2023

MACHINE LEARNING GRADED PROJECT REPORT

DSBA

Girish Chadha
22/01/2023

Contents

List of Figures

List of Tables

EXECUTIVE SUMMARY

Problem 1

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.

Questions for Problem 1:

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

Solution: Reading the data

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

The dataset has 1525 rows and 10 columns

Table 1

Checking data types

```
<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
```

All the variables except vote and gender are int64 datatypes. when looking at the values in the dataset for the other variables, they all look like categorical columns except age. Removing the unwanted variable "Unnamed : 0"

There are 0 Missing Values in dataset

Describing dataset

	count	mean	std	min	25%	50%	75%	max
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

Table 2

Duplicate data

```
Total no of duplicate values = 8
```

	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
67	Labour	35	4	4	5	2	3	2	male
626	Labour	39	3	4	4	2	5	2	male
870	Labour	38	2	4	2	2	4	3	male
983	Conservative	74	4	3	2	4	8	2	female
1154	Conservative	53	3	4	2	2	6	0	female
1236	Labour	36	3	3	2	2	6	2	female
1244	Labour	29	4	4	4	2	2	2	female
1438	Labour	40	4	3	4	2	2	2	male

Duplicate data has been removed.

Unique values for categorical variables

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


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

Skewness

```
age                          0.139800
economic.cond.national      -0.238474
economic.cond.household     -0.144148
Blair                       -0.539514
Hague                        0.146191
Europe                      -0.141891
political.knowledge         -0.422928
dtype: float64
```

Inferences:

The head and tail of the dataset tells us that there are 2 main parties: 'Labour' and 'Conservative'.

With the problem statement we know that the target variable is 'Vote' from the dataset.

The dataset has 9 unique columns, out of which 2 are objects and 7 are integers.

From the descriptive statistics- youngest voter age is 24 years, 50% of the voters are of the age 53 years and the oldest voter is 93 years old

Labour party seems to have highest number of votes 1057 and the most voters are females 808.

Variables 'economic.cond.national', 'economic.cond.household', 'Blair', 'Hague', 'Europe' and 'political.knowledge' are numeric data variables.

Age and Hague seems to be positively skewed(Right tailed distribution) while other variables are negatively skewed(left tail distribution).

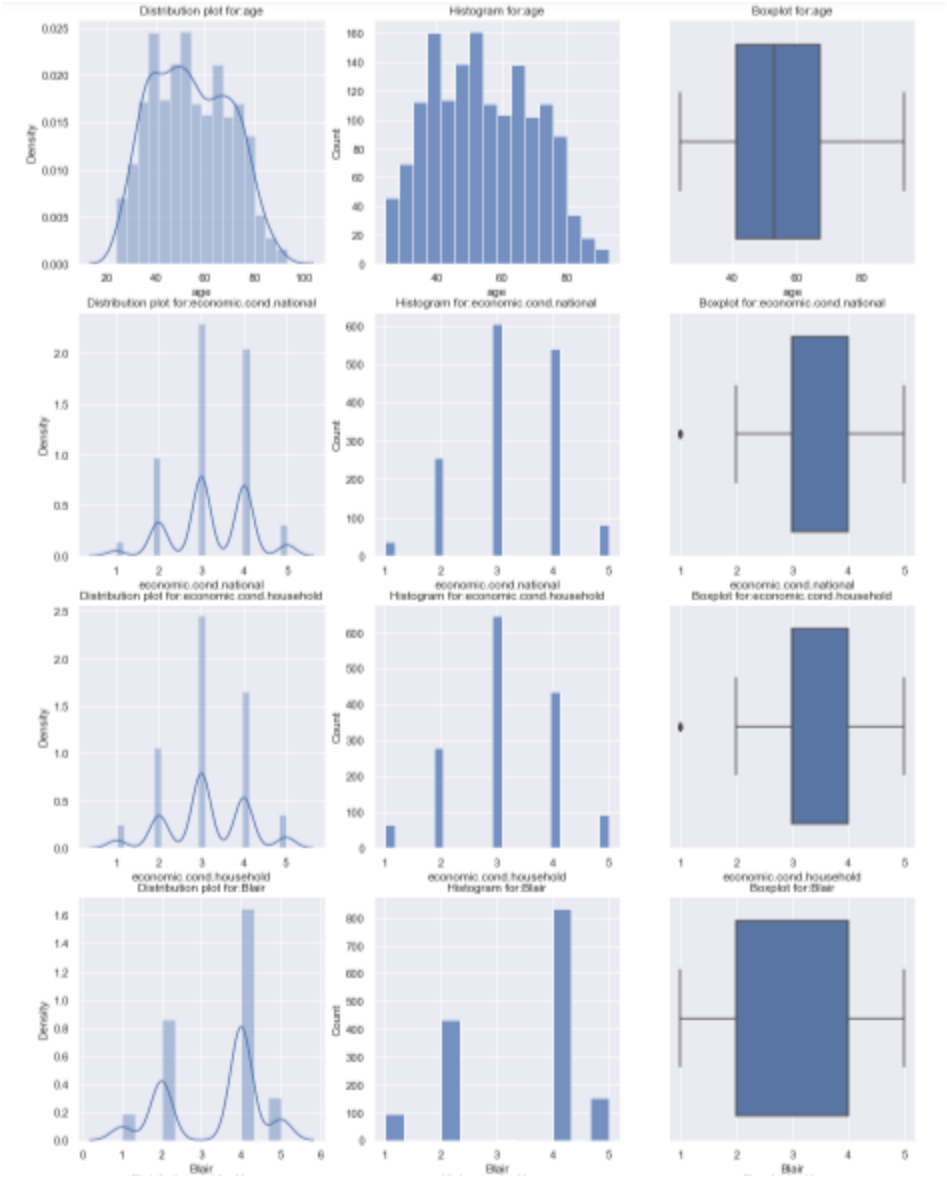
50% of Blair voters have voted 4 for assesment of labour leader & 50% of Hague voters have voted 2 for assesment of conservative leader.

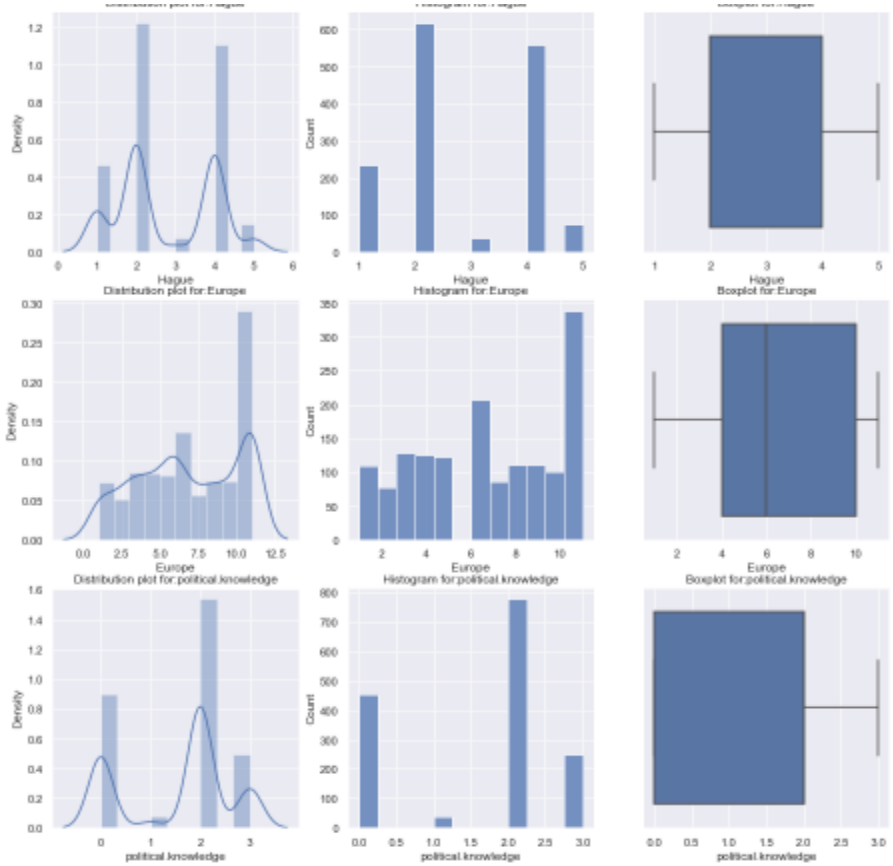
There were 8 duplicate records which were dropped as they do not add any value to our analysis

The percentage of votes are not balanced between the two parties where 69.68% of the voters voted for Labour party and only 30.32% of the voters voted for Conservative party.

