PROJECT REPORT ON MACHINE LEARNING

Akshaya Parthasarathy

Batch: PGPDSBA_online_July E 2020

Problem 1:

Election data with various models

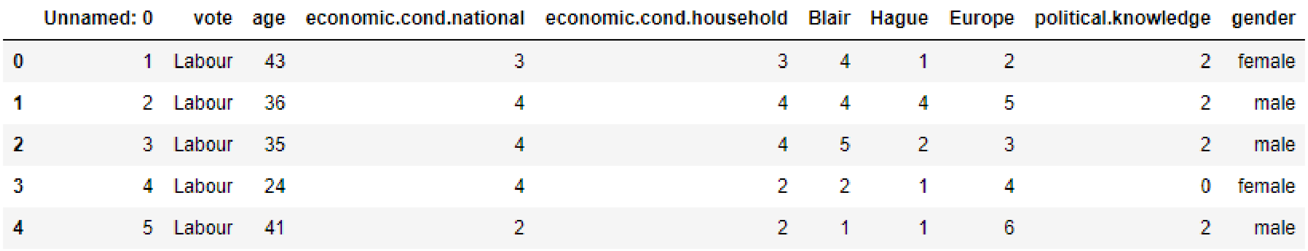
Problem statement:

You are hired by one of the leading news channel 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.

1.1 Read the dataset. Do the descriptive statistics and do null value condition check.

Exploratory Data Analysis:

Head of the dataset: Verify whether the dataset is loaded correctly



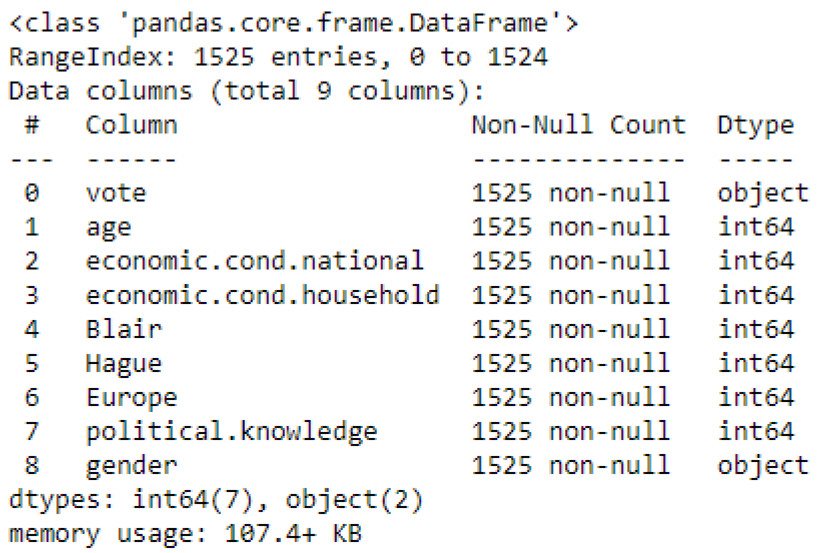
	Unnamed: 0	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	1	Labour	43	3	3	4	1	2	2	female
1	2	Labour	36	4	4	4	4	5	2	male
2	3	Labour	35	4	4	5	2	3	2	male
3	4	Labour	24	4	2	2	1	4	0	female
4	5	Labour	41	2	2	1	1	6	2	male

Dropping the column ‘Unnamed: 0’ since it’s an index column.

Shape of the dataset:

```
There are  1525  rows and  9  columns in the dataset.
```

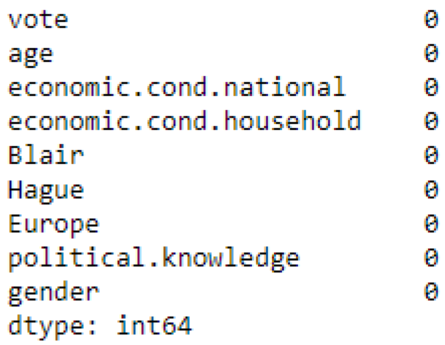
Information of the dataset: There are nine variables in the dataset of which ‘vote and gender’ are of object type and rest are of int type.



```
<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
```

There are no null values in entire dataset.

Null values check in the dataset:

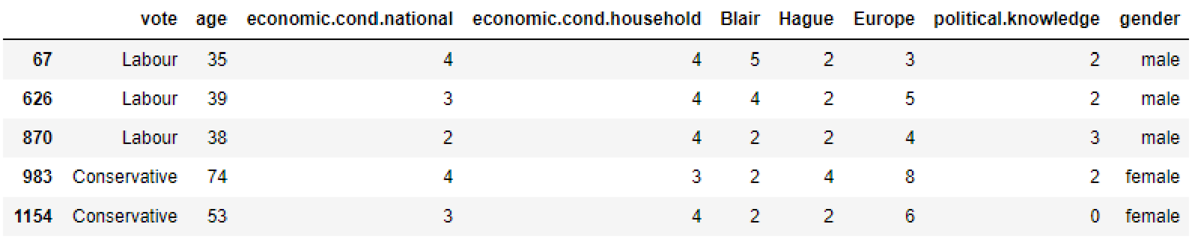


```
vote                       0
age                        0
economic.cond.national     0
economic.cond.household    0
Blair                      0
Hague                      0
Europe                     0
political.knowledge        0
gender                     0
dtype: int64
```

Duplicate records check in the dataset:

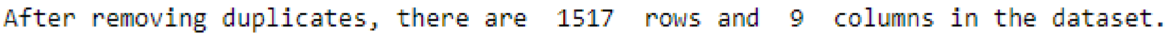
Total number of duplicated records: 8

Since there is no unique identifier in the given dataset, we can consider these 8 records to be purely duplicates and remove from the dataset.



	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

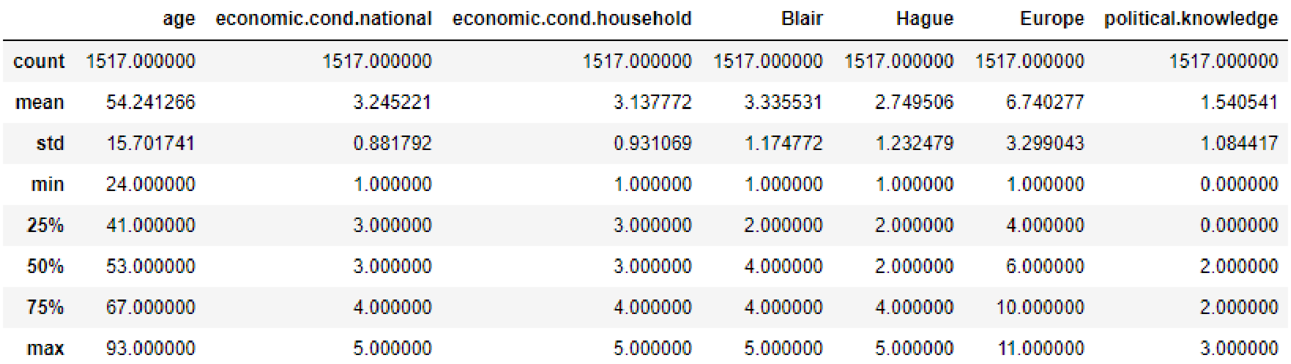
Shape of the dataset after removal of duplicates:



```
After removing duplicates, there are  1517  rows and  9  columns in the dataset.
```

Summary statistics of the dataset:

Numerical columns:

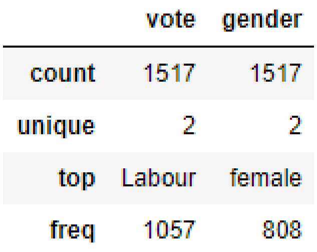


	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
count	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000
mean	54.241266	3.245221	3.137772	3.335531	2.749506	6.740277	1.540541
std	15.701741	0.881792	0.931069	1.174772	1.232479	3.299043	1.084417
min	24.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
25%	41.000000	3.000000	3.000000	2.000000	2.000000	4.000000	0.000000
50%	53.000000	3.000000	3.000000	4.000000	2.000000	6.000000	2.000000
75%	67.000000	4.000000	4.000000	4.000000	4.000000	10.000000	2.000000
max	93.000000	5.000000	5.000000	5.000000	5.000000	11.000000	3.000000

Inference:

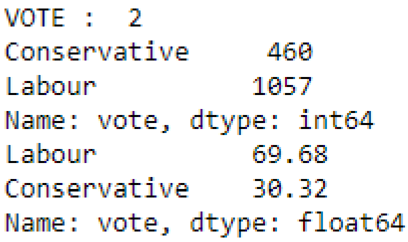
- Looking at the mean and median of the columns, they are almost similar which indicates less skewness in the dataset.
- Except Age variable, all others are within a range of 0-10.
- Outliers are not present in the dataset and there are no values which stands out as abnormal for further analysis.

Categorical columns:

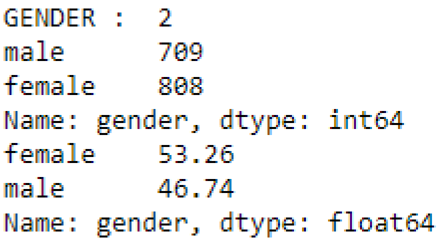


	vote	gender
count	1517	1517
unique	2	2
top	Labour	female
freq	1057	808

Looking at the unique values and the value counts for each:



```
VOTE :  2
Conservative     460
Labour          1057
Name: vote, dtype: int64
Labour          69.68
Conservative    30.32
Name: vote, dtype: float64
```



```
GENDER :  2
male      709
female    808
Name: gender, dtype: int64
female    53.26
male      46.74
Name: gender, dtype: float64
```

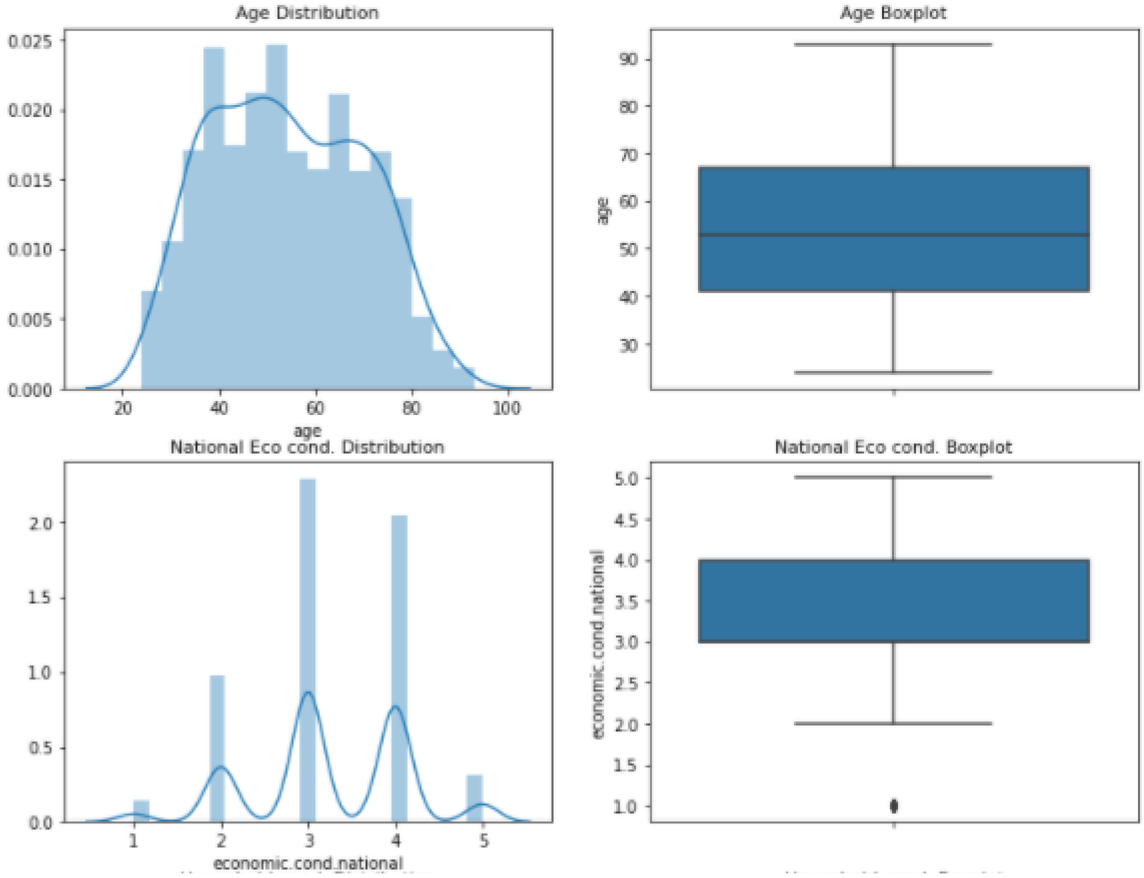
Inference:

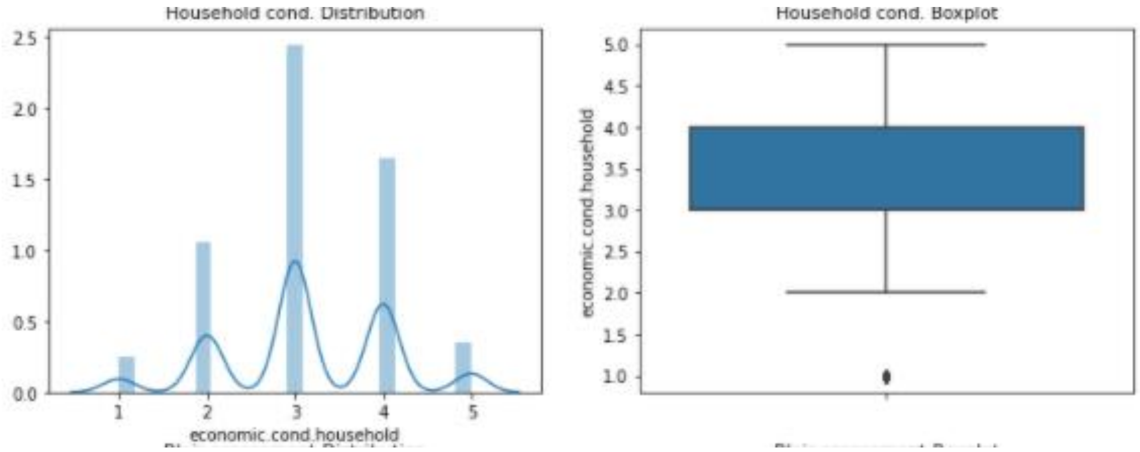
- Gender column seems balanced. In the given dataset, Female voters are on the higher side.

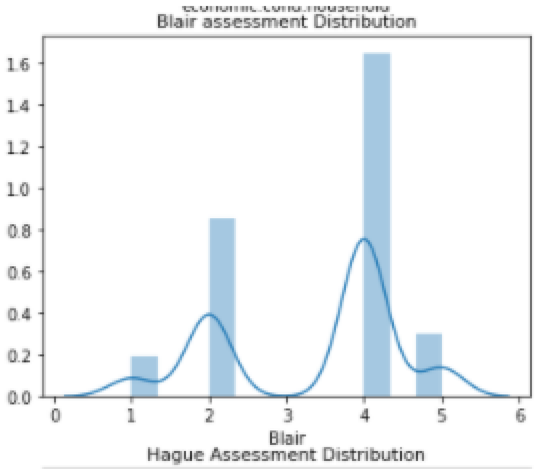
- Out of all the voters, 70% casted their vote to Labour Party and 30% to Conservative party. Still this scenario does not come under unbalanced dataset since we have enough data for further prediction.

