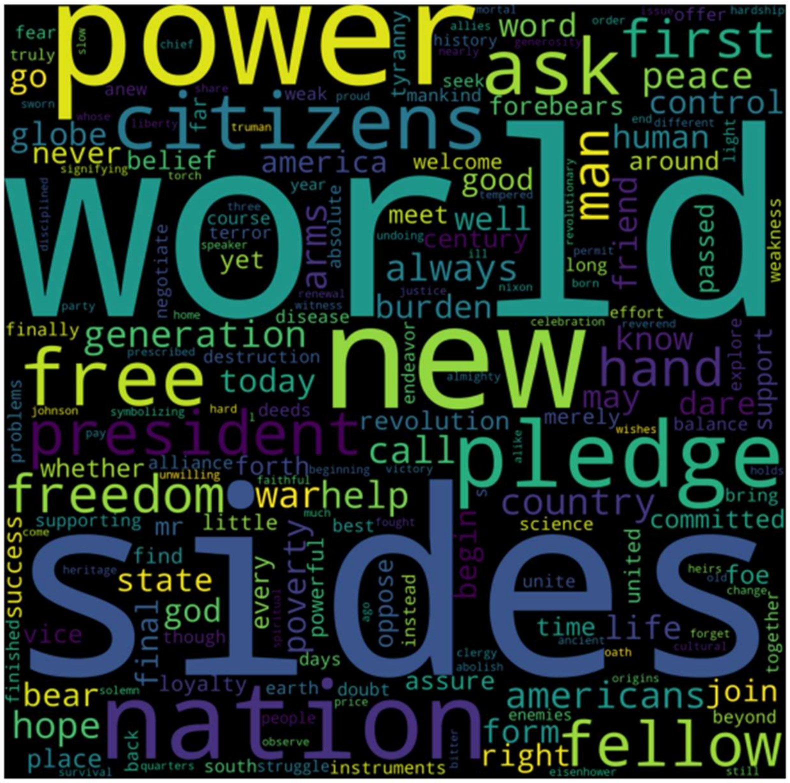
1. America : 21 times
2. peace : 19 times
3. world : 18 times

2.4 Plot the word cloud of each of the speeches of the variable. (after removing the stop words)

- Importing word cloud library for plotting the word cloud image
- **<u>Word Cloud for Roosevelt Speech:</u>**



- **Word Cloud for Kennedy Speech:**



- **Word Cloud for Nixon Speech:**