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

UNIVARIATE ANALYSIS


Distribution plot for age
Density
age
Histogram for age
Count
age
Boxplot for age
age
Distribution plot for economic.cond.national
Density
economic.cond.national
Histogram for economic.cond.national
Count
economic.cond.national
Boxplot for economic.cond.national
economic.cond.national
Distribution plot for economic.cond.household
Density
economic.cond.household
Histogram for economic.cond.household
Count
economic.cond.household
Boxplot for economic.cond.household
economic.cond.household
Distribution plot for Blair
Density
Blair
Histogram for Blair
Count
Blair
Boxplot for Blair
Blair



Inference: We can infer that the Labour party is being favoured more by the voters The ratio of female to male is almost the same with female voters being more than male

Variable “economic.cond.national” and “economic.cond.household” show that most of the voters have rated these two scales as 3 and 4 which is a moderate rating

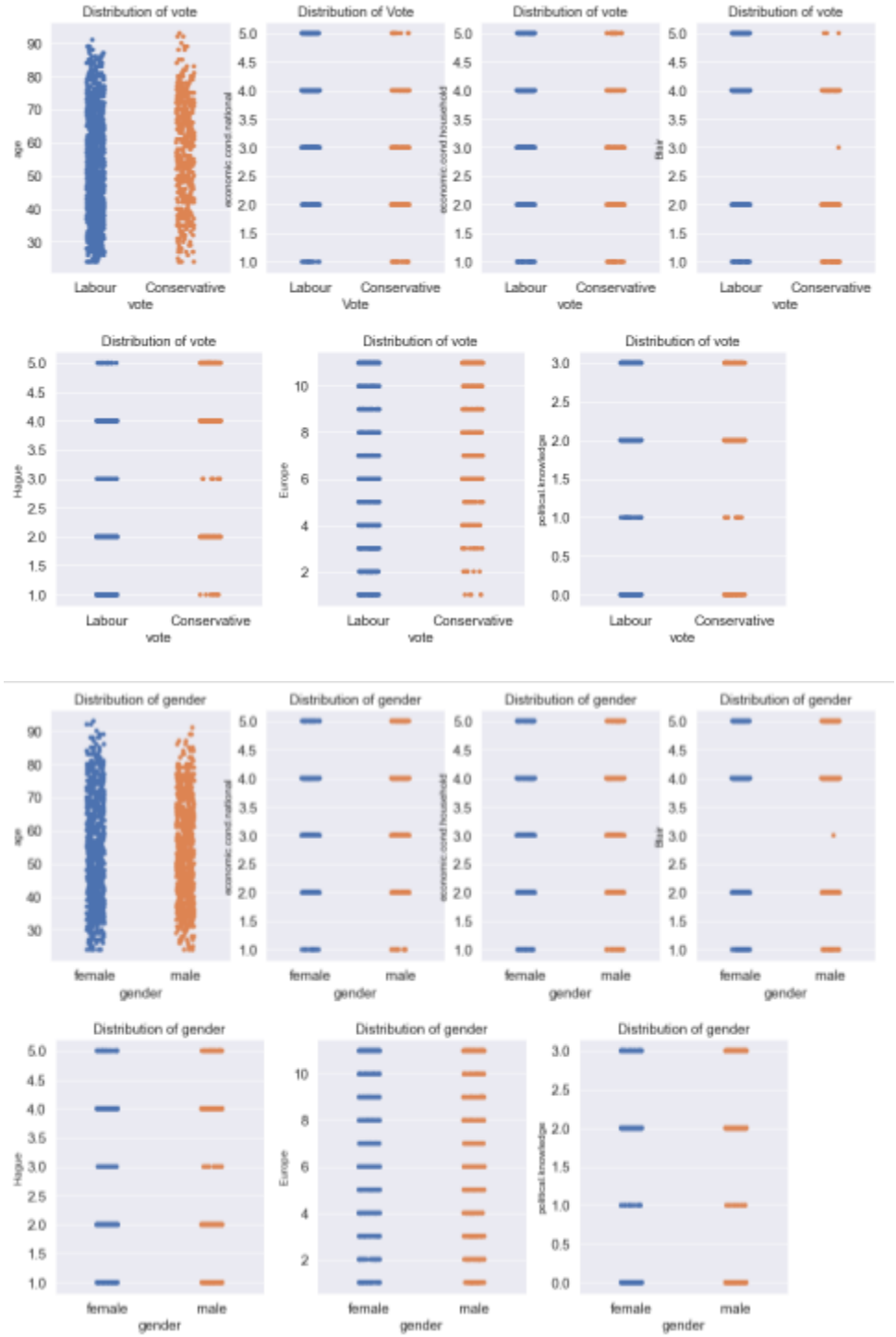
Most of the voters have voted “Blair” as 4 compared to “Hague” Many voters have rated “Hague” as 2 compared to “Blair”

There is a normal distribution in variable “age”. Most of the voters are found to be between the age of 40 to 80 years.

On a scale of 1 to 11, most of the voters have voted that the European integration is between 2 to 10 with maximum as 11. Hence, most of them have an inclination that the parties represent ‘Eurosceptic’ sentiment.

On a scale of 0 to 3, most of the voters have voted the Political knowledge to be 2 which is moderate.

BIVARIATE ANALYSIS



Inference:

As there are more voters for Labour party, the strip looks more dense for Labour party.Most of the voters above the age of 90 have voted for Conservative party.

Most of the voters have rated Labour party as 5 compared to Conservative Party for their assessment on current national economic conditions.

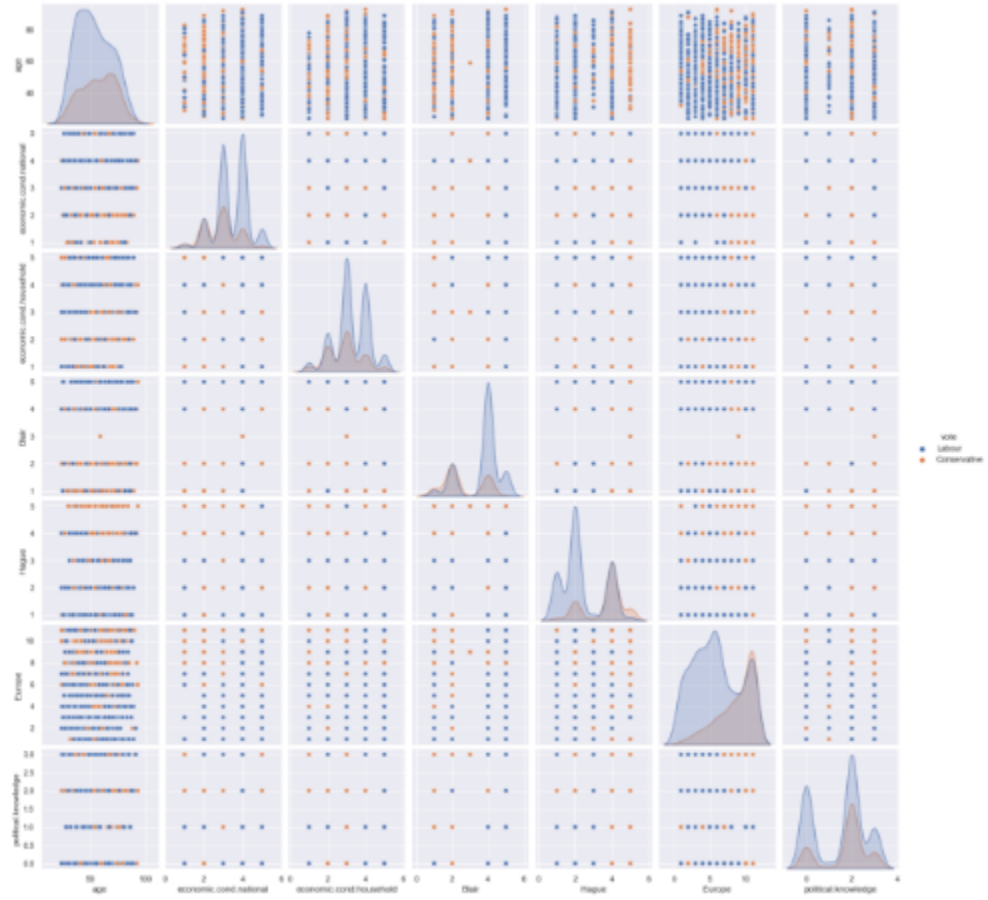
More voters have rated Labour party to be a 5 when it comes to an assessment on economic household conditions.

None of the female voters rated Blair as 3 but few male voters have rated him as 3.

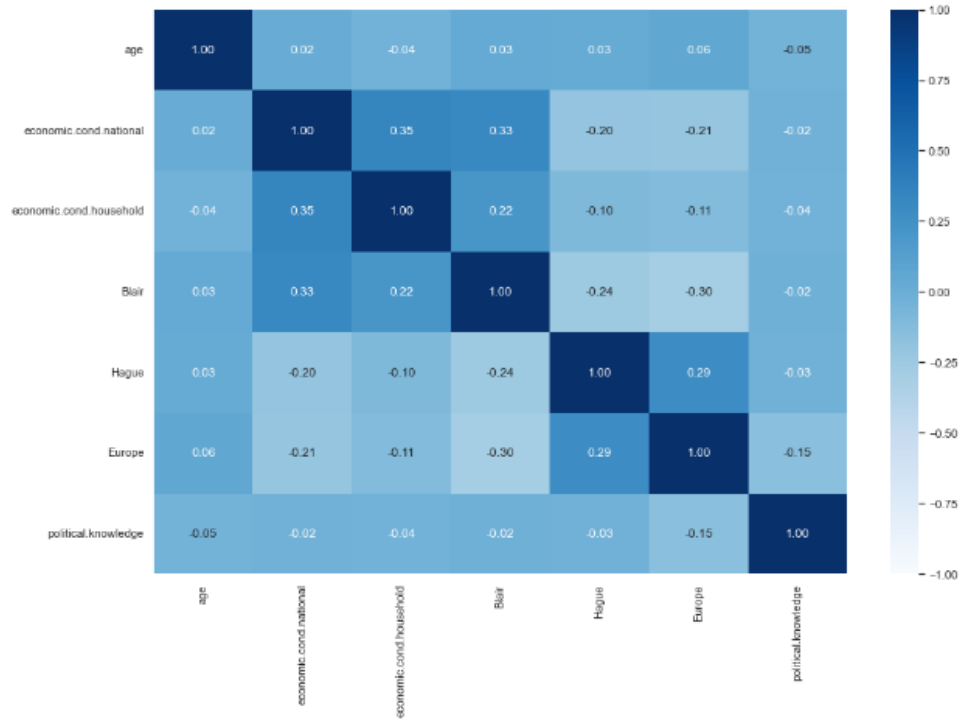
Many voters have rated Labour party between the scale of 2 to 5 compared to the Conservative Party.

On a scale of 0 to 3, Labour party seems to have more voters rating them as 1 compared to Conservative Party.