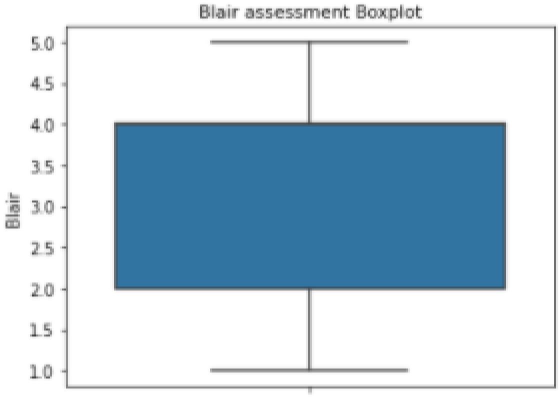
1.2 Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also check for outliers. Interpret the inferences for each

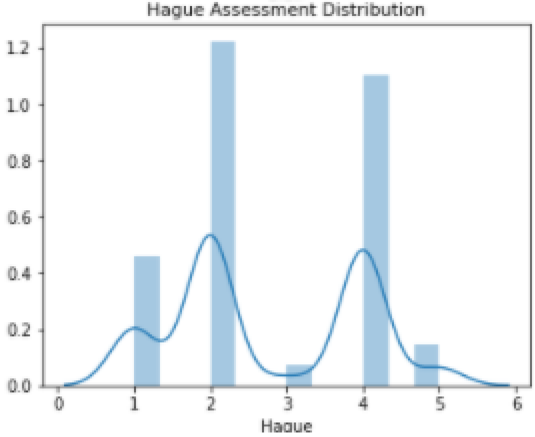
Univariate Analysis:

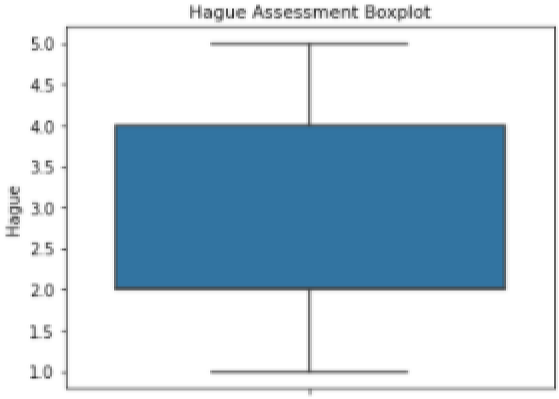


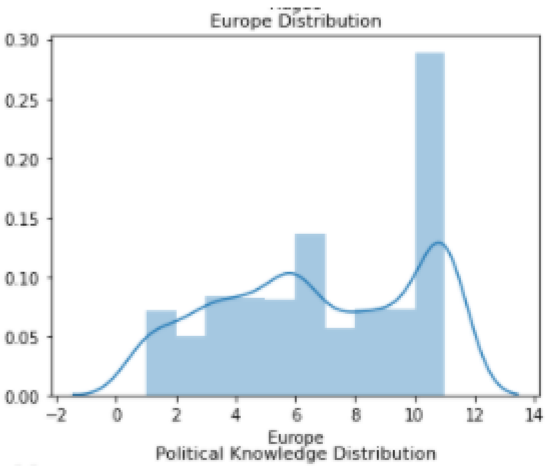


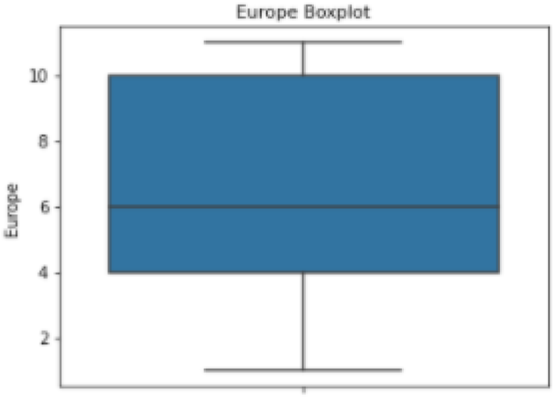

economic.cond.household
Blair assessment Distribution
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0
1
2
3
4
5
6
Blair

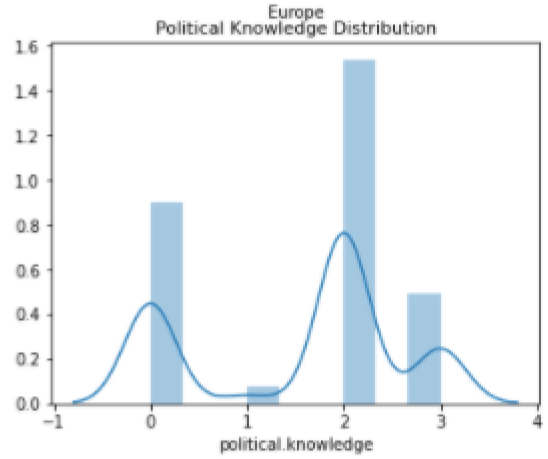

Blair assessment Boxplot
Blair
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

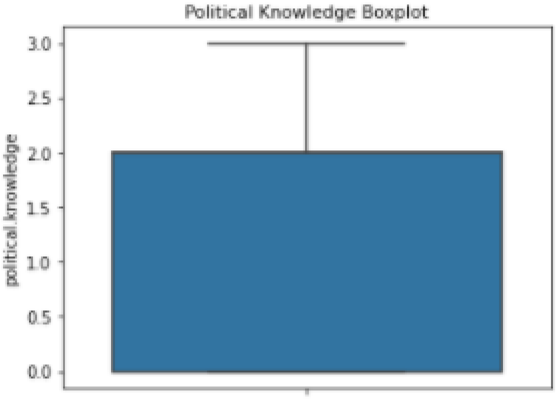

Hague Assessment Distribution
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
1
2
3
4
5
6
Hague


Hague Assessment Boxplot
Hague
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


Hague
Europe Distribution
0.00
0.05
0.10
0.15
0.20
0.25
0.30
−2
0
2
4
6
8
10
12
14
Europe


Europe Boxplot
Europe
2
4
6
8
10


Political Knowledge Distribution
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
−1
0
1
2
3
4
political.knowledge

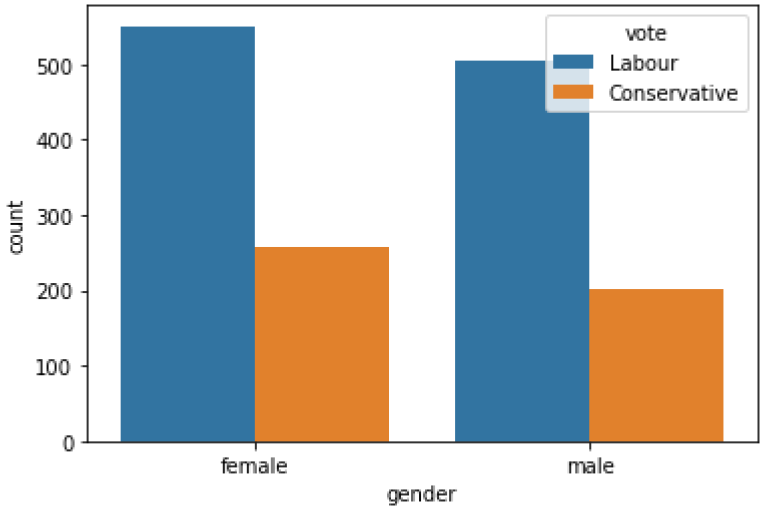

Political Knowledge Boxplot
political.knowledge
0.0
0.5
1.0
1.5
2.0
2.5
3.0

Inference:

- Only National economic condition and Household condition has outlier present in the lower range since the 25% quantile and median are both same. The values less than that are shown as outliers.
- Almost all the columns have clusters of datapoints since the columns have values which are scale based.
- Age is a continuous variable which has no skewness.

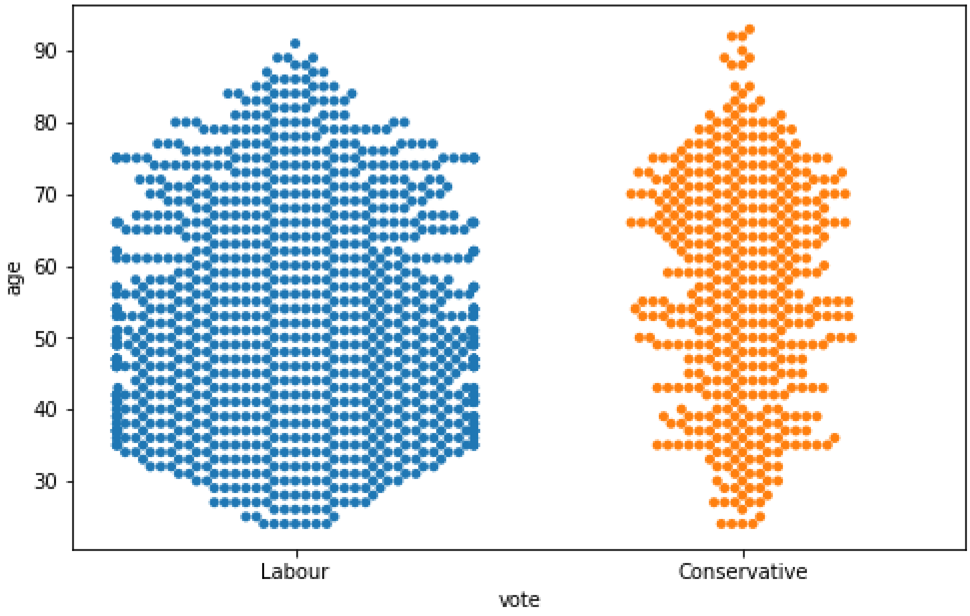
Bivariate analysis:

1. **Count plot of Gender with hue vote:**



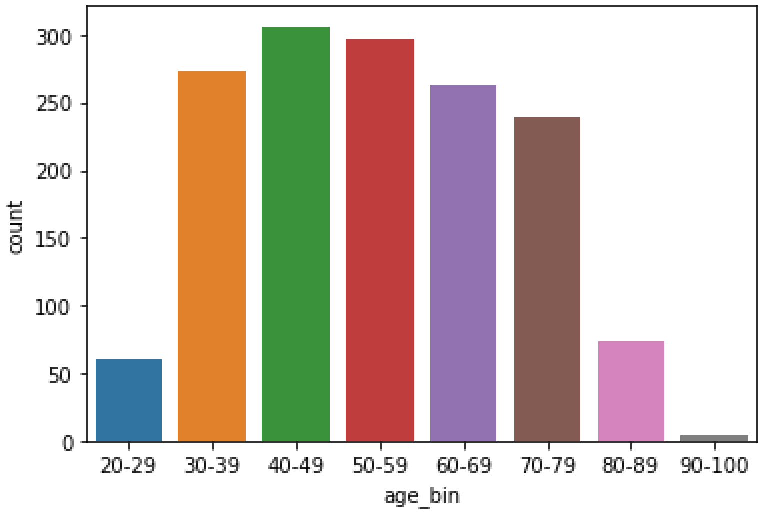
Irrespective of the gender, voters are inclined towards Labour Party than conservative.

2. **Swarm plot of Age and vote:**



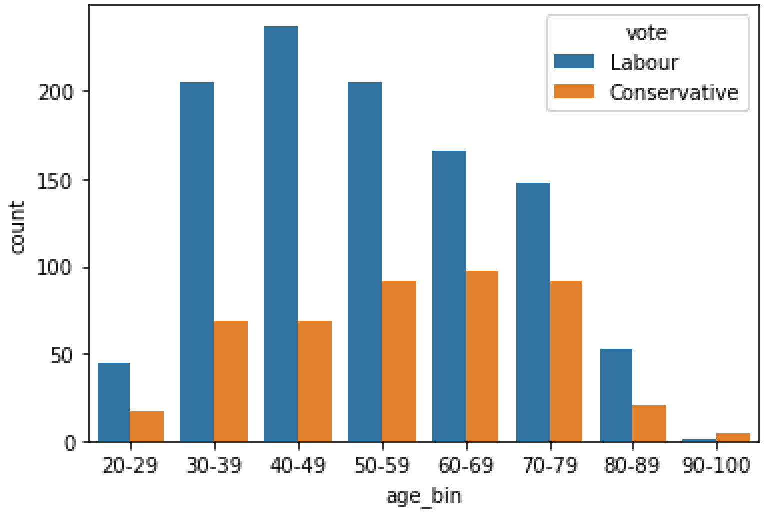
As the age increases, we could see that voters tend to vote for Conservative party. Young age people tend to vote for Labour Party.

3. **Creating 8 bins for age with an interval of 10 and count plot for the same:**



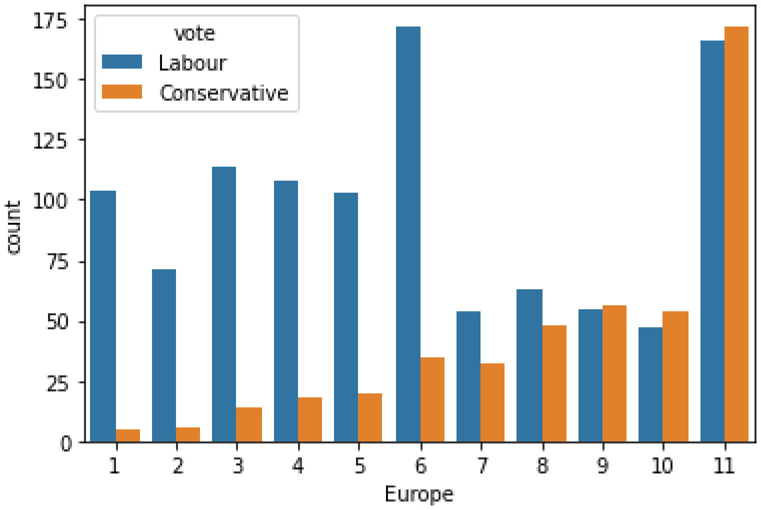
In the given dataset, maximum number of voters are between the age group 30-80.

4. **Count plot for age bins and vote:**



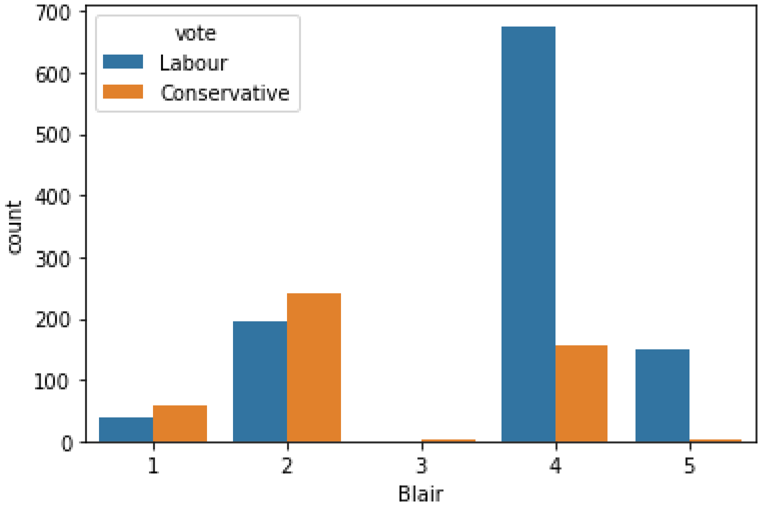
Trend of the vote with respect to the age is clearly visible in this plot. As the age increases voters go for Conservative party.

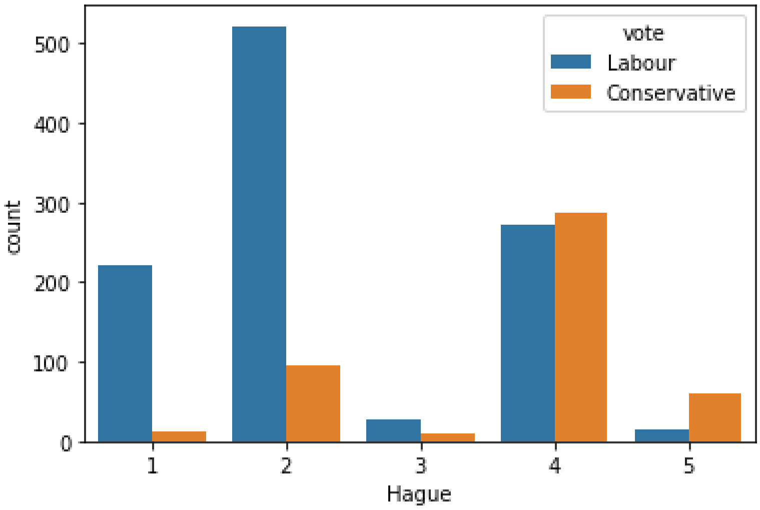
5. Count plot for Europe and vote:



Europe: an 11-point scale that measures respondents' attitudes toward European integration. High scores represent ‘Eurosceptic’ sentiment which is opposing closer connections with Britain and EU. Higher the Eurosceptic sentiment among voters, conservative party gets the vote. Lower the Eurosceptic sentiment, labour party gets the vote. Voters for labour party are in favour of the integration of Europe and Britain.