PAIRPLOT



CORRELATION COEFFICIENT HEATMAP



Inference:

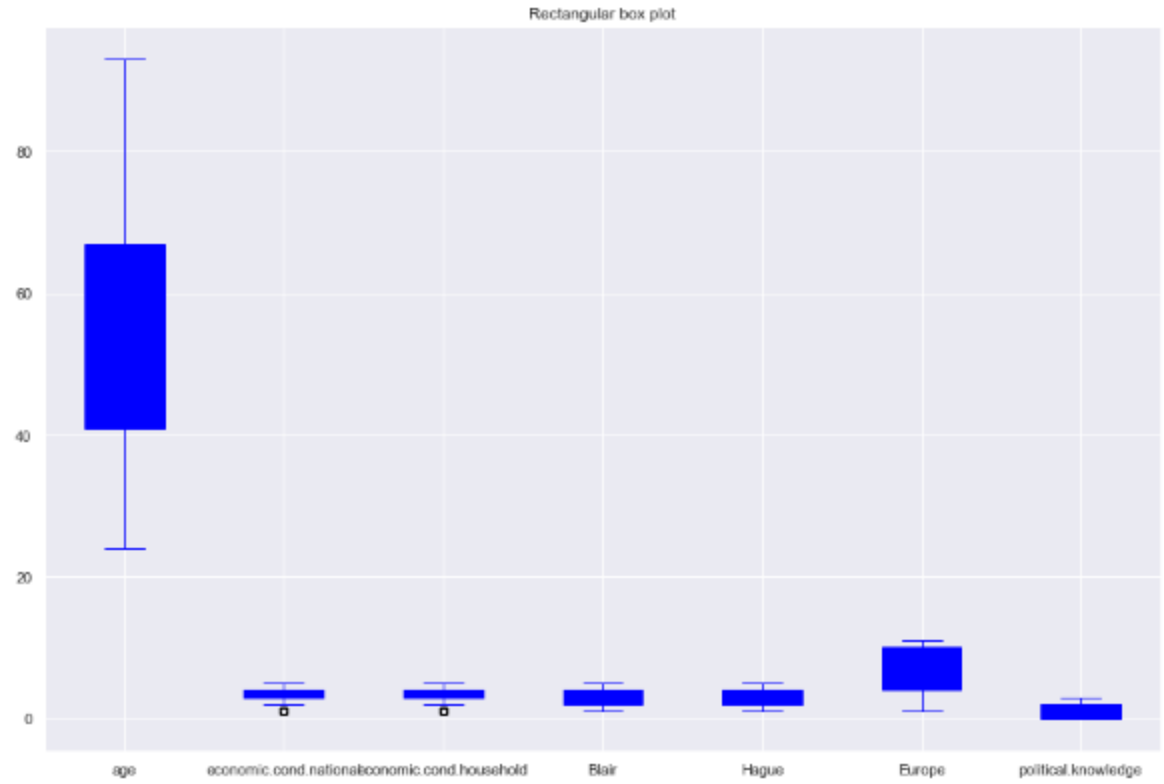
The heat map shows that there is no high correlation between any of the variables.

There is negative correlation between age and political knowledge and “economic.cond.household”

Variables “economic.cond.national” and “economic.cond.household” have the highest correlation of 0.35.

Pairplot shows no correlation between variables.

Check for Outliers



There are nearly no outliers in most of the numerical columns, only outlier is in economic.cond.national variable & economic.cond.household Variable . As 'Age' the only numeric variable is not having any outlier, let's proceed ahead.

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).

Encoding the data creating dummy variables

	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

Taking X as all independent variable and Y as dependent variable(vote_Labour).

Variance before scaling

```
age                        246.544655
economic.cond.national       0.777558
economic.cond.household      0.866890
Blair                        1.380089
Hague                        1.519005
Europe                      10.883687
political.knowledge          1.175961
gender_male                  0.249099
dtype: float64
```

Scaling is done on continuous variables in a dataset with different unit of measures. All variables are either categorical or ordinal except for variable “age”. For Logistic regression, LDA and Naïve Baye’s model we need not perform any scaling, however, for KNN it is necessary to scale the data, as it a distance-based algorithm (typically based on Euclidean distance). Let’s scale the variables.

Applying Z score

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
0	-0.716161	-0.278185	-0.148020	0.565802	-1.419969	-1.437338	0.423832	-0.936736
1	-1.162118	0.856242	0.926367	0.565802	1.014951	-0.527684	0.423832	1.067536
2	-1.225827	0.856242	0.926367	1.417312	-0.608329	-1.134120	0.423832	1.067536
3	-1.926617	0.856242	-1.222408	-1.137217	-1.419969	-0.830902	-1.421084	-0.936736
4	-0.843577	-1.412613	-1.222408	-1.988727	-1.419969	-0.224465	0.423832	1.067536

Variance after scaling

```
age                        1.00066
economic.cond.national     1.00066
economic.cond.household    1.00066
Blair                      1.00066
Hague                      1.00066
Europe                     1.00066
political.knowledge        1.00066
gender_male                1.00066
dtype: float64
```

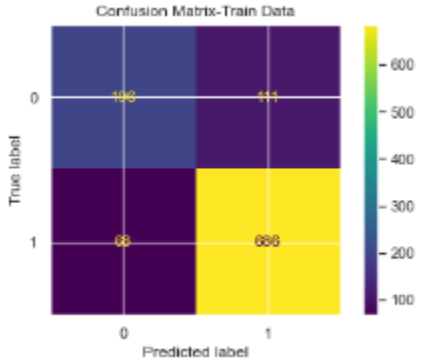
Now variance is same.

Splitting the data into test and train sets in 30:70 ratio taking random state as 1.

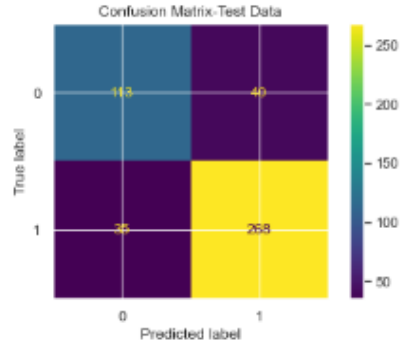
1.4 Apply Logistic Regression and LDA (linear discriminant analysis).
LOGISTIC REGRESSION MODEL

```
LogisticRegression(max_iter=10000, n_jobs=2, penalty='none', solver='newton-cg',
                   verbose=True)
```

```
The Logistic Regression Model Score on train data set is 0.831
[[196 111]
 [ 68 686]]
              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
```




```
The Logistic Regression Model Score on test data set is 0.836
[[113  40]
 [ 35 268]]
              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
```



Training Data and Test Data Comparison & Inference

The accuracy of model in training set is 0.83 and on testing set is 0.84, which is good and very close to each other. The recall of Conservative party is better on Testing data whereas the recall of Labour party is better on Training data Overall it is a good model and there is no over fitting found.

Applying GridSearchCV for Logistic Regression(Model Tuning)

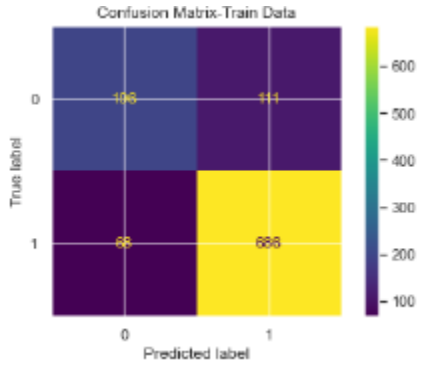
```
GridSearchCV(cv=RepeatedStratifiedKFold(n_repeats=3, n_splits=10, random_state=1),
             estimator=LogisticRegression(max_iter=10000, n_jobs=2,
                                          penalty='none', solver='newton-cg',
                                          verbose=True),
             n_jobs=2,
             param_grid={'max_iter': [10000, 5000, 15000],
                         'penalty': ['l2', 'none', 'l1', 'elasticnet'],
                         'solver': ['liblinear', 'lbfgs', 'newton-cg'],
                         'tol': [0.0001, 1e-05]})
```

```
{'max_iter': 10000, 'penalty': 'l2', 'solver': 'lbfgs', 'tol': 0.0001}
```

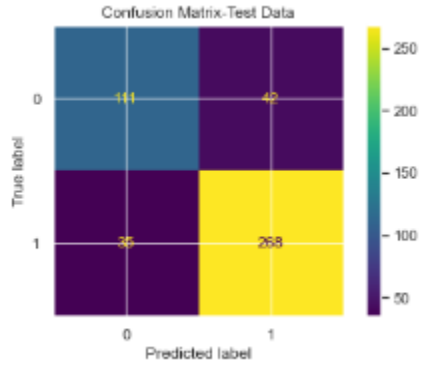
```
LogisticRegression(max_iter=10000, n_jobs=2, verbose=True)
```

Model Evaluation for Train Data

The Best Logistic Regression Model Score on train data set post tuning is 0.831



The Best Logistic Regression Model Score on test data set post tuning is 0.831



```
Classification Report of the training data:

              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

Classification Report of the test data:

              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
```

Difference in accuracy of train & test set : 0.0001508838070671814

The difference is very less hence the LR tuned model is a good fit.

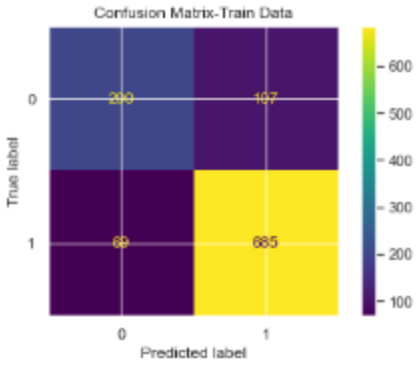
LINEAR DISCRIMINANT ANALYSIS

LinearDiscriminantAnalysis()

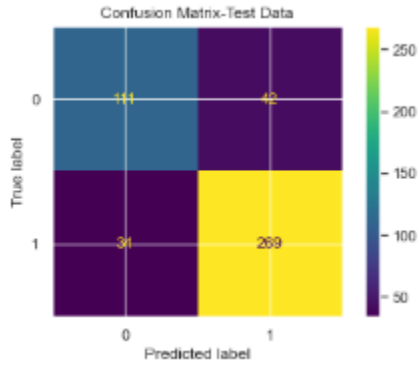
```
The LDA Model Score on train data set is 0.834
[[200 107]
 [ 69 685]]
              precision    recall  f1-score   support

           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

The LDA Model Score on test data set is 0.833
[[111  42]
 [ 34 269]]
              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
```

Applying GridSearchCV for LDA(Model tuning)

The Best LDA Model Score on train data set post tuning is 0.834



The Best LDA Model Score on test data set post tuning is 0.833



Classification Report of the training data:

	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

Classification Report of the test data:

	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

Difference in accuracy of train & test set: 0.0007854225573358242
The difference is very less hence this lda tuned model is a good fit. No overfitting or underfitting.

The accuracy of model in training set and testing set is the same which is 0.83.

The recall of Conservative party is better on Testing data whereas the recall of Labour party is better on Training data Overall the model is performing well.

Comparison of two models:

While comparing both these models, we find both results are almost same, but LDA works better since the recall with LDA is slightly better on Testing data.

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

Good KNN performance usually requires preprocessing of data to make all variables similarly scaled and centered. We have already done that to our dataset.

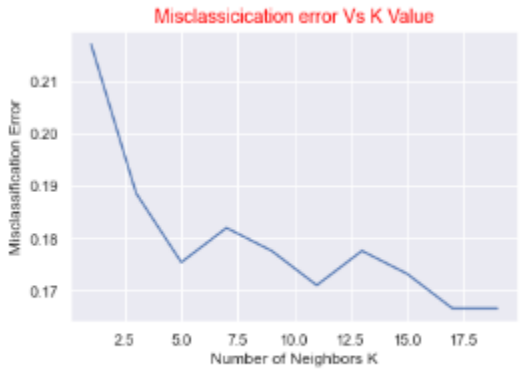
KNeighborsClassifier()

```
The KNN Model Score on Train data 0.856
[[218  89]
 [ 64 690]]
              precision    recall  f1-score   support

           0       0.77      0.71      0.74       307
           1       0.89      0.92      0.90       754

    accuracy                           0.86      1061
   macro avg       0.83      0.81      0.82      1061
weighted avg       0.85      0.86      0.85      1061


The KNN Model Score on Test data 0.825
[[105  48]
 [ 32 271]]
              precision    recall  f1-score   support

           0       0.77      0.69      0.72       153
           1       0.85      0.89      0.87       303

    accuracy                           0.82       456
   macro avg       0.81      0.79      0.80       456
weighted avg       0.82      0.82      0.82       456
```

Run the KNN with no of neighbours to be 1,3,5..19 and *Find the optimal number of neighbours from K=1,3,5,7....19 using the Mis classification error Misclassification error (MCE) = 1 - Test accuracy score. Calculated MCE for each model with neighbours = 1,3,5...19 and find the model with lowest MCE

```
[0.2171052631578947,
 0.1885964912280702,
 0.17543859649122806,
 0.18201754385964908,
 0.17763157894736847,
 0.17105263157894735,
 0.17763157894736847,
 0.17324561403508776,
 0.16666666666666663,
 0.16666666666666663]
```

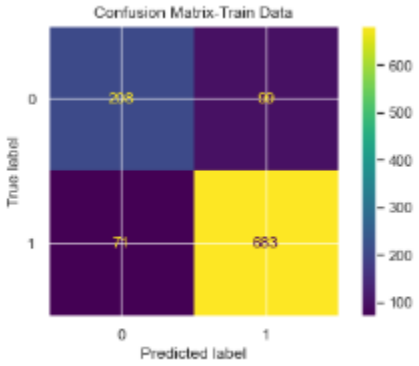
Plot misclassification error vs k (with k value on X-axis)



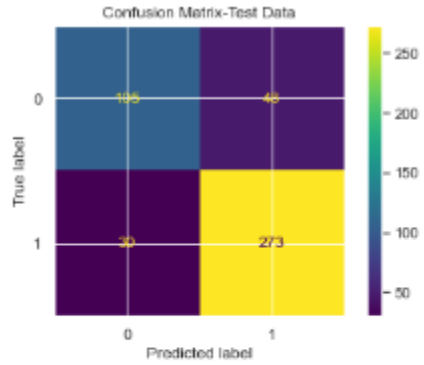
For K = 11 it is giving the best test accuracy. We will build the model with k=11

Performance Matrix of KNN New Model on train & test data set

```
The KNN Model Score on Train data 0.840
[[208  99]
 [ 71 683]]
              precision    recall  f1-score   support

           0       0.75      0.68      0.71       307
           1       0.87      0.91      0.89       754

    accuracy                           0.84      1061
   macro avg       0.81      0.79      0.80      1061
weighted avg       0.84      0.84      0.84      1061
```




```
The KNN Model Score on Test data 0.829
[[105  48]
 [ 30 273]]
              precision    recall  f1-score   support

           0       0.78      0.69      0.73       153
           1       0.85      0.90      0.88       303

    accuracy                           0.83       456
   macro avg       0.81      0.79      0.80       456
weighted avg       0.83      0.83      0.83       456
```



Accuracy on Train data is 0.84 & on test data is 0.83.

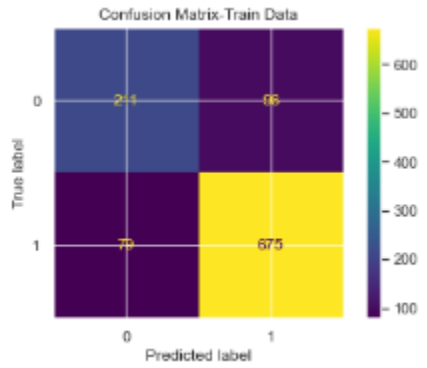
Training and Testing results shows that the model is excellent with good precision and recall values.

This KNN model have good accuracy and recall values.

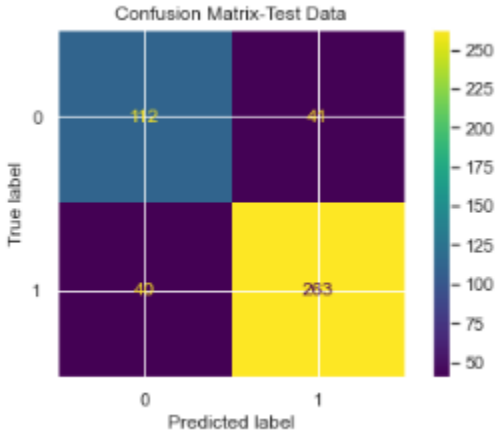
Naive Bayes

GaussianNB()

```
The Naive Bayes Model Score on train data is 0.835
[[211  96]
 [ 79 675]]
              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
```

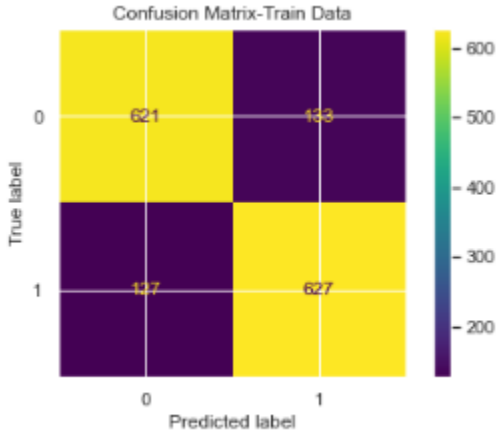



```
The Naive Bayes Model Score on test data is 0.822
[[112  41]
 [ 40 263]]
              precision    recall  f1-score   support

           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
```

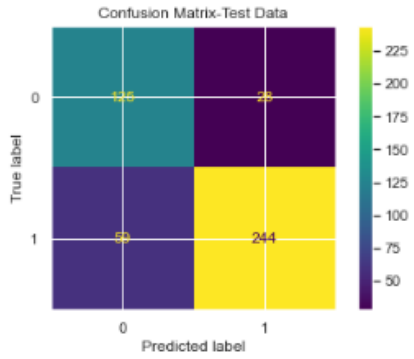


Naive Bayes with SMOTE

```
The SMOTE Model Score for train data set is 0.828
[[621 133]
 [127 627]]
              precision    recall  f1-score   support

           0       0.83      0.82      0.83       754
           1       0.82      0.83      0.83       754

    accuracy                           0.83      1508
   macro avg       0.83      0.83      0.83      1508
weighted avg       0.83      0.83      0.83      1508
```




```
The SMOTE Model Score for test data set is 0.809
[[125  28]
 [ 59 244]]
              precision    recall  f1-score   support

           0       0.68      0.82      0.74       153
           1       0.90      0.81      0.85       303

    accuracy                           0.81       456
   macro avg       0.79      0.81      0.80       456
weighted avg       0.82      0.81      0.81       456
```



The SMOTE model accuracy on train data is 0.83 & on test data is 0.81. No overfitting or underfitting.

Training and Testing results shows that the model neither overfitting nor underfitting.

The Naive Bayes model also performs well with better accuracy and recall values.

Even though NB and KNN have same Train and Test accuracy. Based on their recallvalue in test dataset it is evident that KNN performs better than Naive Bayes.

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

MODEL TUNING

Logistic regression model & LDA model Tuning has already been done using gridsearch. Results are as follows -

LOGISTIC REGRESSION MODEL TUNED

```
Classification Report of the training data:

              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

Classification Report of the test data:

              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
```

LDA MODEL TUNED

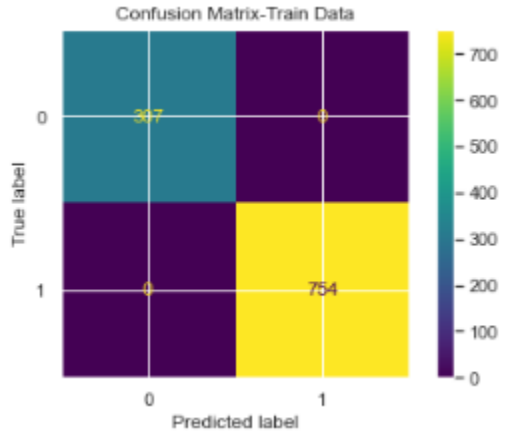
```
Classification Report of the training data:

              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

Classification Report of the test data:

              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
```

The difference is very less hence this lda tuned model is a good & better fit. No overfitting or underfitting.
KNN model with k=11 is a better & good fit.
Naive Bayes with SMOTE model is a better & good fit.