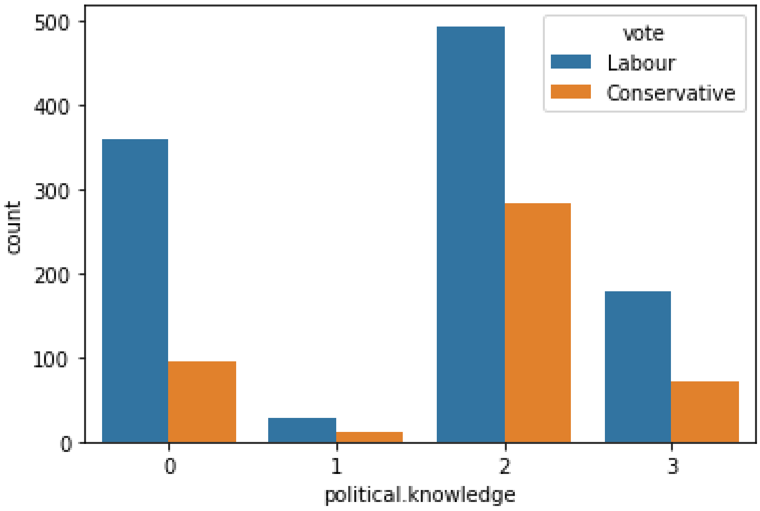
6. Count plot for Blair/Hague with vote:





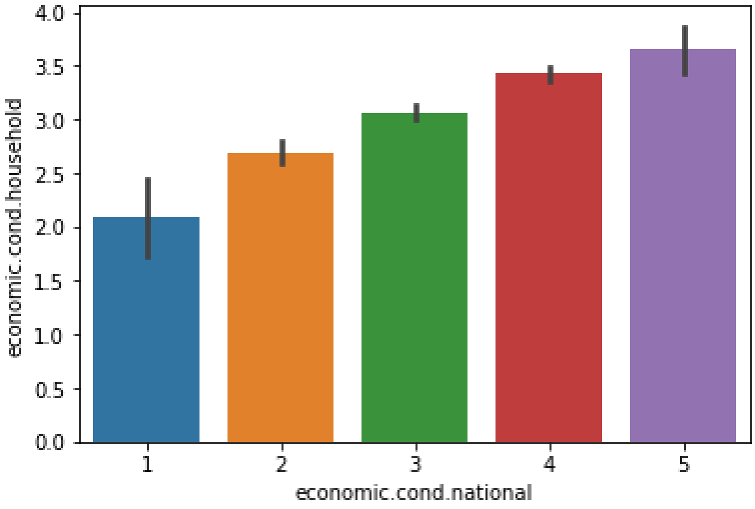
As the assessment of each leader increases, respective voting to that party increases. Looks like the voters tend to have strong opinions to whom their support is, hence we are able to see distinct difference in the assessment given by the voters. Proportion of people who have given 3 is low for both Hague and Blair.

7. Count plot for political knowledge and vote:



Irrespective of the political knowledge, Labour party voters are more in each level. We can also infer that voters with zero knowledge of party’s attitude towards European integration is on the higher side.

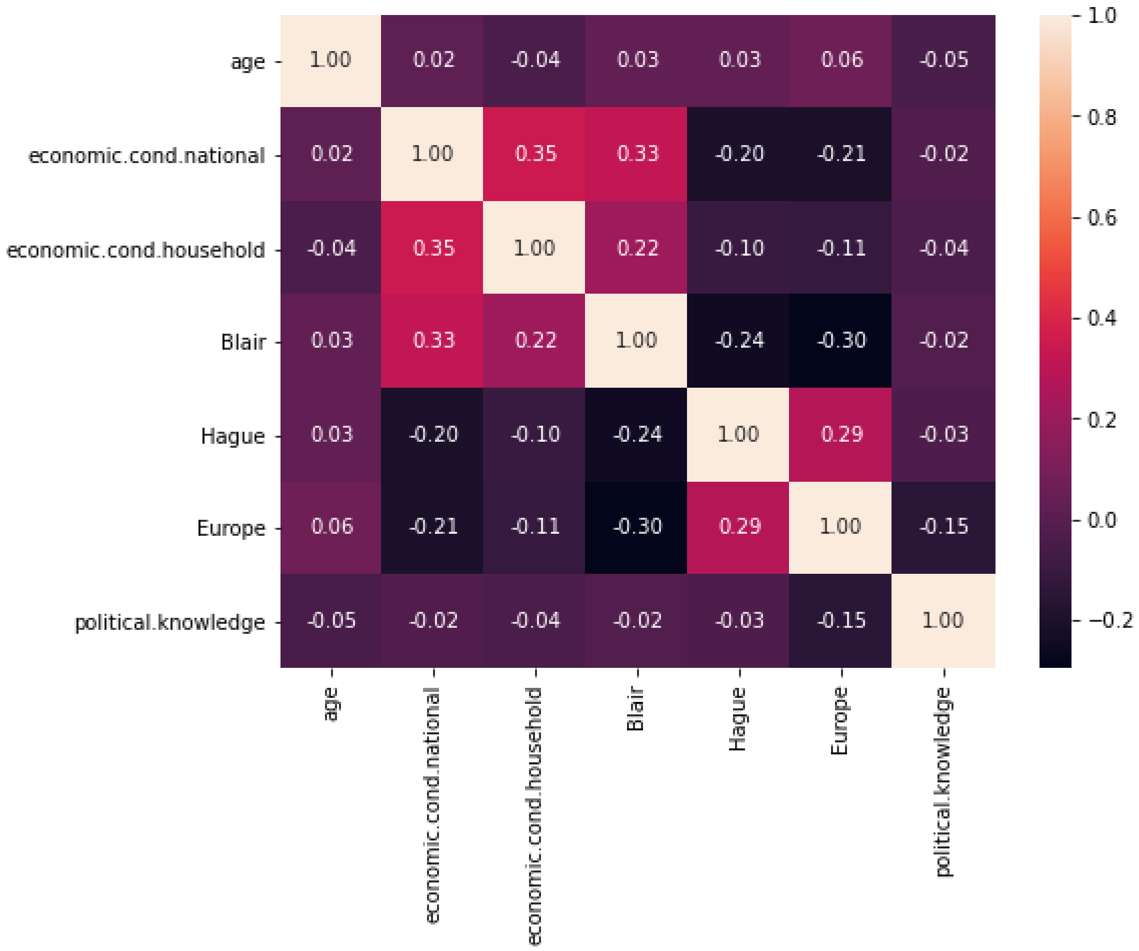
8. **Bar plot of Economic and household condition:**



As the national economic condition increases, household economic conditions also increase.

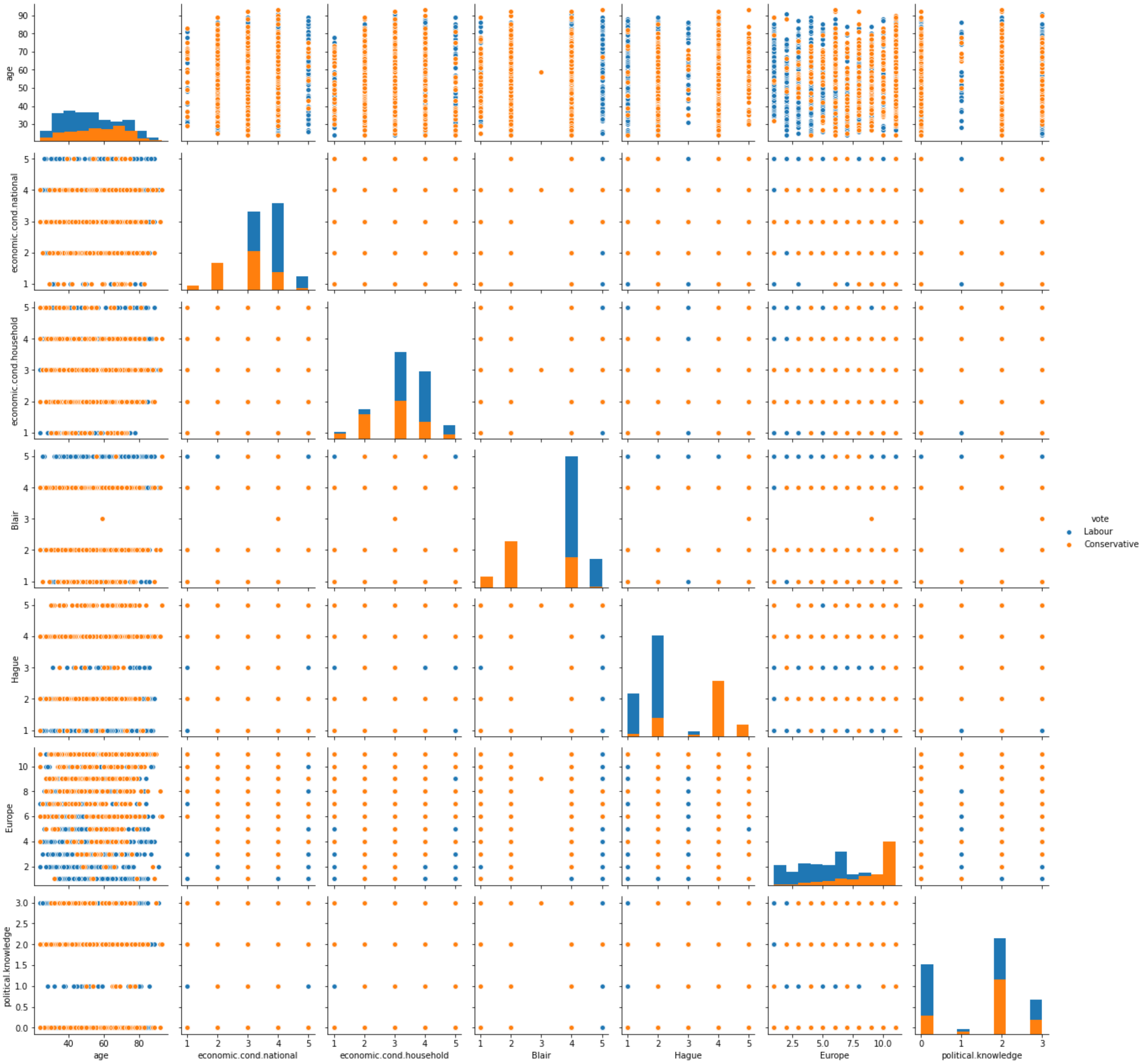
Multivariate analysis:

1. **Correlation plot using heatmap:**



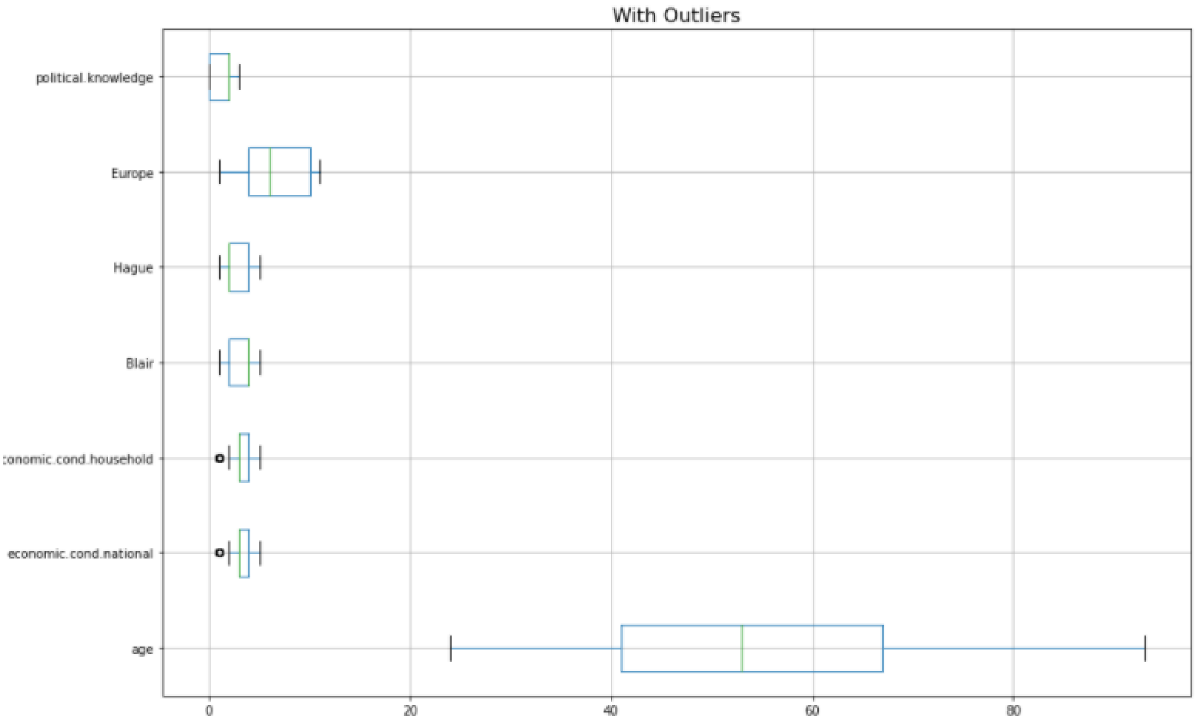
Mostly we can see negative correlations between the variables. But all the correlations are on the weaker side irrespective of the direction. No evidence for multicollinearity between the variables.

2. Pair plot with hue as vote:



Best predictors of target variable vote can be Hague, Blair, Europe. Since these variables show a slight distinction between the vote (Labour/Conservative).

Outliers check:



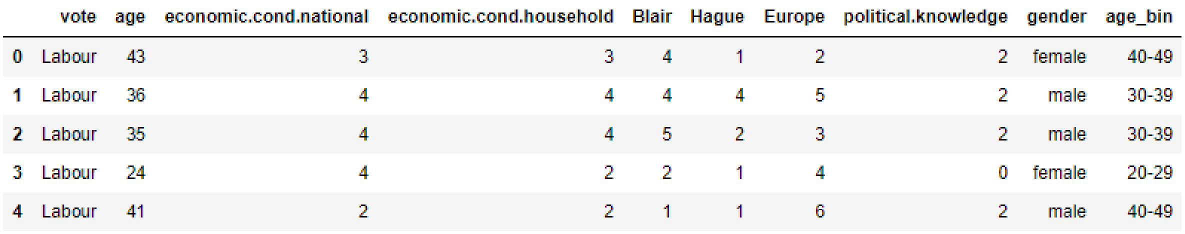
Only ‘economic condition household’ and ‘economic condition national’ have outliers in the lower range. Since it can be a valid value, outliers are not treated for this case study.

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).

Scaling:

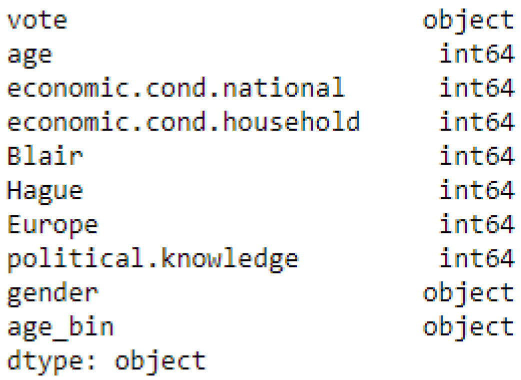
Since the dataset does not contain variables which are of different scale. Scaling will not be necessary for this particular case.

However, age column has values ranging from 25-90. This range is large compared to other variables. Hence creating a new column for age by separating it into 8 bins each of interval size 10, starting from 20 till 100 and using this column for further modelling and analysis.



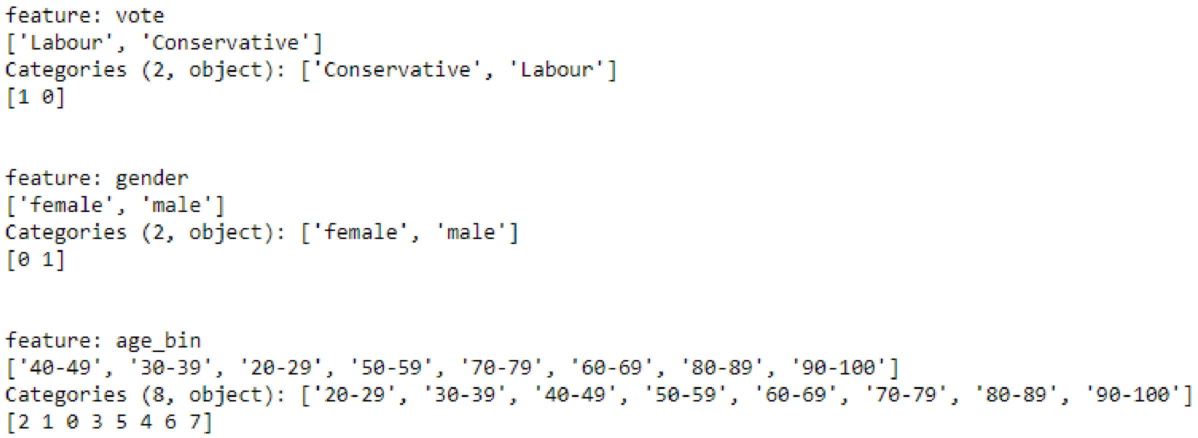
	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender	age_bin
0	Labour	43	3	3	4	1	2	2	female	40-49
1	Labour	36	4	4	4	4	5	2	male	30-39
2	Labour	35	4	4	5	2	3	2	male	30-39
3	Labour	24	4	2	2	1	4	0	female	20-29
4	Labour	41	2	2	1	1	6	2	male	40-49

Encoding the data with ‘object’ type:



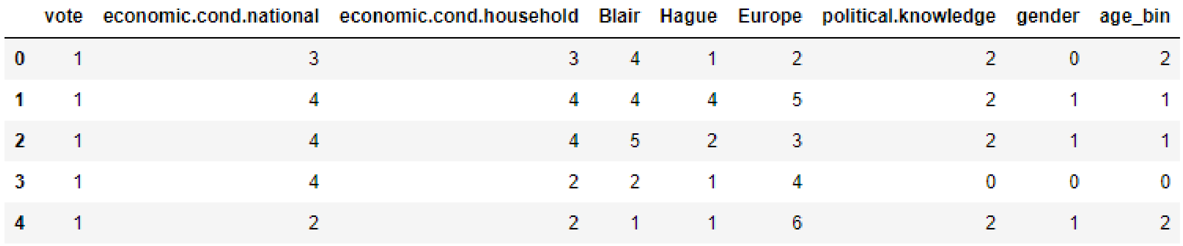
```
vote                         object
age                           int64
economic.cond.national        int64
economic.cond.household       int64
Blair                         int64
Hague                         int64
Europe                        int64
political.knowledge           int64
gender                       object
age_bin                      object
dtype: object
```