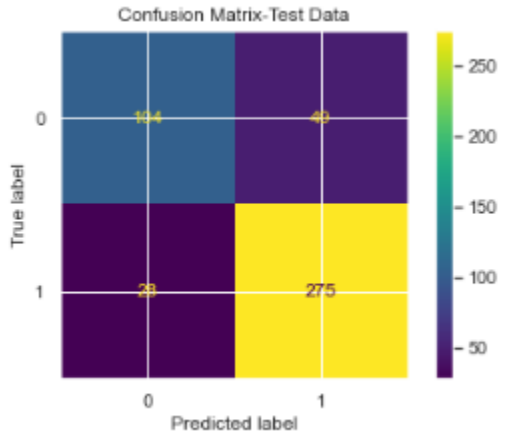
RANDOM FOREST

RandomForestClassifier(random_state=1)

```
The random Forest Score on train data is 1.00
[[307   0]
 [  0 754]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       307
           1       1.00      1.00      1.00       754

    accuracy                           1.00      1061
   macro avg       1.00      1.00      1.00      1061
weighted avg       1.00      1.00      1.00      1061
```



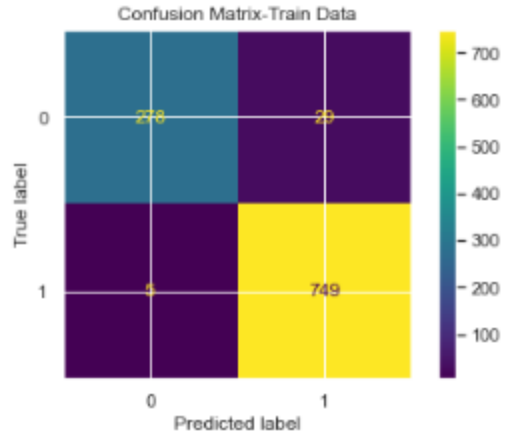
```
The random Forest Score on test data is 0.831
[[104  49]
 [ 28 275]]
              precision    recall  f1-score   support

           0       0.79      0.68      0.73       153
           1       0.85      0.91      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.79      0.80       456
weighted avg       0.83      0.83      0.83       456
```



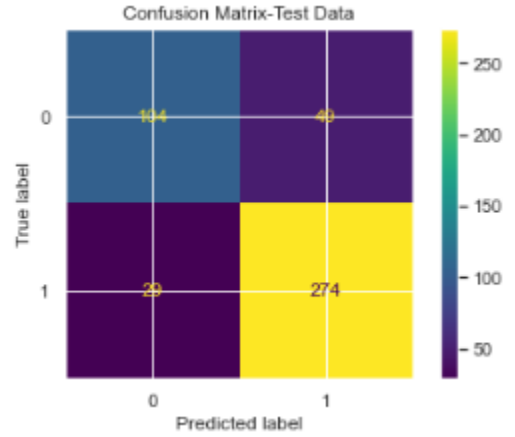
There is a lot of difference(0.16) between train and test set. This seems to be a case of overfitting model.

Bagging

```
BaggingClassifier(base_estimator=RandomForestClassifier(), n_estimators=100,
                  random_state=1)

The Bagging Model Score for train data set is 0.97
[[278  29]
 [  5 749]]
              precision    recall  f1-score   support

           0       0.98      0.91      0.94       307
           1       0.96      0.99      0.98       754

    accuracy                           0.97      1061
   macro avg       0.97      0.95      0.96      1061
weighted avg       0.97      0.97      0.97      1061
```




```
The Bagging Model Score for test data set is 0.83
[[104  49]
 [ 29 274]]
              precision    recall  f1-score   support

           0       0.78      0.68      0.73       153
           1       0.85      0.90      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.79      0.80       456
weighted avg       0.83      0.83      0.83       456
```



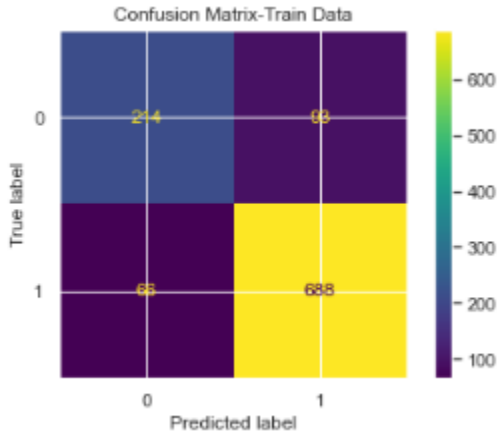
Random Forest Bagging has reduced the difference to 0.13 of model score of train and test data. Hence it's better fit.

Boosting

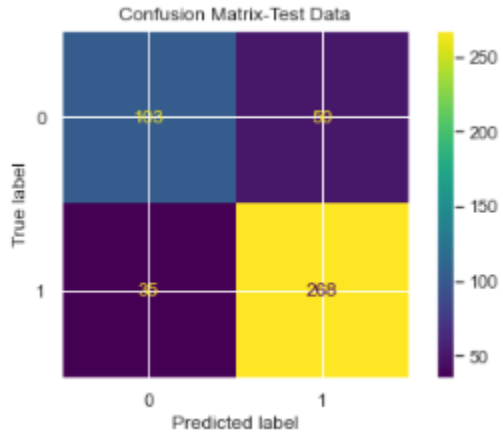
Ada Boost

AdaBoostClassifier(n_estimators=100, random_state=1)

```
The ADA boost Model Score for train data set is 0.850
[[214  93]
 [ 66 688]]
              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
```




```
The ADA boost Model Score for test data set is 0.814
[[103  50]
 [ 35 268]]
              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
```

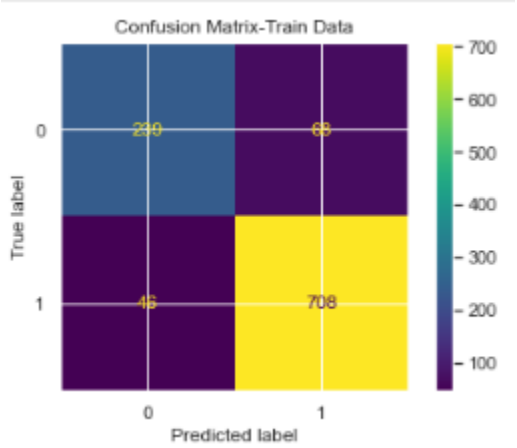


Less difference(0.03) hence it is not a case of overfitting or underfitting

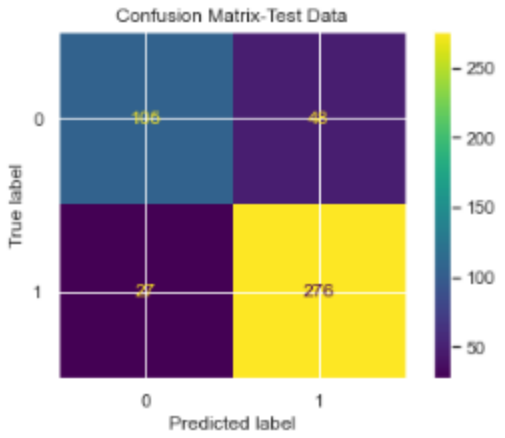
Gradient Boosting

GradientBoostingClassifier(random_state=1)

```
The Gradient Boosting Score for train data set is 0.89
[[239  68]
 [ 46 708]]
              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
```




```
The Gradient Boosting Score for train data set is 0.84
[[105  48]
 [ 27 276]]
              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
```



Less difference(0.05) shows good fit. Hence it's not a case of overfitting or underfitting.

ADA BOOST SEEMS TO PERFORM BETTER than gradient boosting.

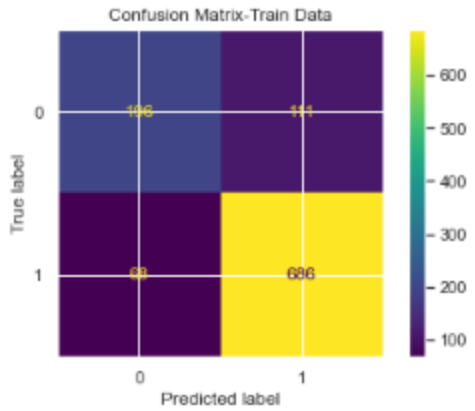
1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model, classification report (4 pts) Final Model - Compare and comment on all models on the basis of the performance metrics in a structured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

Performance Matrix of Logistic Regression on train & test data set

Classification Report of the training data:

	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

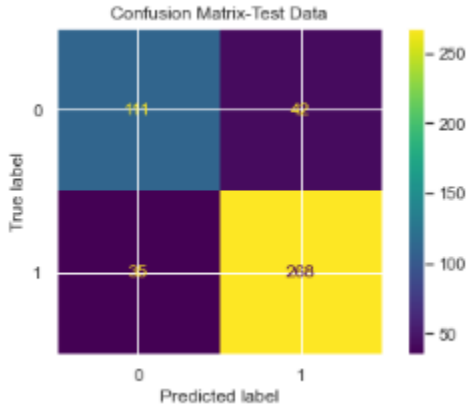
The Best Logistic Regression Model Score on train data set post tuning is 0.831



Classification Report of the test data:

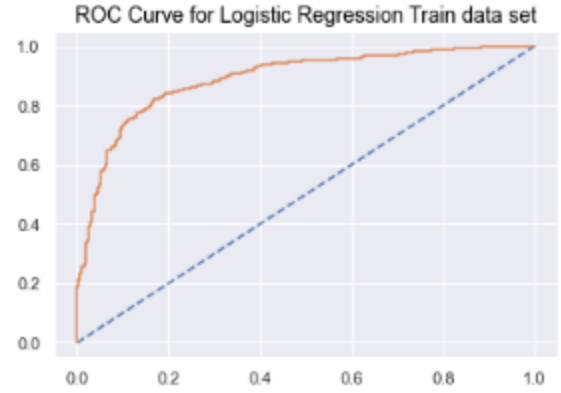
	precision	recall	f1-score	support
0	0.76	0.73	0.74	153
1	0.86	0.88	0.87	303
accuracy			0.83	456
macro avg	0.81	0.80	0.81	456
weighted avg	0.83	0.83	0.83	456

The Best Logistic Regression Model Score on test data set post tuning is 0.831

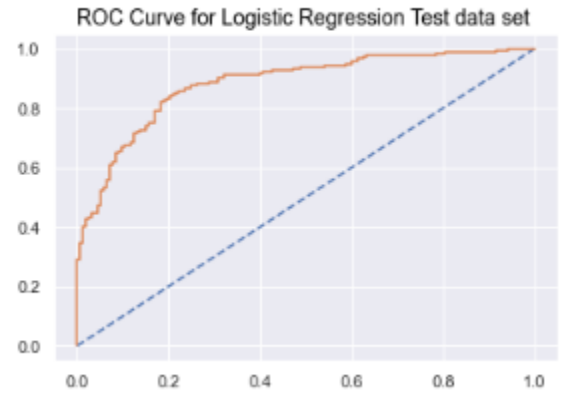


ROC_AUC Curve for Logistic Regression on train & test data set

```
The ROC_AUC score for Logistic Regression Train data set: 0.890
```




```
The ROC_AUC score for Logistic Regression Test data set : 0.883
```

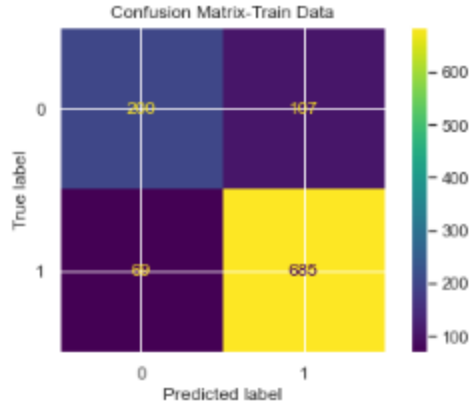


Performance Matrix of LDA (linear discriminant analysis) on train & test data set

Classification Report of the training data:

	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

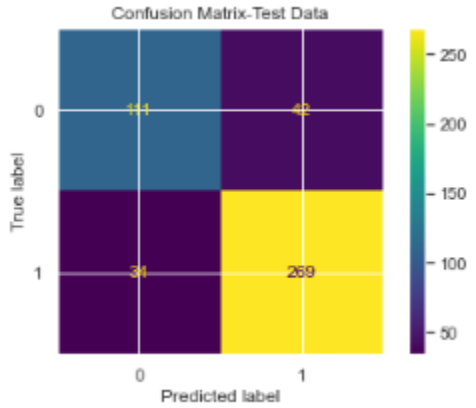
The Best LDA Model Score on train data set post tuning is 0.834



Classification Report of the test data:

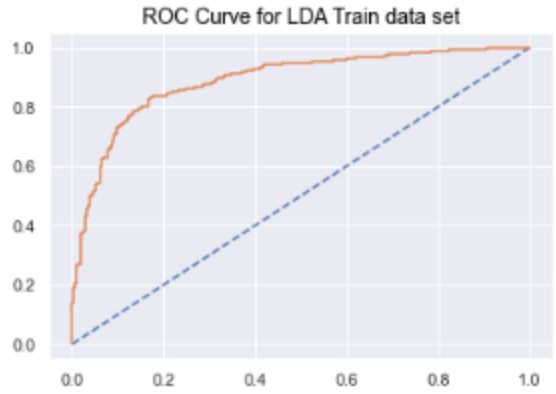
	precision	recall	f1-score	support
0	0.77	0.73	0.74	153
1	0.86	0.89	0.88	303
accuracy			0.83	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.83	0.83	456

The Best LDA Model Score on test data set post tuning is 0.833

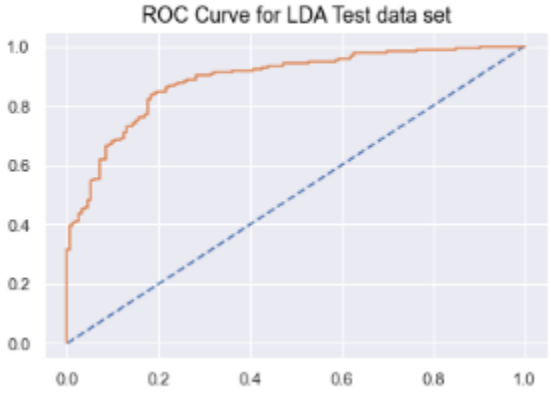


ROC_AUC Curve for LDA (linear discriminant analysis) on train & test data set

```
The ROC_AUC score for LDA Train data set 0.89
```

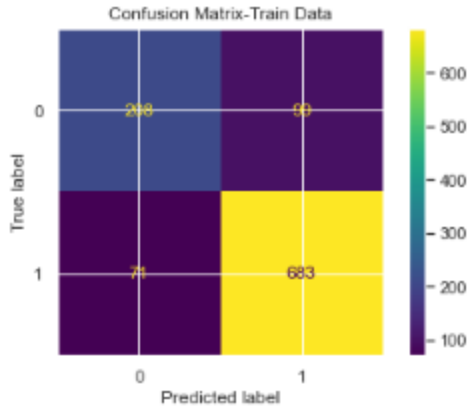



```
AUC: 0.888
The ROC_AUC score for LDA Test data set is ' 0.888
```

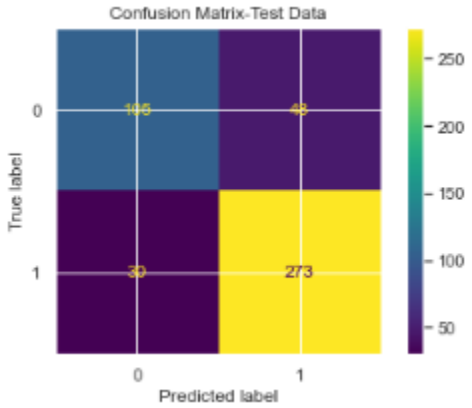


Performance Matrix of KNN on train & test data set

```
The KNN Model Score on Train data 0.840
[[208  99]
 [ 71 683]]
              precision    recall  f1-score   support

           0       0.75      0.68      0.71       307
           1       0.87      0.91      0.89       754

    accuracy                           0.84      1061
   macro avg       0.81      0.79      0.80      1061
weighted avg       0.84      0.84      0.84      1061
```

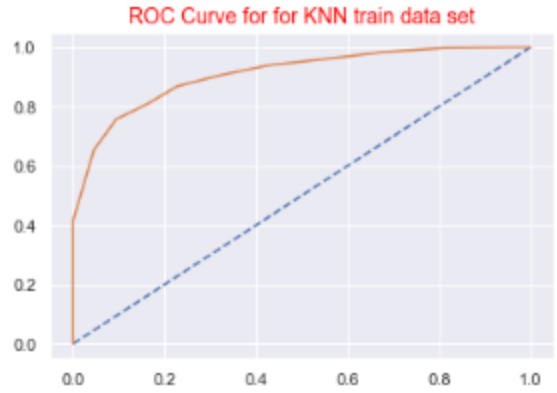



```
The KNN Model Score on Test data 0.829
[[105  48]
 [ 30 273]]
              precision    recall  f1-score   support

           0       0.78      0.69      0.73       153
           1       0.85      0.90      0.88       303

    accuracy                           0.83       456
   macro avg       0.81      0.79      0.80       456
weighted avg       0.83      0.83      0.83       456
```

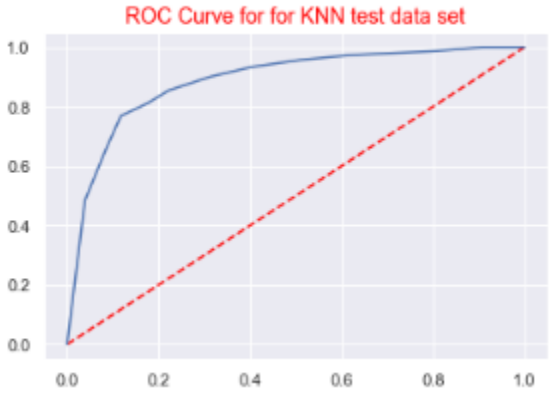


ROC_AUC Curve for KNN on train & test data set

```
The ROC_AUC score for KNN train data set 0.91
```

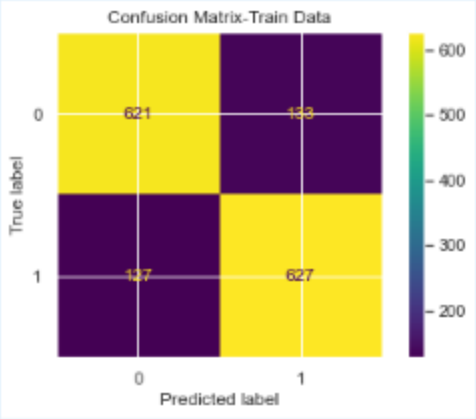



```
The ROC_AUC score for KNN train data set 0.89
```

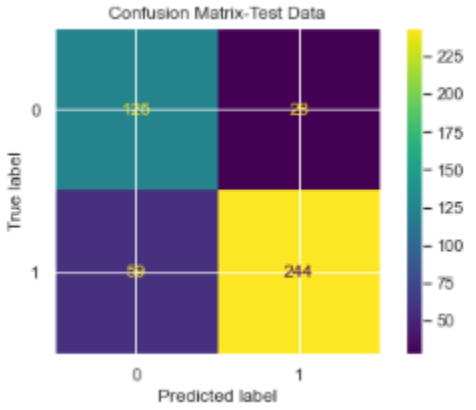


Performance Matrix of Naive Bayes with SMOTE on train & test data set

```
The SMOTE Model Score for train data set is 0.828
[[621 133]
 [127 627]]
              precision    recall  f1-score   support

           0       0.83      0.82      0.83       754
           1       0.82      0.83      0.83       754

    accuracy                           0.83      1508
   macro avg       0.83      0.83      0.83      1508
weighted avg       0.83      0.83      0.83      1508
```