Using pd.Categorical codes, getting the codes for the object type variables:



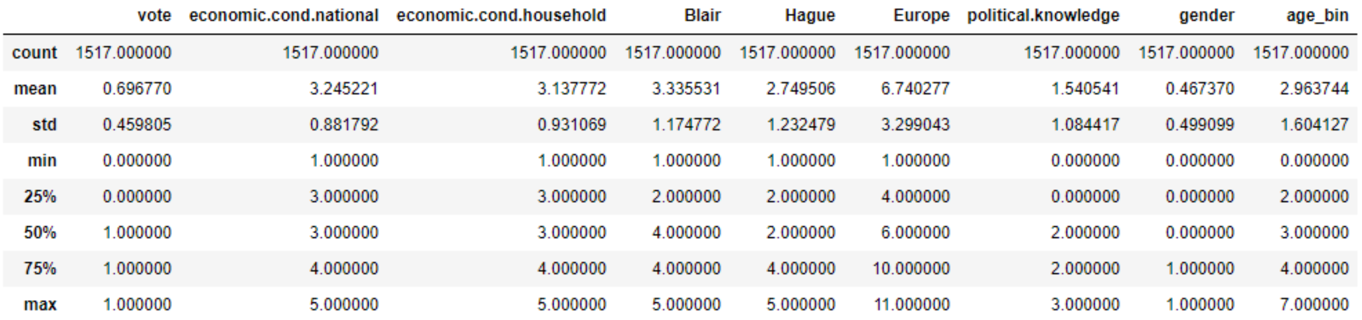
```
feature: vote
['Labour', 'Conservative']
Categories (2, object): ['Conservative', 'Labour']
[1 0]


feature: gender
['female', 'male']
Categories (2, object): ['female', 'male']
[0 1]


feature: age_bin
['40-49', '30-39', '20-29', '50-59', '70-79', '60-69', '80-89', '90-100']
Categories (8, object): ['20-29', '30-39', '40-49', '50-59', '60-69', '70-79', '80-89', '90-100']
[2 1 0 3 5 4 6 7]
```

Dataset after encoding:



	vote	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender	age_bin
0	1	3	3	4	1	2	2	0	2
1	1	4	4	4	4	5	2	1	1
2	1	4	4	5	2	3	2	1	1
3	1	4	2	2	1	4	0	0	0
4	1	2	2	1	1	6	2	1	2

Summary statistics of the above dataset:



	vote	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender	age_bin
count	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000
mean	0.696770	3.245221	3.137772	3.335531	2.749506	6.740277	1.540541	0.467370	2.963744
std	0.459805	0.881792	0.931069	1.174772	1.232479	3.299043	1.084417	0.499099	1.604127
min	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000
25%	0.000000	3.000000	3.000000	2.000000	2.000000	4.000000	0.000000	0.000000	2.000000
50%	1.000000	3.000000	3.000000	4.000000	2.000000	6.000000	2.000000	0.000000	3.000000
75%	1.000000	4.000000	4.000000	4.000000	4.000000	10.000000	2.000000	1.000000	4.000000
max	1.000000	5.000000	5.000000	5.000000	5.000000	11.000000	3.000000	1.000000	7.000000

- **First step** of building a model is to **Separate the dataset into X and y variable.**

For the given business problem of election data, ‘vote’ is the target variable since the problem is to come up with a model to create exit poll for predicting overall win and seats covered by a particular party.

X – Independent variable (Removing ‘vote’ variable)

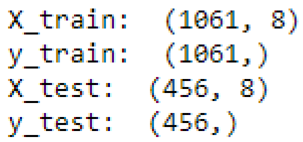
Y – Dependent/ Target variable (Having only ‘vote’ variable)

- **Second step** is to **Split the data into training and testing test.**

Splitting the data as 70% training and 30% testing.

Output of this step will be: Training independent variable (X_train), Testing independent variable (X-test), Training dependent variable (y_train) and testing dependent variable (y_test).

Shape of the dataset:



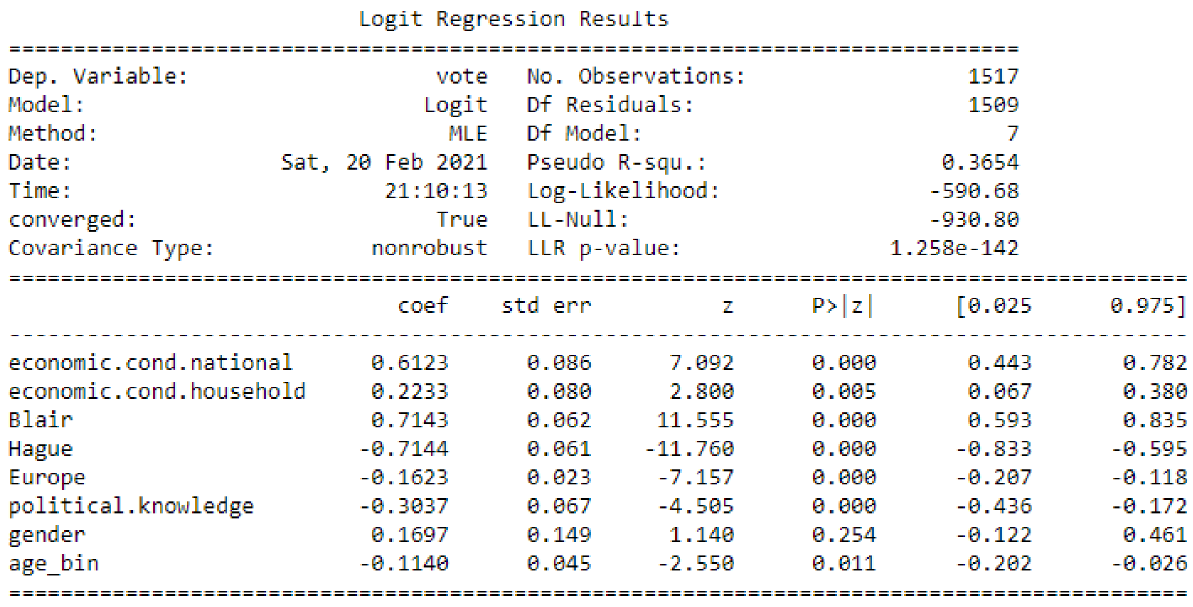
```
X_train:  (1061, 8)
y_train:  (1061,)
X_test:  (456, 8)
y_test:  (456,)
```

1.4 Apply Logistic Regression and LDA (Linear Discriminant Analysis). Interpret the inferences of both models

- **Third step is to build model for each LDA and Logistic Regression and fourth step is to predict on training and testing set**

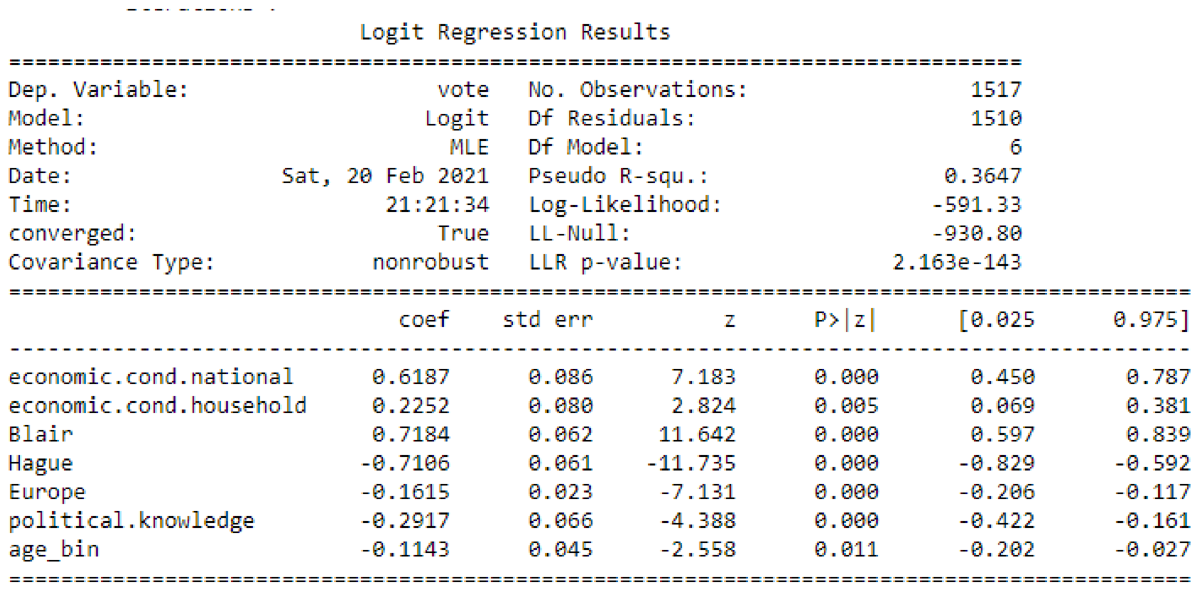
Logistic Regression is a supervised learning method for classification. It establishes relationship between dependent class variables and independent class variables using regression. Logistic regression assign probabilities to different classes to which a data point is likely to belong. In order to do this, the classifier takes the weighted sum of the features and bias to represent the class of interest of a particular data point, this linear output is passed through a sigmoid function in order to get the values between the range (0,1).

Using **Logit function from statsmodels** in order to determine the p-value of variables and to determine if it’s a good predictor.



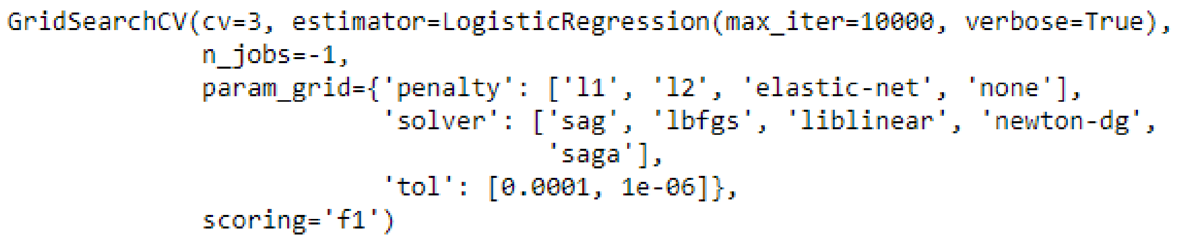
```
                           Logit Regression Results
==============================================================================
Dep. Variable:                   vote   No. Observations:                 1517
Model:                          Logit   Df Residuals:                     1509
Method:                           MLE   Df Model:                            7
Date:                Sat, 20 Feb 2021   Pseudo R-squ.:                  0.3654
Time:                        21:10:13   Log-Likelihood:                -590.68
converged:                       True   LL-Null:                       -930.80
Covariance Type:            nonrobust   LLR p-value:                1.258e-142
=============================================================================================
                              coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------
economic.cond.national      0.6123      0.086      7.092      0.000       0.443       0.782
economic.cond.household     0.2233      0.080      2.800      0.005       0.067       0.380
Blair                       0.7143      0.062     11.555      0.000       0.593       0.835
Hague                      -0.7144      0.061    -11.760      0.000      -0.833      -0.595
Europe                     -0.1623      0.023     -7.157      0.000      -0.207      -0.118
political.knowledge        -0.3037      0.067     -4.505      0.000      -0.436      -0.172
gender                      0.1697      0.149      1.140      0.254      -0.122       0.461
age_bin                    -0.1140      0.045     -2.550      0.011      -0.202      -0.026
=============================================================================================
```

p-values for all variables are less than 0.05 except ‘gender’. Hence removing it for further model building.



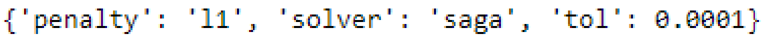
```
                           Logit Regression Results
==============================================================================
Dep. Variable:                   vote   No. Observations:                 1517
Model:                          Logit   Df Residuals:                     1510
Method:                           MLE   Df Model:                            6
Date:                Sat, 20 Feb 2021   Pseudo R-squ.:                  0.3647
Time:                        21:21:34   Log-Likelihood:                -591.33
converged:                       True   LL-Null:                       -930.80
Covariance Type:            nonrobust   LLR p-value:                2.163e-143
=============================================================================================
                              coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------
economic.cond.national      0.6187      0.086      7.183      0.000       0.450       0.787
economic.cond.household     0.2252      0.080      2.824      0.005       0.069       0.381
Blair                       0.7184      0.062     11.642      0.000       0.597       0.839
Hague                      -0.7106      0.061    -11.735      0.000      -0.829      -0.592
Europe                     -0.1615      0.023     -7.131      0.000      -0.206      -0.117
political.knowledge        -0.2917      0.066     -4.388      0.000      -0.422      -0.161
age_bin                    -0.1143      0.045     -2.558      0.011      -0.202      -0.027
=============================================================================================
```

Grid search CV parameters used:

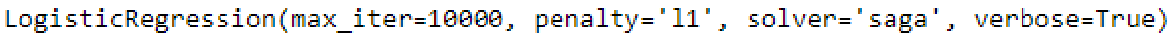


```
GridSearchCV(cv=3, estimator=LogisticRegression(max_iter=10000, verbose=True),
             n_jobs=-1,
             param_grid={'penalty': ['l1', 'l2', 'elastic-net', 'none'],
                         'solver': ['sag', 'lbfgs', 'liblinear', 'newton-dg',
                                    'saga'],
                         'tol': [0.0001, 1e-06]},
             scoring='f1')
```

Best parameters obtained:



```
{'penalty': 'l1', 'solver': 'saga', 'tol': 0.0001}
```



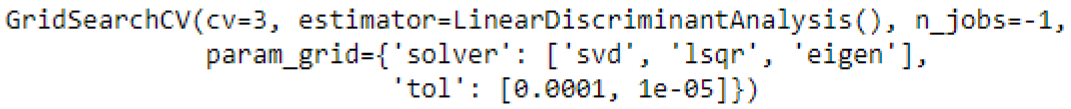
```
LogisticRegression(max_iter=10000, penalty='l1', solver='saga', verbose=True)
```

Linear Discriminant Analysis is a linear classification machine learning algorithm.

The algorithm involves developing a probabilistic model per class based on the specific distribution of observations for each input variable. A new data point is then classified by calculating the conditional probability of it belonging to each class and selecting the class with the highest probability.

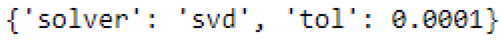
This model is useful when we have independent variables are a clear distinguishers of target variable.

Grid search CV parameters used:

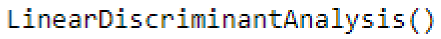


```
GridSearchCV(cv=3, estimator=LinearDiscriminantAnalysis(), n_jobs=-1,
             param_grid={'solver': ['svd', 'lsqr', 'eigen'],
                         'tol': [0.0001, 1e-05]})
```

Best parameters obtained:



```
{'solver': 'svd', 'tol': 0.0001}
```



```
LinearDiscriminantAnalysis()
```

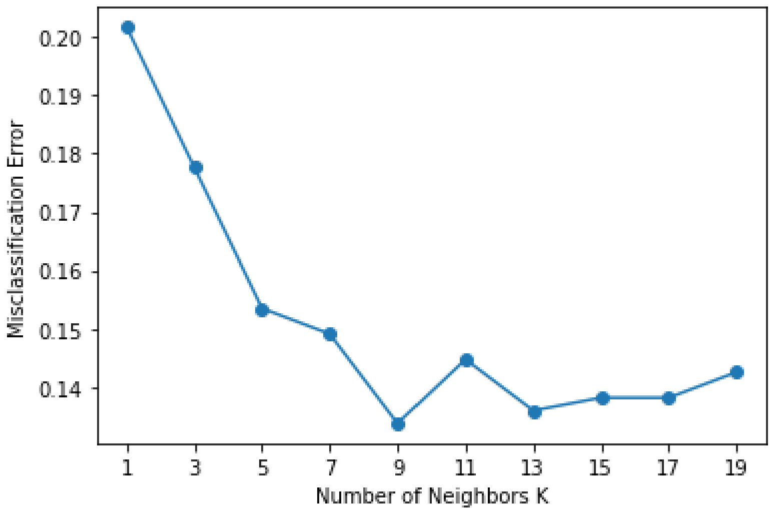
1.5 Apply KNN Model and Naïve Bayes Model. Interpret the inferences of each model

KNN Model is a non-parametric method which decides from a neighbouring data point what a new data point should be classified as. It makes use of distance measure to find the nearest neighbours.

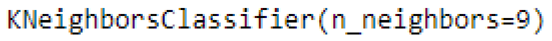
Choosing K-value:

Using the misclassification error vs K-neighbours plot, we can determine the optimal k-value. The value for which misclassification error is the least is chosen.