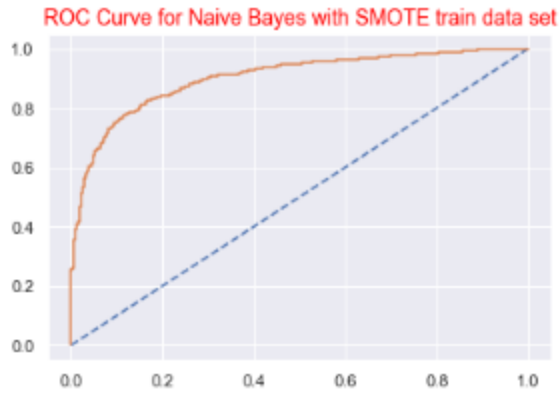



```
The SMOTE Model Score for test data set is 0.809
[[125  28]
 [ 59 244]]
              precision    recall  f1-score   support

           0       0.68      0.82      0.74       153
           1       0.90      0.81      0.85       303

    accuracy                           0.81       456
   macro avg       0.79      0.81      0.80       456
weighted avg       0.82      0.81      0.81       456
```

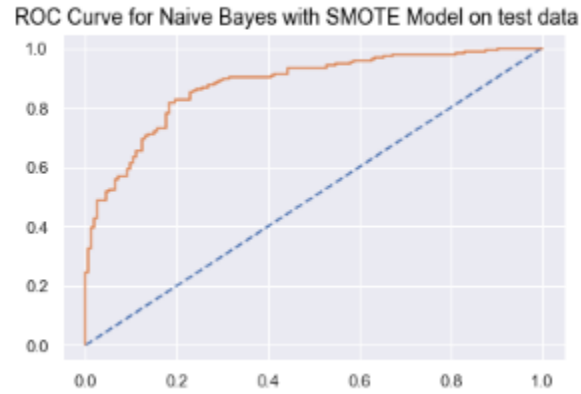


ROC_AUC Curve for Naive Bayes with SMOTE Model on train & test data set

```
The ROC_AUC score for Naive Bayes with SMOTE train data set 0.90
```

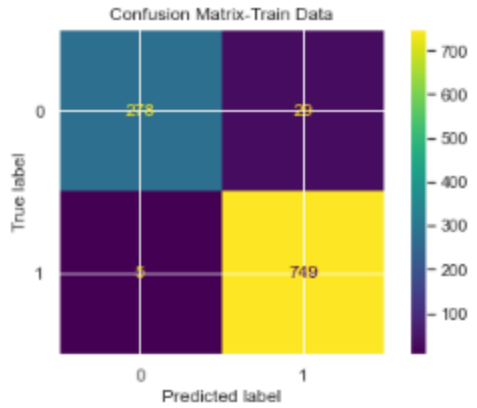



```
The ROC_AUC score for Naive Bayes with SMOTE Model on test data set 0.88
```

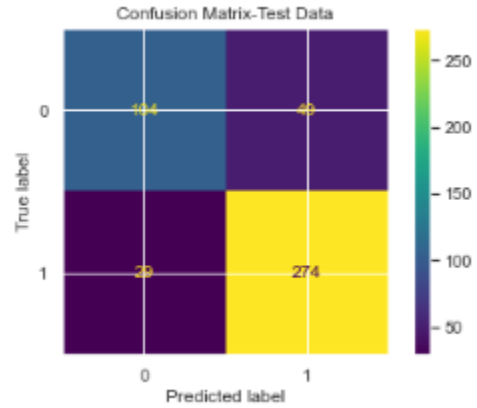


Performance Matrix of Bagging on train & test data set

```
The Bagging Model Score for train data set is 0.97
[[278  29]
 [  5 749]]
              precision    recall  f1-score   support

           0       0.98      0.91      0.94       307
           1       0.96      0.99      0.98       754

    accuracy                           0.97      1061
   macro avg       0.97      0.95      0.96      1061
weighted avg       0.97      0.97      0.97      1061
```



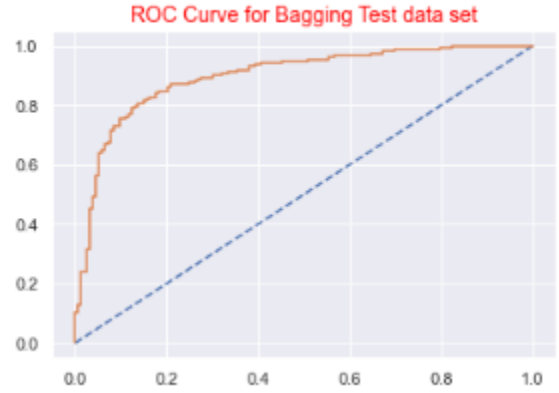

```
The Bagging Model Score for test data set is 0.83
[[104  49]
 [ 29 274]]
              precision    recall  f1-score   support

           0       0.78      0.68      0.73       153
           1       0.85      0.90      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.79      0.80       456
weighted avg       0.83      0.83      0.83       456
```



ROC_AUC Curve for Bagging on train & test data set

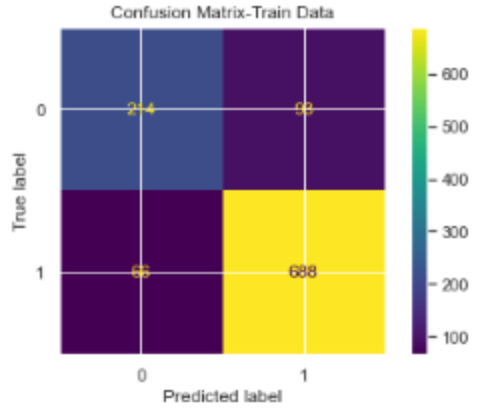
The ROC_AUC score for Bagging train data set 1.00

The AUC_ROC score for Bagging test data set 0.90

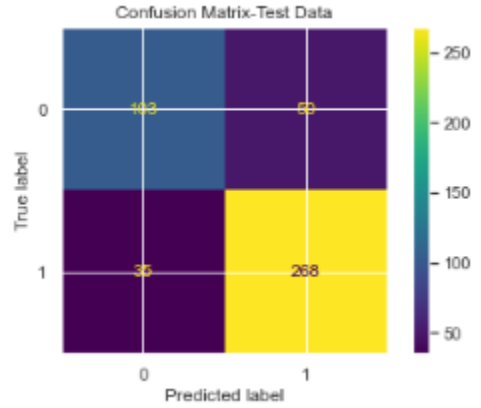


Performance Matrix of Ada Boost on train data set

```
The ADA boost Model Score for train data set is 0.850
[[214  93]
 [ 66 688]]
              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
```

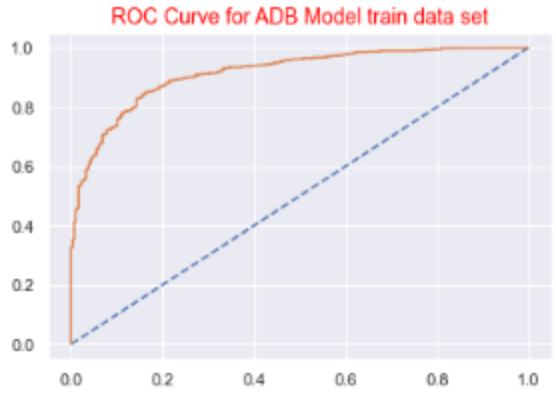



```
The ADA boost Model Score for test data set is 0.814
[[103  50]
 [ 35 268]]
              precision    recall  f1-score   support

           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
```

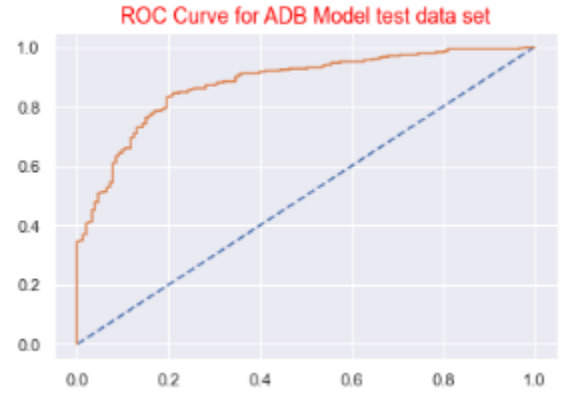


ROC_AUC Curve for Ada Boost on train & test data set

The AUC_ROC score for ADB Model train data set 0.91



The AUC_ROC score for ADB Model test data set 0.88



Comparison of Different Models

The Logistic Regression Model Score Post Tuning on train data set is 0.831

The Logistic Regression Model Score Post Tuning on test data set is 0.831

The LDA Model Score Post Tuning on train data set is 0.834

The LDA Model Score Post Tuning on test data set is 0.833

The KNN Model Score Post Tuning on Train data 0.840

The KNN Model Score Post Tuning on Test data 0.829

The Naive Bayes Model Score Post Tuning on train data is 0.828

The Naive Bayes Model Score Post Tuning on test data is 0.809

The Bagging Model Score Post Tuning on Train data 0.968

The Bagging Model Score Post Tuning on Test data 0.829

The AdaBoost Model Score on Train data 0.850

The AdaBoost Model Score on Test data 0.814

The GradientBoost Model Score on Train data 0.893

The GradientBoost Model Score on Test data 0.836
The basis on which models are evaluated are known as performance metrics. The metrics on which the model will be evaluated are

• Accuracy

• AUC

• Recall

• Precision

• F1-Score

From the above models,

• Basis on the accuracy – GradientBoost performed better than others.

• Basis on the AUC score – Logistic Regression performed better than others.

• Basis on recall – Bagging performed slightly better than others.

• Basis on Precision – Naïve Bayes performed slightly better than others.

• Basis on F1-Score – Logistic Regression along with some others performed well.

All the models performed well with slight difference ranging from (1-5%). Bagging seems to have large error in predicting test data as accuracy is too high for train data than test data. From the above, on the basis of accuracy logistic regression performed better than others.