For a k-value of 9, we get least MCE. Hence building the KNN model with this value.



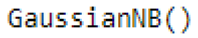
Model obtained:



```
KNeighborsClassifier(n_neighbors=9)
```

Naïve Bayes model is a probabilistic model based on bayes theorem. 'Naïve' is due to the assumption that the features in the dataset are mutually independent. The idea is to factor all available evidence in form of predictors to obtain more accurate probability for class prediction. It estimates the conditional probability that an event will happen given that another event has already occurred.

Model built:



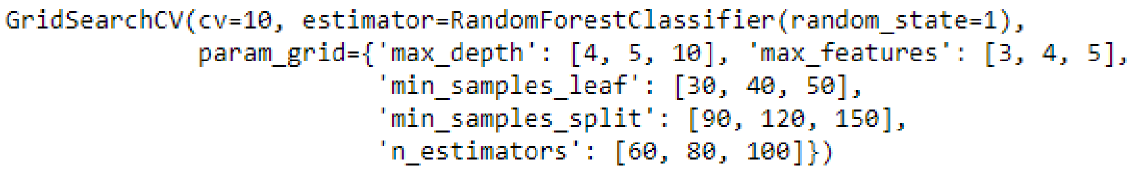
```
GaussianNB()
```

1.6 Model Tuning, Bagging and Boosting.

Random forest is an ensemble technique that combines several base models (decision trees) in order to produce one optimal predictive model.

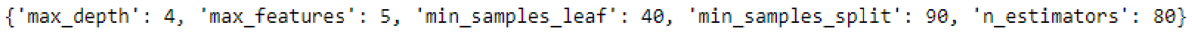
Building model using grid search cross validation method and tuning the hyper parameters in order to obtain a stable random forest.

Grid search CV parameters used:



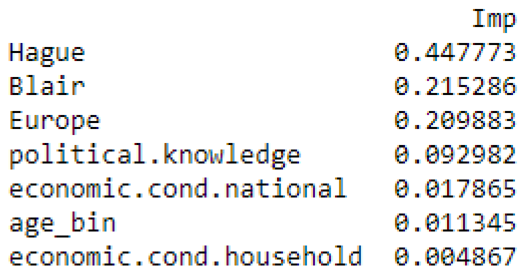
```
GridSearchCV(cv=10, estimator=RandomForestClassifier(random_state=1),
             param_grid={'max_depth': [4, 5, 10], 'max_features': [3, 4, 5],
                         'min_samples_leaf': [30, 40, 50],
                         'min_samples_split': [90, 120, 150],
                         'n_estimators': [60, 80, 100]})
```

Regularised model with pruning parameters:



```
{'max_depth': 4, 'max_features': 5, 'min_samples_leaf': 40, 'min_samples_split': 90, 'n_estimators': 80}
```

Variable importance:

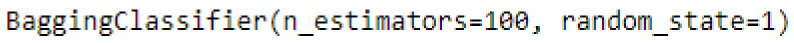


```
                              Imp
Hague                    0.447773
Blair                    0.215286
Europe                   0.209883
political.knowledge      0.092982
economic.cond.national   0.017865
age_bin                  0.011345
economic.cond.household  0.004867
```

Random forest classifier gives high importance to Hague comparatively. Moreover, in this model we could see that age, economic condition household and economic condition national has less importance (<1%).

Bagging model also called bootstrap aggregating, generates new training subsets of original dataset each of the same size by sampling with replacement. This is a parallel model building in which each model can be complex in its own because the overfitting is on the sampled data and not on the main dataset.

Model obtained:

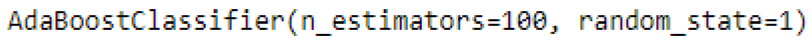


```
BaggingClassifier(n_estimators=100, random_state=1)
```

Boosting methods are a sequential model building technique which combines various weak learners to build a strong model for better prediction.

AdaBoost model, adaptive boosting, here the successive learners are created with a focus on the ill fitted data of the previous learner. Each successive learner focuses more on the harder to fit data which are the residuals of the previous model.

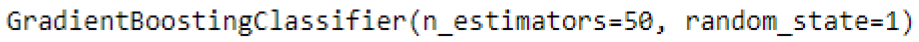
Model obtained:



```
AdaBoostClassifier(n_estimators=100, random_state=1)
```

Gradient Boosting method: Each learner is fit on a modified version of original data which is replaced with the X values and residuals from the previous learner. By fitting new models to the residuals, the overall model performance gradually improves in area where the residuals are initially high.

Model obtained:



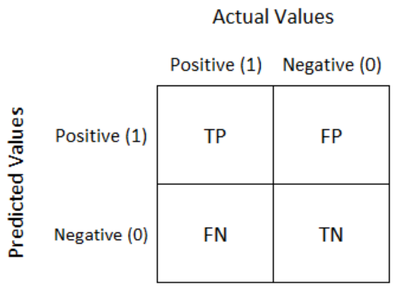
```
GradientBoostingClassifier(n_estimators=50, random_state=1)
```

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 all models on the basis of the performance metrics in a structured tabular manner. Describe on which model is best/optimized

Fifth step of the model is to evaluate it and see how good it will perform for future records.

Some of the model evaluation techniques are:

- Accuracy – how precisely the model classifies the data points.
- Confusion Matrix – 2 * 2 tabular structure reflecting the model performance in four blocks

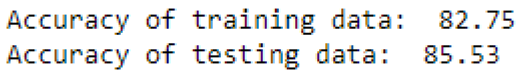


		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

- Receiver operating characteristics (ROC) curve – A technique to visualize classifier performance
- ROC_AUC score – Area under curve, which is by calculating the percentage area below the curve.

Logistic Regression performance metrics:

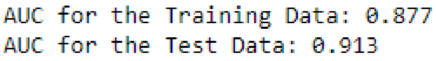
Accuracy score:



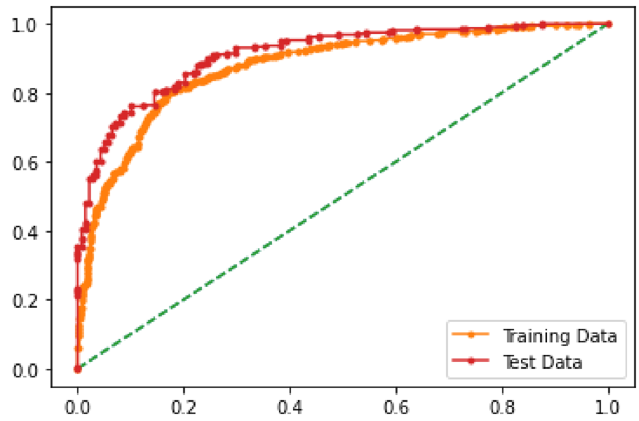
```
Accuracy of training data:  82.75
Accuracy of testing data:  85.53
```

Performance metrics:

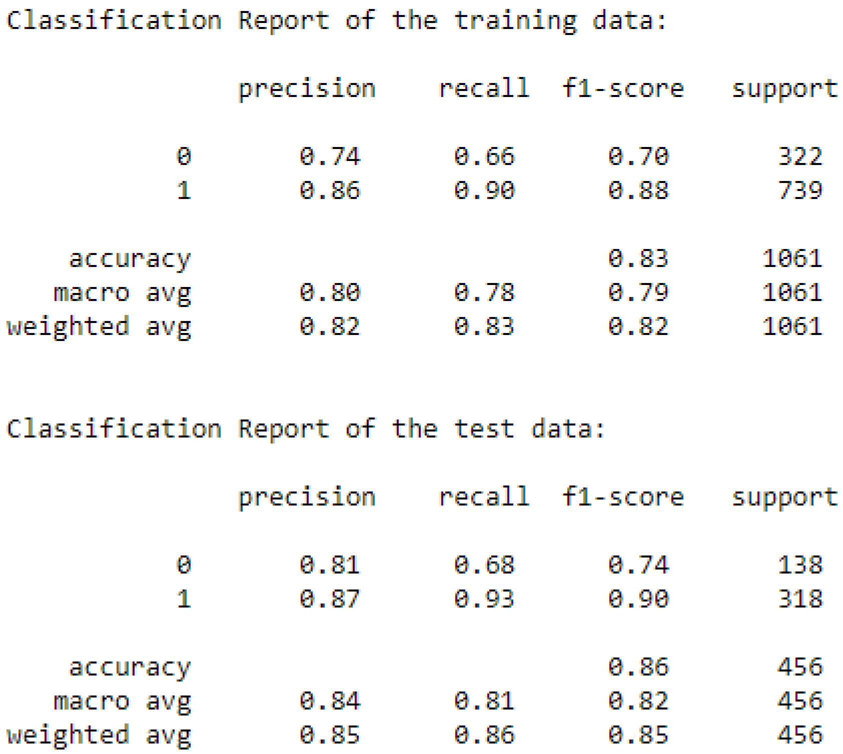
AUC score and ROC-AUC curve:



```
AUC for the Training Data: 0.877
AUC for the Test Data: 0.913
```

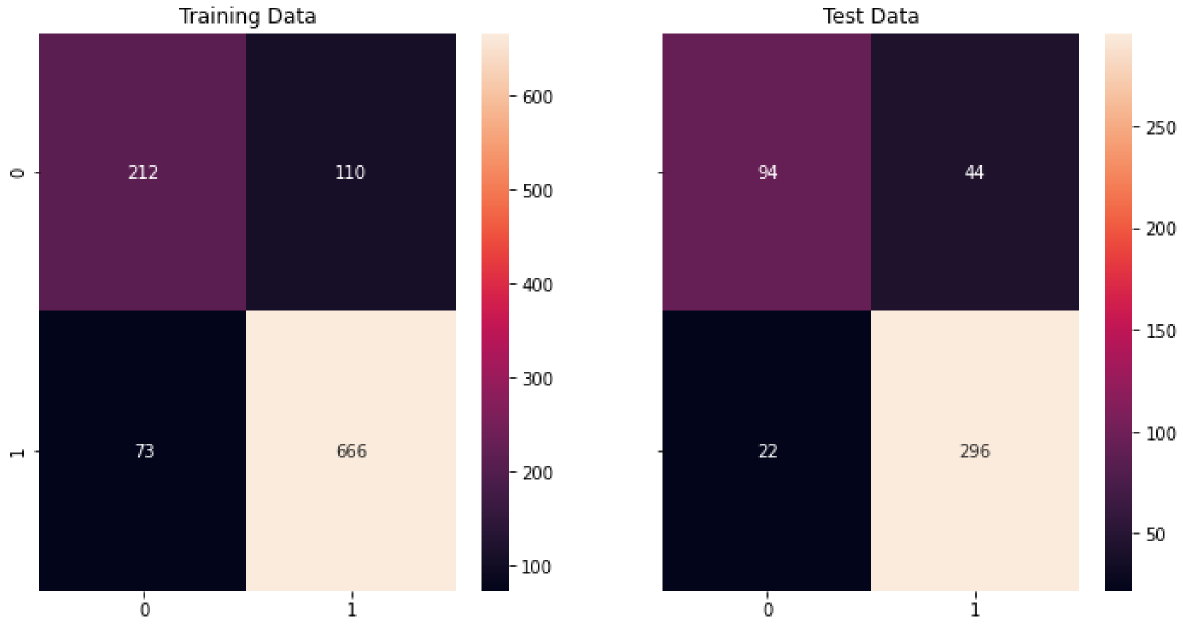


Classification Report:



```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.74      0.66      0.70       322
           1       0.86      0.90      0.88       739

    accuracy                           0.83      1061
   macro avg       0.80      0.78      0.79      1061
weighted avg       0.82      0.83      0.82      1061


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.81      0.68      0.74       138
           1       0.87      0.93      0.90       318

    accuracy                           0.86       456
   macro avg       0.84      0.81      0.82       456
weighted avg       0.85      0.86      0.85       456
```

Confusion matrix:



Class 1: Labour party

Class 0: Conservative party

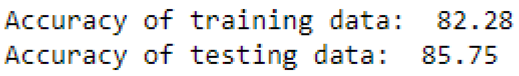
Inference:

Metrics	**Training set**	**Testing set**
Accuracy	0.83	0.86
F1 score for class 0 (Conservative)	0.70	0.74
F1 score for class 1 (Labour)	0.88	0.90

The metrics accuracy has increased in the testing set than training. Since the proportion of the classes (1,0) are not well balanced, accuracy score cannot reliable to check the performance of the model. There is no evidence of over fitting or under fitting in the model. There is an increase in F1 score in testing set than training set. Since both recall and precision are equally important in this case study, F1 score is preferred for evaluating this classification model.

LDA performance metrics:

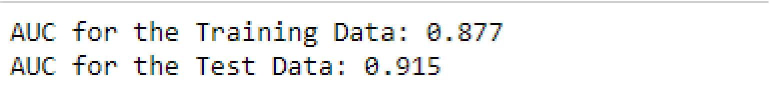
Accuracy score:



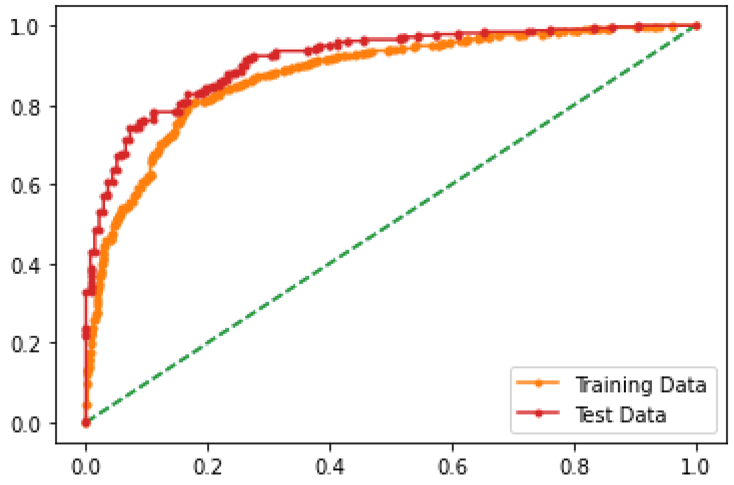
```
Accuracy of training data:  82.28
Accuracy of testing data:  85.75
```

Performance metrics:

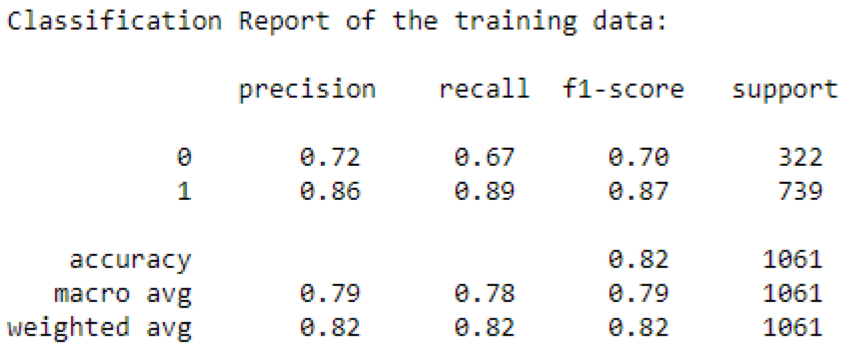
AUC score and ROC-AUC curve:



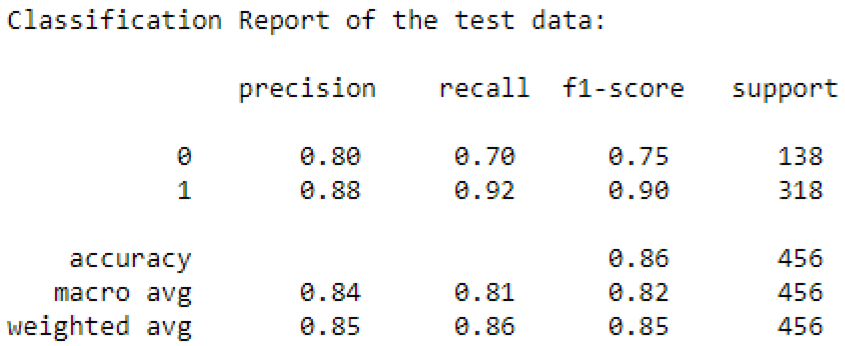
```
AUC for the Training Data: 0.877
AUC for the Test Data: 0.915
```



Classification Report:

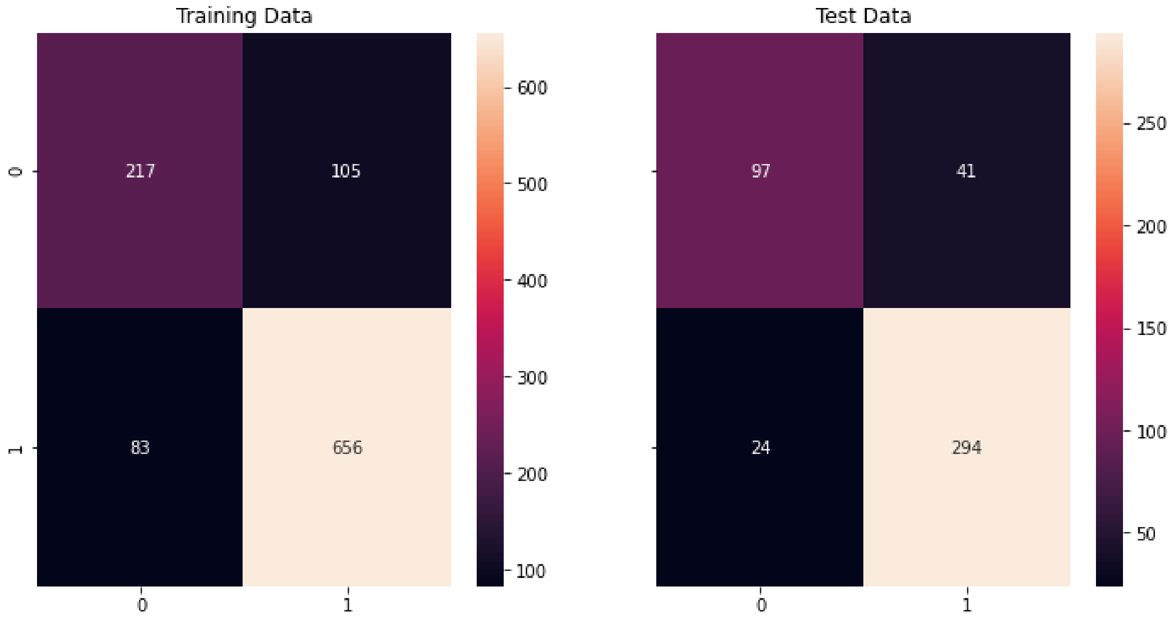


```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.72      0.67      0.70       322
           1       0.86      0.89      0.87       739

    accuracy                           0.82      1061
   macro avg       0.79      0.78      0.79      1061
weighted avg       0.82      0.82      0.82      1061
```



```
Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.80      0.70      0.75       138
           1       0.88      0.92      0.90       318

    accuracy                           0.86       456
   macro avg       0.84      0.81      0.82       456
weighted avg       0.85      0.86      0.85       456
```

Confusion matrix:



Class 1: Labour party

Class 0: Conservative party

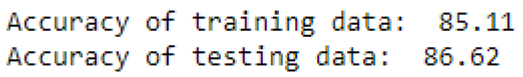
Inference:

Metrics	**Training set**	**Testing set**
Accuracy	0.82	0.86
F1 score for class 0 (Conservative)	0.70	0.75
F1 score for class 1 (Labour)	0.87	0.90

LDA model performance is similar to that of Logistic regression. The metrics accuracy has increased in the testing set than training. There is no evidence of over fitting or under fitting in the model. There is an increase in F1 score in testing set than training set.

KNN model with k=9 performance metrics:

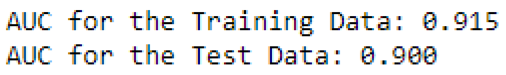
Accuracy score:



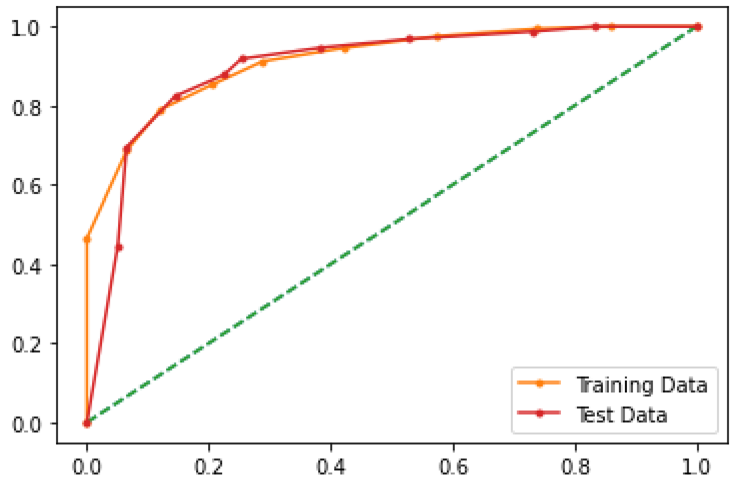
```
Accuracy of training data:  85.11
Accuracy of testing data:  86.62
```

Performance metrics:

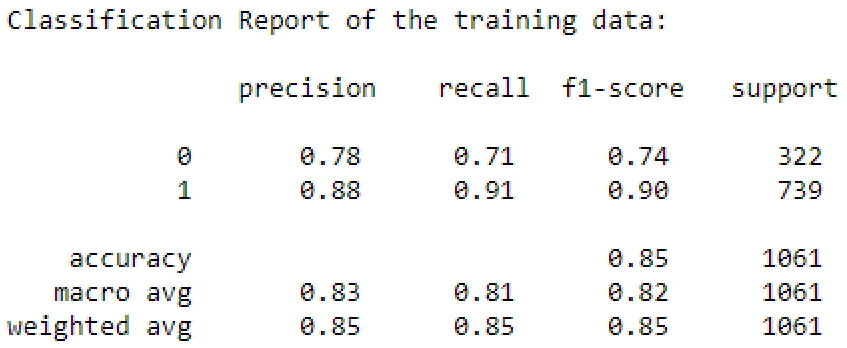
AUC score and ROC-AUC curve:



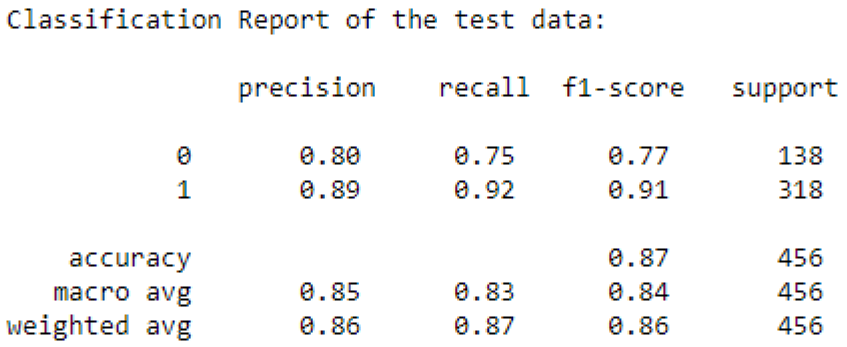
```
AUC for the Training Data: 0.915
AUC for the Test Data: 0.900
```



Classification Report:

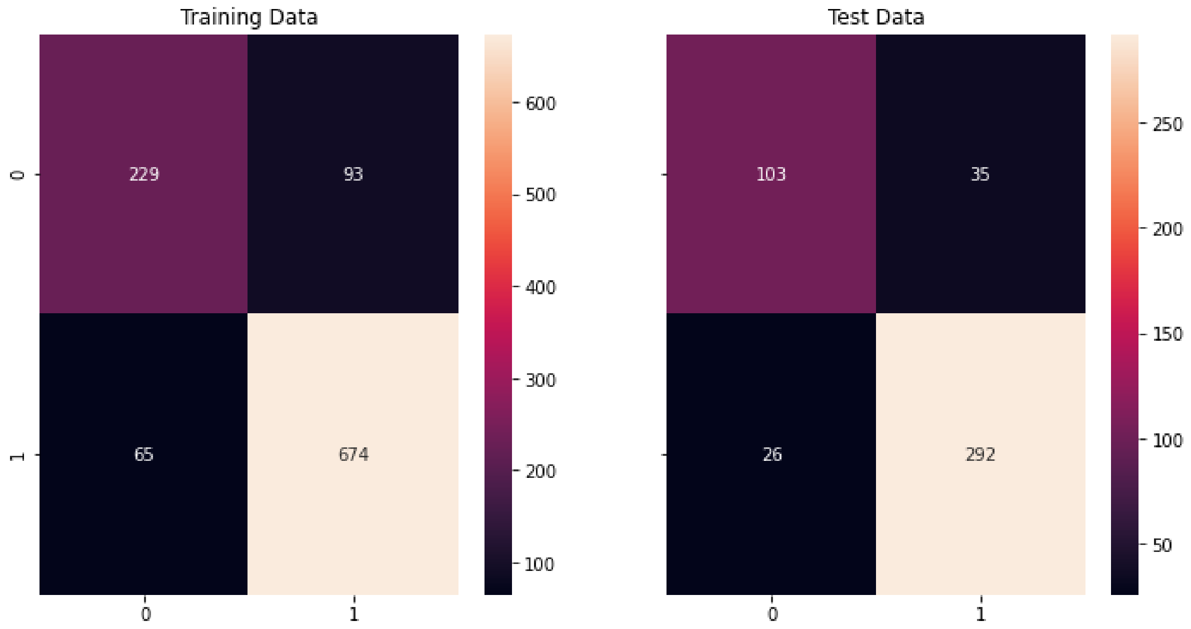


```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.78      0.71      0.74       322
           1       0.88      0.91      0.90       739

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



```
Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.80      0.75      0.77       138
           1       0.89      0.92      0.91       318

    accuracy                           0.87       456
   macro avg       0.85      0.83      0.84       456
weighted avg       0.86      0.87      0.86       456
```

Confusion matrix:



Class 1: Labour party

Class 0: Conservative party

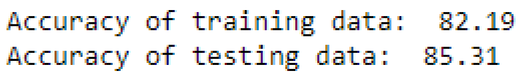
Inference:

Metrics	Training set	Testing set
Accuracy	0.85	0.87
F1 score for class 0 (Conservative)	0.74	0.77
F1 score for class 1 (Labour)	0.90	0.91

KNN model performance are on a better side compared to that of Logistic regression and LDA. Since the metrics vary by 1-2% in training and testing set. Almost similar performance in both training and testing set. The metrics accuracy has increased in the testing set than training. There is no evidence of over fitting or under fitting in the model. There is an increase in F1 score in testing set than training set.

Naïve bayes model performance metrics:

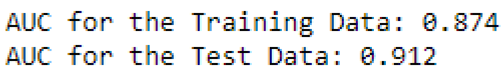
Accuracy score:



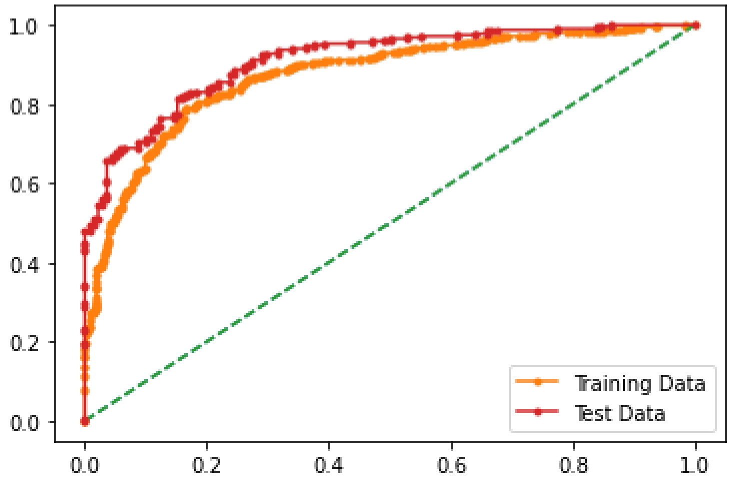
```
Accuracy of training data:  82.19
Accuracy of testing data:  85.31
```

Performance metrics:

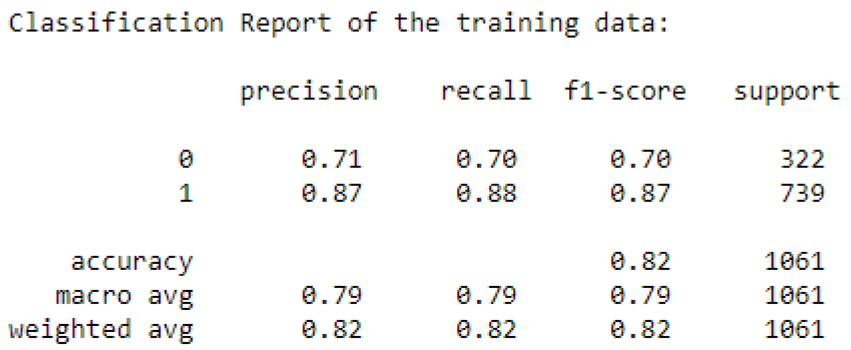
AUC score and ROC-AUC curve:



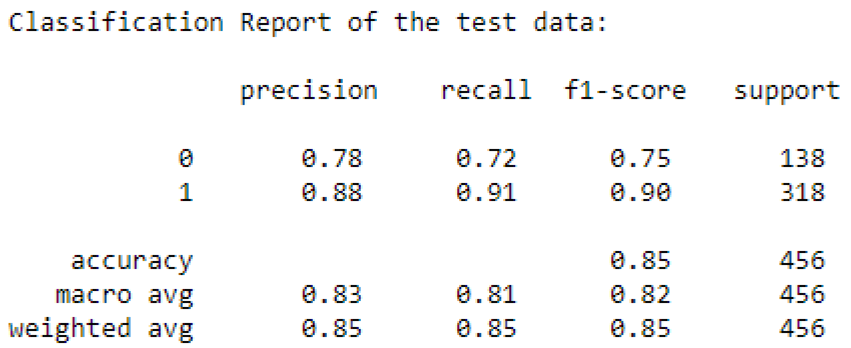
```
AUC for the Training Data: 0.874
AUC for the Test Data: 0.912
```



Classification Report:

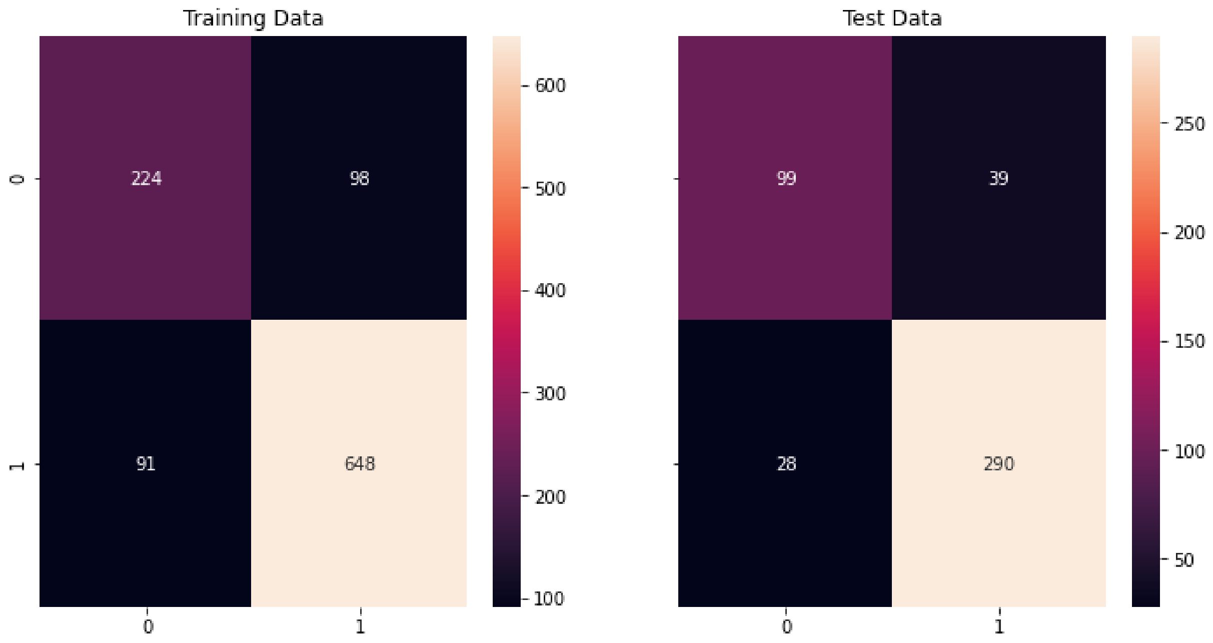


```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.71      0.70      0.70       322
           1       0.87      0.88      0.87       739

    accuracy                           0.82      1061
   macro avg       0.79      0.79      0.79      1061
weighted avg       0.82      0.82      0.82      1061
```