Observations From above it can be observed that using SMOTE didn't increase the performance of the models. Overall models without SMOTE performed well for data. Thus, there is no use of applying SMOTE here. As for the scaled models, scaling only improved the performance of the distance based on algorithm for others it slightly decreased the performance overall. Here, only KNN for scaled data model performed slightly well.

Best Optimized model – On the basis of all the comparisons and performance metrics "Logistic Regression" with scaling performed the best out of all. Other models are having high error in accuracy in train and test data.

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

Insights and Recommendations

Accuracy on all the models appears to be similar to each other on train and test sets.

AUC is exactly same for both the train and the test data for all the models.

From the summary of the confusion matrix, we can see that the actual and the predicted data are very close to each other. This is the reflection of the right fit model.

F1 score for both the models are almost same for both the models on train and test data.

Model tuning on random forest models gives the better results than other models. But however, bagging on Random forest performs well on both the train and test sets with good accuracy and overall other performance measures as well.

Boosting techniques has also shown a good performance results.

By comparing the performance metrics, we can conclude the following:

Logistic Regression, LDA, KNN and Gausion Naïve Bayes are good models because they work well on both Training and Testing data with model accuracy similar across both training and testing data.

However, LDA has better accuracy and recall and f1-score.

Gradient Boosting and Bagging using Random Forest is not a good model because it is overfitting on training data and doesn't perform well on testing data

Comparing all the Models we see that Logistic Regression, LDA, KNN and Gradient Boosting are good models, however, Logistic & LDA Model gives better results. Best model is logistic regression as the difference of error is very less in training and test data has a best fit model.

We observe Labour has higher possibility of winning

Labour has higher voting possibility among all age groups except for very old people

Irrespective of the political knowledge levels or gender, Labour has an edge on higher votes

Where the Eurosceptic sentiment is more, Conservative has scope for winning

Problem 2

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**

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

Importing the necessary libraries along with the standard import, Code Snippet to extract the three speeches:

```
[nltk_data] Downloading package inaugural to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\inaugural.zip.
```

Let us check the version of the various libraries

```
Numpy version: 1.21.5
Pandas version: 1.4.2
Regular Expression version: 2.2.1
Natural Language Tool Kit version: 3.7
Matplotlib version: 3.5.1
```

President Franklin D. Roosevelt in 1941

```
'On each national day of inauguration since 1789, the people have renewed their sense of dedication to the United States.\n\nIn
Washington\'s day the task of the people was to create and weld together a nation.\n\nIn Lincoln\'s day the task of the people
was to preserve that Nation from disruption from within.\n\nIn this day the task of the people is to save that Nation and its i
nstitutions from disruption from without.\n\nTo us there has come a time, in the midst of swift happenings, to pause for a mome
nt and take stock -- to recall what our place in history has been, and to rediscover what we are and what we may be. If we do n
ot, we risk the real peril of inaction.\n\nLives of nations are determined not by the count of years, but by the lifetime of th
e human spirit. The life of a man is three-score years and ten: a little more, a little less. The life of a nation is the fulln
ess of the measure of its will to live.\n\nThere are men who doubt this. There are men who believe that democracy, as a form of
Government and a frame of life, is limited or measured by a kind of mystical and artificial fate that, for some unexplained rea
son, tyranny and slavery have become the surging wave of the future -- and that freedom is an ebbing tide.\n\nBut we Americans
know that this is not true.\n\nEight years ago, when the life of this Republic seemed frozen by a fatalistic terror, we proved
that this is not true. We were in the midst of shock -- but we acted. We acted quickly, boldly, decisively.\n\nThese later year
s have been living years -- fruitful years for the people of this democracy. For they have brought to us greater security and,
I hope, a better understanding that life\'s ideals are to be measured in other than material things.\n\nMost vital to our prese
nt and our future is this experience of a democracy which successfully survived crisis at home; put away many evil things; buil
t new structures on enduring lines; and, through it all, maintained the fact of its democracy.\n\nFor action has been taken wit
hin the three-way framework of the Constitution of the United States. The coordinate branches of the Government continue freely
to function. The Bill of Rights remains inviolate. The freedom of elections is wholly maintained. Prophets of the downfall of A
merican democracy have seen their dire predictions come to naught.\n\nDemocracy is not dying.\n\nWe know it because we have see
n it revive--and grow.\n\nWe know it cannot die -- because it is built on the unhampered initiative of individual men and women
```

President John F. Kennedy in 1961

'Vice President Johnson, Mr. Speaker, Mr. Chief Justice, President Eisenhower, Vice President Nixon, President Truman, reverend clergy, fellow citizens, we observe today not a victory of party, but a celebration of freedom -- symbolizing an end, as well a s a beginning -- signifying renewal, as well as change. For I have sworn I before you and Almighty God the same solemn oath our forebears l prescribed nearly a century and three quarters ago.\n\nThe world is very different now. For man holds in his mortal hands the power to abolish all forms of human poverty and all forms of human life. And yet the same revolutionary beliefs for w hich our forebears fought are still at issue around the globe -- the belief that the rights of man come not from the generosity of the state, but from the hand of God.\n\nWe dare not forget today that we are the heirs of that first revolution. Let the wor d go forth from this time and place, to friend and foe alike, that the torch has been passed to a new generation of Americans - - born in this century, tempered by war, disciplined by a hard and bitter peace, proud of our ancient heritage -- and unwilling to witness or permit the slow undoing of those human rights to which this Nation has always been committed, and to which we are committed today at home and around the world.\n\nLet every nation know, whether it wishes us well or ill, that we shall pay any price, bear any burden, meet any hardship, support any friend, oppose any foe, in order to assure the survival and the success of liberty.\n\nThis much we pledge -- and more.\n\nTo those old allies whose cultural and spiritual origins we share, we pledge the loyalty of faithful friends. United, there is little we cannot do in a host of cooperative ventures. Divided, there is litt le we can do -- for we dare not meet a powerful challenge at odds and split asunder.\n\nTo those new States whom we welcome to

President Richard Nixon in 1973

'Mr. Vice President, Mr. Speaker, Mr. Chief Justice, Senator Cook, Mrs. Eisenhower, and my fellow citizens of this great and go od country we share together:\n\nWhen we met here four years ago, America was bleak in spirit, depressed by the prospect of see mingly endless war abroad and of destructive conflict at home.\n\nAs we meet here today, we stand on the threshold of a new era of peace in the world.\n\nThe central question before us is: How shall we use that peace? Let us resolve that this era we are a bout to enter will not be what other postwar periods have so often been: a time of retreat and isolation that leads to stagnati on at home and invites new danger abroad.\n\nLet us resolve that this will be what it can become: a time of great responsibilit ies greatly borne, in which we renew the spirit and the promise of America as we enter our third century as a nation.\n\nThis p ast year saw far-reaching results from our new policies for peace. By continuing to revitalize our traditional friendships, and by our missions to Peking and to Moscow, we were able to establish the base for a new and more durable pattern of relationships among the nations of the world. Because of America\'s bold initiatives, 1972 will be long remembered as the year of the greates t progress since the end of World War II toward a lasting peace in the world.\n\nThe peace we seek in the world is not the flim sy peace which is merely an interlude between wars, but a peace which can endure for generations to come.\n\nIt is important th at we understand both the necessity and the limitations of America\'s role in maintaining that peace.\n\nUnless we in America w ork to preserve the peace, there will be no peace.\n\nUnless we in America work to preserve freedom, there will be no freedo m.\n\nBut let us clearly understand the new nature of America\'s role, as a result of the new policies we have adopted over the

Number of Characters

Number of Characters in Speech of President Franklin D. Roosevelt in 1941 are 7571

Number of Characters in Speech of President John F. Kennedy in 1961 are 7618

Number of Characters in Speech of President Richard Nixon in 1973 are 9991

Number of Words

Number of words in Speech of President Franklin D. Roosevelt in 1941 are 1536

Number of words in Speech of President John F. Kennedy in 1961 are 1546

Number of words in Speech of President Richard Nixon in 1973 are 2028

Number of Sentences

Number of Sentences in Speech of President Franklin D. Roosevelt in 1941 are 68

Number of Sentences in Speech of President John F. Kennedy in 1961 are 52

Number of Sentences in Speech of President Richard Nixon in 1973 are 69

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

Word count before and after removal of stopwords

```
Number of words in Speech of President Franklin D. Roosevelt in 1941-before removal of stopwords are 1536
Number of words in Speech of President Franklin D. Roosevelt in 1941-after removal of stopwords are 632
Number of words in Speech of President John F. Kennedy in 1961-before removal of stopwords are 1546
Number of words in Speech of President John F. Kennedy in 1961-after removal of stopwords are 697
Number of words in Speech of President Richard Nixon in 1973-before removal of stopwords are 2028
Number of words in Speech of President Richard Nixon in 1973-after removal of stopwords are 836
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Sample sentence

```
Original example sentence ['On', 'each', 'national', 'day', 'of', 'inauguration', 'since', '1789', ',', 'the', 'people', 'hav
e', 'renewed', 'their', 'sense', 'of', 'dedication', 'to', 'the', 'United', 'States', '.']
Filtered sentence ['On', 'national', 'day', 'inauguration', 'since', '1789', 'people', 'renewed', 'sense', 'dedication', 'Unite
d', 'States']
```

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

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
Top word which occured the most number of times in inaugural address of President Franklin in 1941 is [('nation', 12)]
Top word which occured the most number of times in inaugural address of President John F. Kennedy is [('let', 16)]
Top word which occured the most number of times in inaugural address of President Richard Nixon is [('us', 26)]
```

```
Top 3 words which occured the most number of times in inaugural address of President Franklin in 1941 is [('nation', 12), ('kno
w', 10), ('spirit', 9)]
Top 3 words which occured the most number of times in inaugural address of President John F. Kennedy is [('let', 16), ('us', 1
2), ('world', 8)]
Top 3 words which occured the most number of times in inaugural address of President Richard Nixon is [('us', 26), ('let', 22),
('america', 21)]
```

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

Word Cloud for President Franklin D. Roosevelt speech in 1941

Word Cloud for President John F. Kennedy speech in 1961

Word Cloud for President Richard Nixon speech in 1973

THANK YOU