```
Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.78      0.72      0.75       138
           1       0.88      0.91      0.90       318

    accuracy                           0.85       456
   macro avg       0.83      0.81      0.82       456
weighted avg       0.85      0.85      0.85       456
```

Confusion matrix:



Class 1: Labour party

Class 0: Conservative party

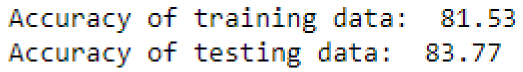
Inference:

Metrics	Training set	Testing set
Accuracy	0.82	0.85
F1 score for class 0 (Conservative)	0.70	0.75
F1 score for class 1 (Labour)	0.87	0.90

Naïve Bayes model performance is similar to that of LDA and Logistic regression. The metrics accuracy has increased in the testing set than training. There is no evidence of over fitting or under fitting in the model. There is an increase in F1 score in testing set than training set.

Random forest model performance metrics:

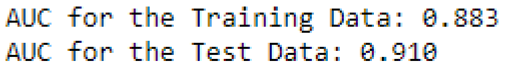
Accuracy score:



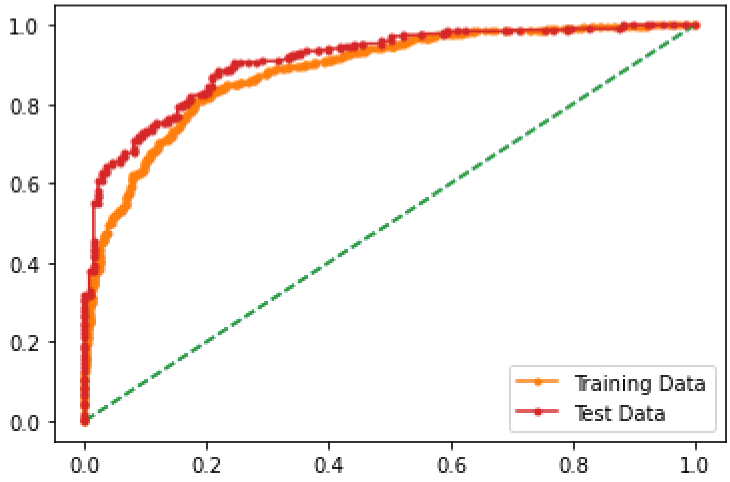
```
Accuracy of training data:  81.53
Accuracy of testing data:  83.77
```

Performance metrics:

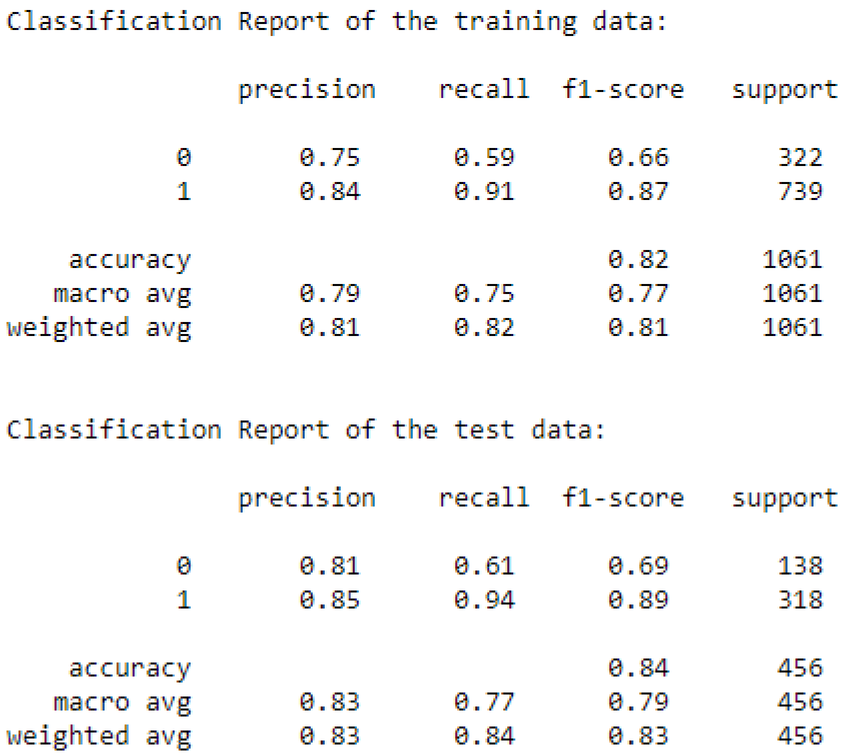
AUC score and ROC-AUC curve:



```
AUC for the Training Data: 0.883
AUC for the Test Data: 0.910
```

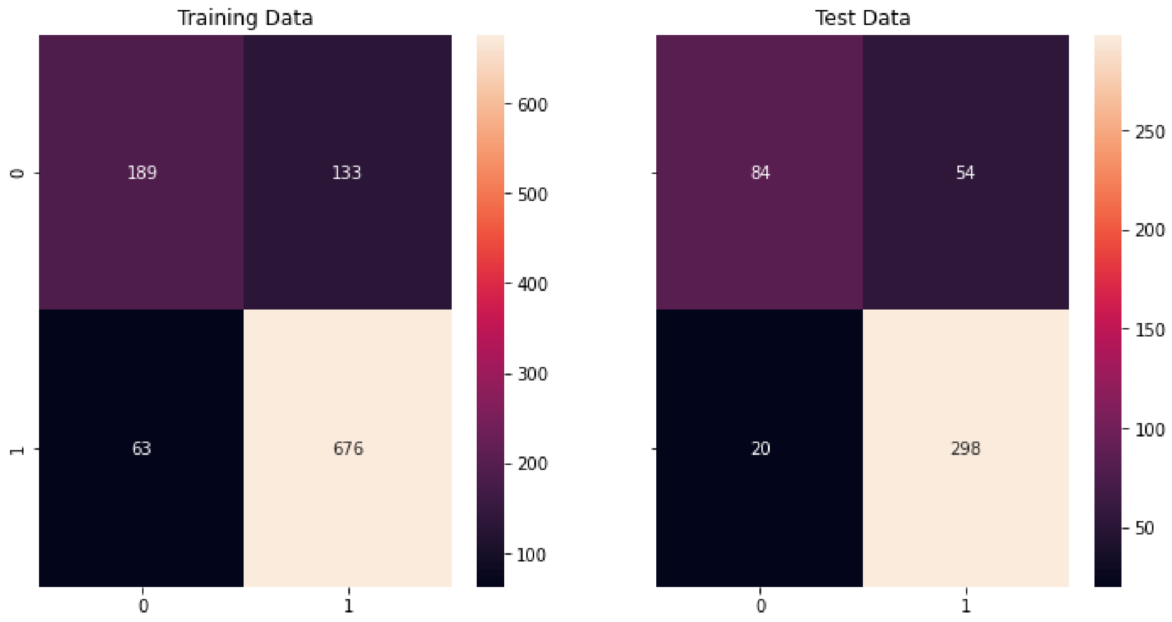


Classification Report:



```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.75      0.59      0.66       322
           1       0.84      0.91      0.87       739

    accuracy                           0.82      1061
   macro avg       0.79      0.75      0.77      1061
weighted avg       0.81      0.82      0.81      1061

Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.81      0.61      0.69       138
           1       0.85      0.94      0.89       318

    accuracy                           0.84       456
   macro avg       0.83      0.77      0.79       456
weighted avg       0.83      0.84      0.83       456
```

Confusion matrix:



Class 1: Labour party

Class 0: Conservative party

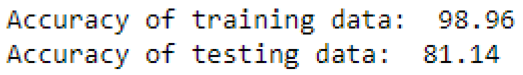
Inference:

Metrics	**Training set**	**Testing set**
Accuracy	0.82	0.84
F1 score for class 0 (Conservative)	0.66	0.69
F1 score for class 1 (Labour)	0.87	0.89

Random forest model performance is comparatively poor compared to other models before. Almost similar performance in both training and testing set but lesser than previous models. The metrics accuracy has increased in the testing set than training. There is no evidence of over fitting or under fitting in the model. There is an increase in F1 score in testing set than training set. Random forest classifying class 0 (conservative) as class 1 (labour) is high, since this prediction of False positives is also important for this case study, we can infer that RF is not doing a better job in classifying.

Bagging method performance metrics:

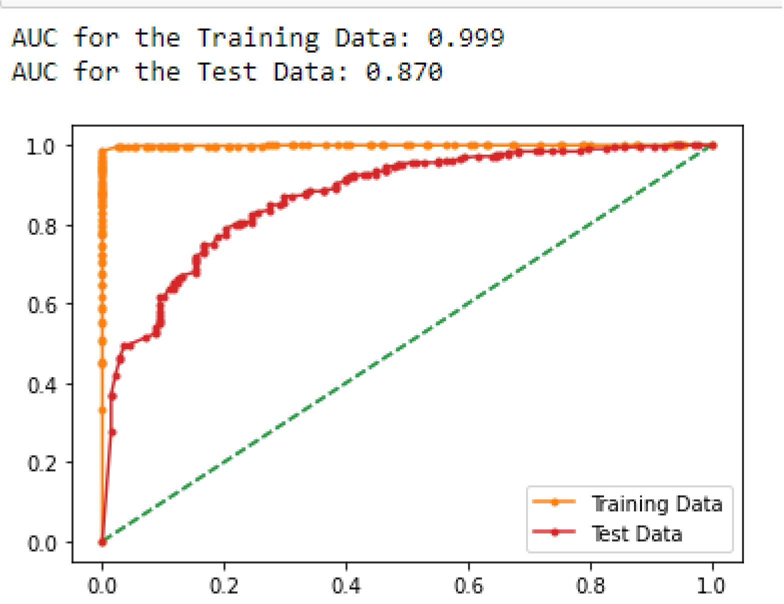
Accuracy score:



```
Accuracy of training data:  98.96
Accuracy of testing data:  81.14
```

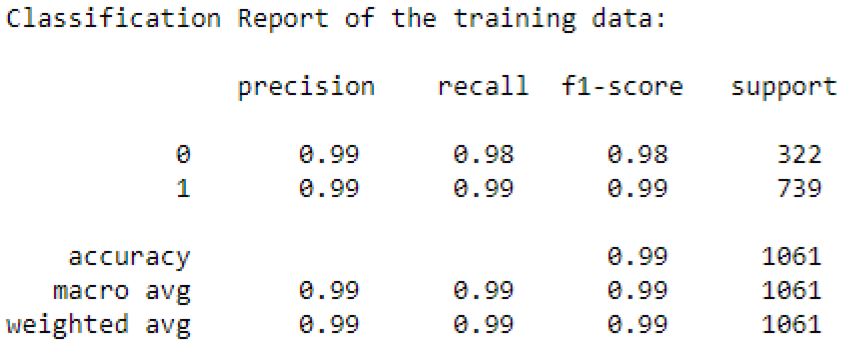
Performance metrics:

AUC score and ROC-AUC curve:

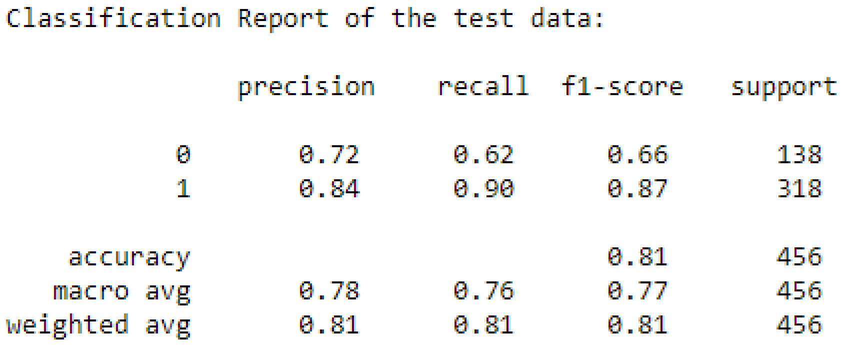


```
AUC for the Training Data: 0.999
AUC for the Test Data: 0.870
```

Classification Report:

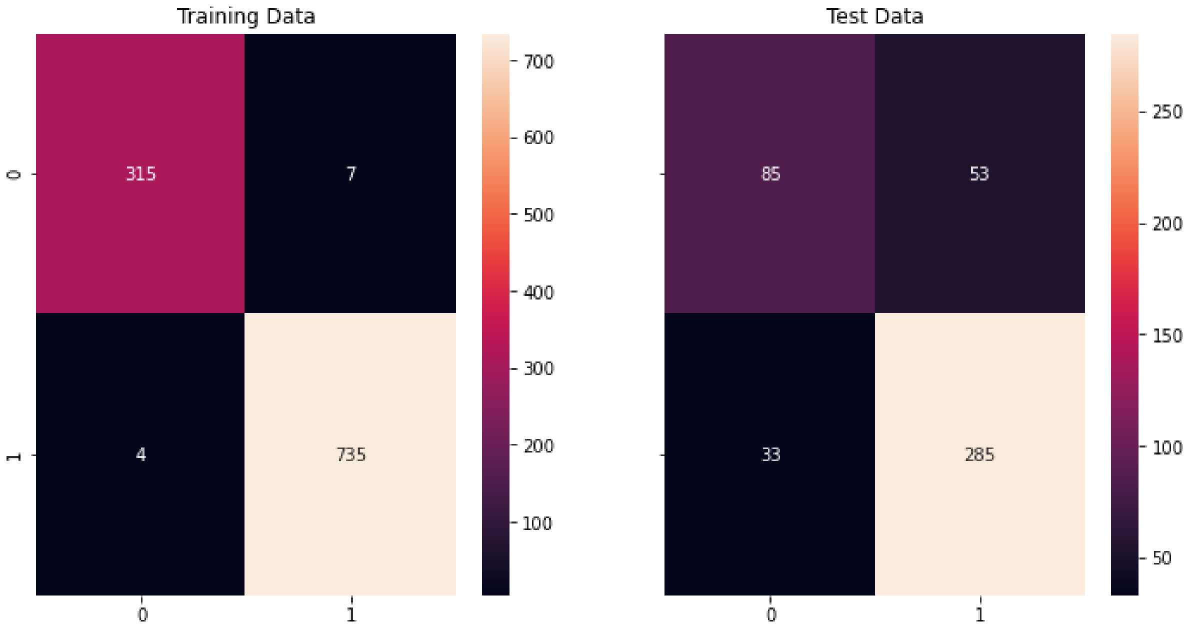


```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.99      0.98      0.98       322
           1       0.99      0.99      0.99       739

    accuracy                           0.99      1061
   macro avg       0.99      0.99      0.99      1061
weighted avg       0.99      0.99      0.99      1061
```



```
Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.72      0.62      0.66       138
           1       0.84      0.90      0.87       318

    accuracy                           0.81       456
   macro avg       0.78      0.76      0.77       456
weighted avg       0.81      0.81      0.81       456
```

Confusion matrix:



Class 1: Labour party

Class 0: Conservative party

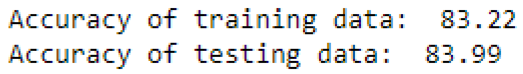
Inference:

Metrics	**Training set**	**Testing set**
Accuracy	0.99	0.81
F1 score for class 0 (Conservative)	0.98	0.66
F1 score for class 1 (Labour)	0.99	0.87

Bagging model shows a classic output of overfitted model where the model performs well in training data and under performs in testing data. Random forest which is also a type of type of bagging method with tuned hyper parameters has also shown less performance in classifying for the given case study.

Ada boost method performance metrics:

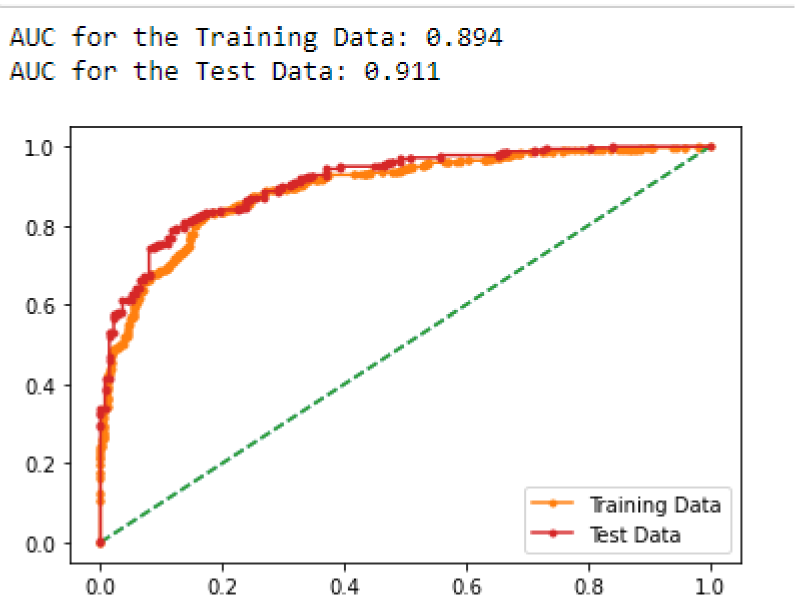
Accuracy score:



```
Accuracy of training data:  83.22
Accuracy of testing data:  83.99
```

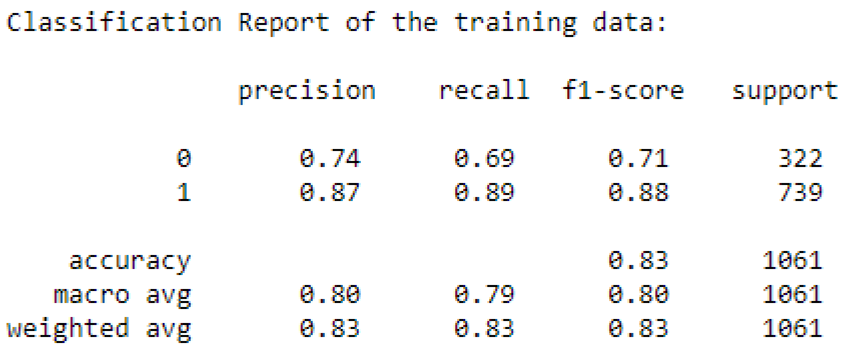
Performance metrics:

AUC score and ROC-AUC curve:

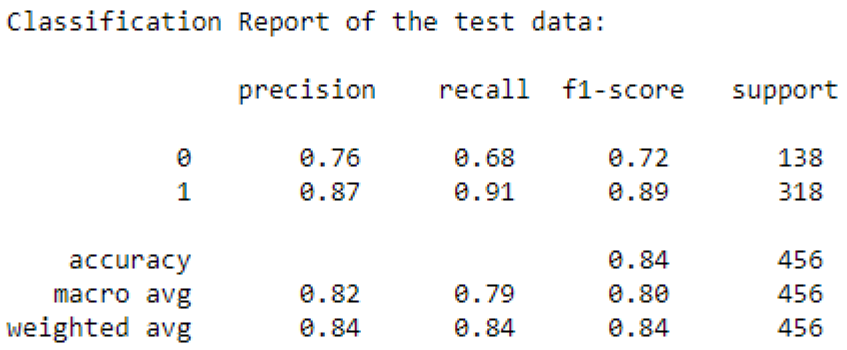


```
AUC for the Training Data: 0.894
AUC for the Test Data: 0.911
```

Classification Report:

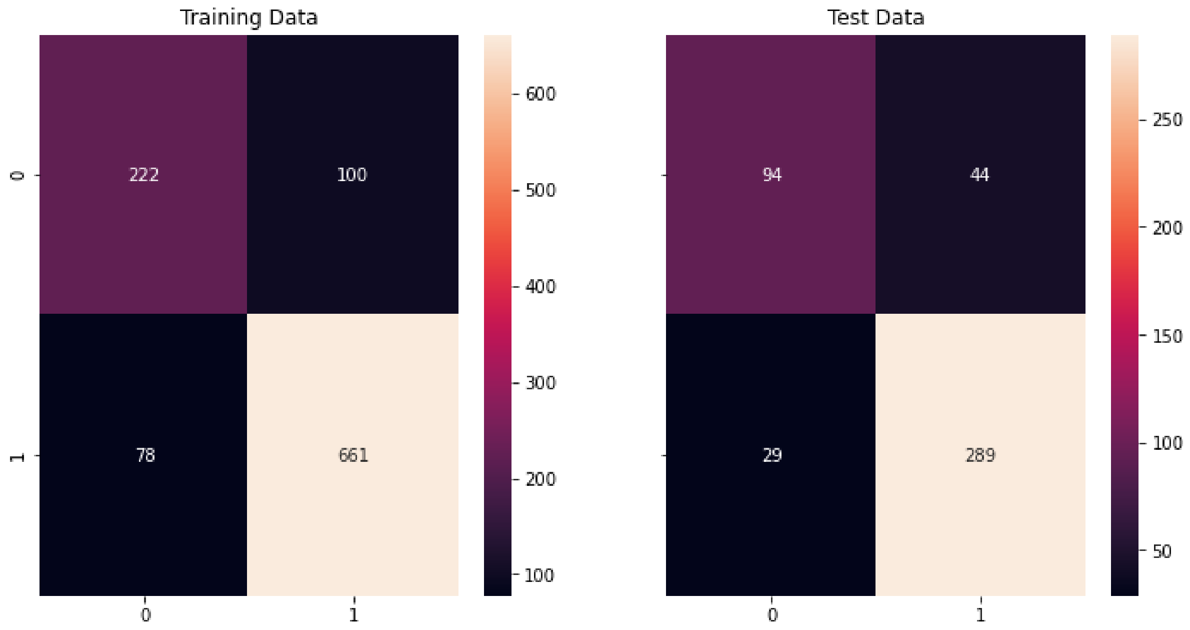


```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.74      0.69      0.71       322
           1       0.87      0.89      0.88       739

    accuracy                           0.83      1061
   macro avg       0.80      0.79      0.80      1061
weighted avg       0.83      0.83      0.83      1061
```



```
Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.76      0.68      0.72       138
           1       0.87      0.91      0.89       318

    accuracy                           0.84       456
   macro avg       0.82      0.79      0.80       456
weighted avg       0.84      0.84      0.84       456
```

Confusion matrix:



Class 1: Labour party

Class 0: Conservative party

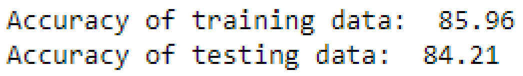
Inference:

Metrics	**Training set**	**Testing set**
Accuracy	0.83	0.84
F1 score for class 0 (Conservative)	0.71	0.72
F1 score for class 1 (Labour)	0.88	0.89

Ada boost model also gives a similar output as that of LDA and logistic regression. Almost similar performance in both training and testing set. The metrics accuracy has increased in the testing set than training. There is no evidence of over fitting or under fitting in the model. There is an increase in F1 score in testing set than training set.

Gradient boost method performance metrics:

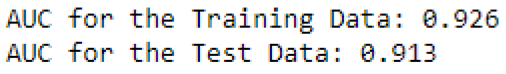
Accuracy score:



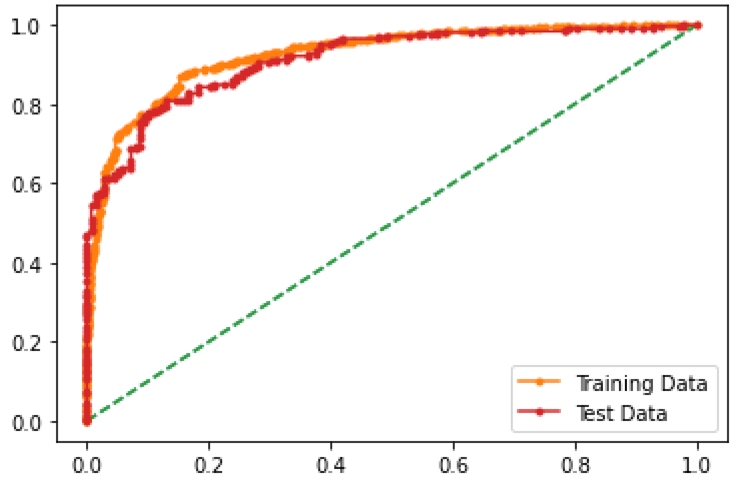
```
Accuracy of training data:  85.96
Accuracy of testing data:  84.21
```

Performance metrics:

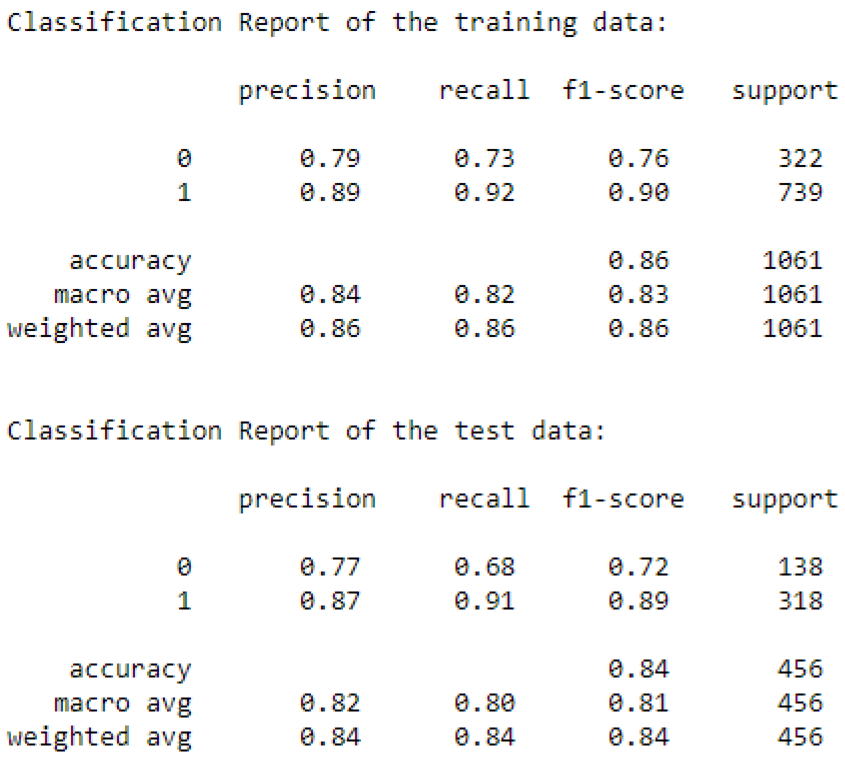
AUC score and ROC-AUC curve:



```
AUC for the Training Data: 0.926
AUC for the Test Data: 0.913
```

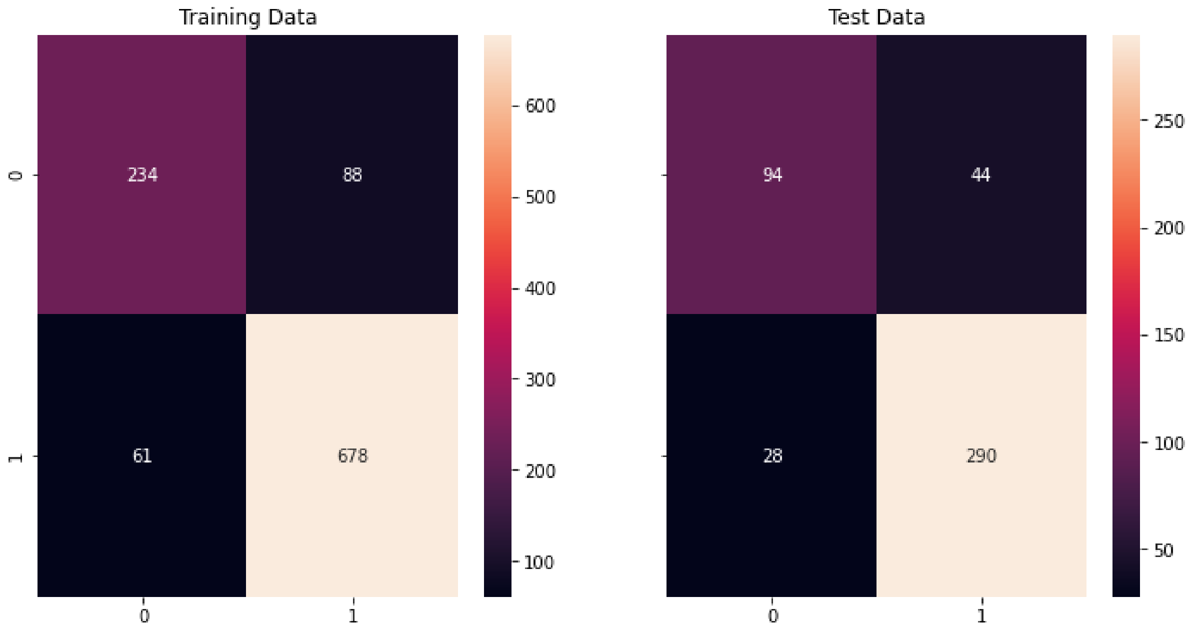


Classification Report:



```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.79      0.73      0.76       322
           1       0.89      0.92      0.90       739

    accuracy                           0.86      1061
   macro avg       0.84      0.82      0.83      1061
weighted avg       0.86      0.86      0.86      1061


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.77      0.68      0.72       138
           1       0.87      0.91      0.89       318

    accuracy                           0.84       456
   macro avg       0.82      0.80      0.81       456
weighted avg       0.84      0.84      0.84       456
```

Confusion matrix:



Class 1: Labour party

Class 0: Conservative party

Inference:

Metrics	**Training set**	**Testing set**
Accuracy	0.86	0.84
F1 score for class 0 (Conservative)	0.76	0.72
F1 score for class 1 (Labour)	0.90	0.89

Gradient boost model also gives a similar output as that of KNN model. Almost similar performance in both training and testing set. The metrics accuracy has decreased in the testing set than training. There is no evidence of over fitting or under fitting in the model. There is a decrease in F1 score in testing set than training set but it is within the acceptable range.

Best model for the given case study:

From the above results,

- Logistic regression, LDA and Naïve bayes model performance is similar. The metrics accuracy has increased in the testing set than training. There is no evidence of over fitting or under fitting in the model.

- Random forest and Bagging model performance are less. Bagging for this case study is not preferred because of the overfit.

- Both the methods in boosting offers a good performance in case of identifying the classes correctly. Gradient and Ada boosting shows same performance in testing test.

- KNN model with k=9 seems to be the optimized model for the given election dataset. Since in this case study all the variables are scalable, calculating distance between two data points becomes easier. Moreover, the number of data points are less. In testing set, the number of false positives and false negatives are less compared to other models. Considering the problem statement, KNN model suits well.

Moreover, the training and test set results are in line for **KNN model** rather than other models. Because the small amount of data and just two classes in the predictor variable, choosing KNN as the final optimised model.

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.

The goal of the business problem is to create an exit poll, predict overall win and seats covered by particular party. Since the answers of exit polls are almost confirmed, they are considered to be genuine sources to estimate the results of the elections. Exit polls can also impact the opinions of those who are still to vote since they are post-election polls. Knowing whether the voter will choose to vote for Labour/Conservative will help the news channel CNBE to stay ahead of their competitors in releasing the election results and by predicting the seats covered by each party will be a way for the news channel to focus on the winning party.

Using the given dataset provided by the company, models like LDA, logistic regression, KNN, Naïve bayes, Random forest, bagging and boosting models were built and evaluated the performance metrics.

Here the target variable is 'vote' (1 – Labour/2 – Conservative). Factors that are of significant importance in classifying the target variable are Hague, Blair, Europe, political knowledge. Rest of the factors have very little importance.

Assumption made for this case study:

- Exit polls are usually conducted at the end of day/after each voter have casted their vote. Making an assumption that the given dataset is for the Day 1 of the election.
- Entry age of people to cast their vote as 21.
- Data is from one voting centre.

Notable inferences from the analysis:

- Female voters are on the higher side comparatively.
- As the age increases, voters tend to go for conservative party. Assuming that the eligible age for casting vote as 21, we can see that there are less voters in that category below 30s.
- Higher the Eurosceptic sentiment among voters, conservative party gets the vote. Lower the Eurosceptic sentiment, labour party gets the vote. Voters for labour party are in favour of the integration of Europe and Britain.
- Seems like the voters are clear on what they want and which party to vote for, from the assessment score they have given for the respective parties. Higher assessment score for Blair and lower assessment score for Hague, Labour party votes increases.

Insights from the model:

- 70% of the data points are in 'Labour' vote category and 30% of the data points are in 'Conservative' voting category. Since the data points are not well balanced between the categories, model accuracy score cannot be reliable performance measure.
- Since both the classes predictions are important, we are using F1 score as a performance metric which gives a balance between recall and precision.

- Out of all the models, KNN has performed consistently with both training and testing data.

Our business problem is to predict which party a voter will vote for.

- False positive (FP) - Datapoints that are actually false but predicted as true. This is also known as type 1 error. In order to reduce the type 1 error, we have to increase the precision of the model (among the points identified as positives by the model how many are actually positive).

  > Type 1 error in this case study means model has classified the data point as Labour party instead of Conservative party.

- False negative (FN) – Datapoints that are actually true but predicted as false. This is known as type 2 error. In order to reduce to type 2 error. We have to increase recall (how many actual true data points are identified as true by the model)

  > Type 2 error for our case study means the model has classified the data point as Conservative party instead of Labour party.

Both False positives and false negatives are important for this election case study since the goal of the problem is to correctly identity which party the vote goes to. Since there is a trade off between this FP and FN, F1 score is considered which balances the trade-off. If precision is low, the F1 is low and if the recall is low again F1 score will be low.

Correctly predicting the vote will improve the overall accuracy as well.

KNN model gives an F1 score of 0.77 for conservative class and 0.91 for labour class. Given that there are more data points with labour votes, F1 score for that class is higher.

Recommendations for the business:

Goal of the news channel is to telecast the right information to their viewers. In order to predict the votes correctly, exit poll should contain questions which can determine the votes. It becomes crucial in order to estimate the results of the election.

Assuming it's the day 1 of elections, with the data collected we can say that Labour party is in the lead and that the party is in favour of the integration of Europe and Britain. Voters who chose this party are also aware of the integration. However, if we had data from all centres and from different locations, this prediction can change. The KNN model built using the dataset will be effective even for new data coming in.

Here we have built model with 7 independent variables for predicting the 'vote' dependent variable. Exit polls have consistently been reliable sources to gather demographic information of voters. Since voting is always anonymous, exit polls are the sole source to gather demographic details and other information such as reasons why the voter voted for a particular candidate.

- Including some more information like education, income, family status, living status, religion, employment status would be even more helpful in order to weigh the sentiment of the voters with respect to the party they might vote for.

- Based on the assumption made, conducting this exit poll for the entire course of the election by the news channel will be effective. Any lapse in time could lead to highly biased data.

- Collecting exit polls from various platforms like online, form filling can increase the accuracy of predictions.

- In case a few voters refuse participation, the researcher can collect their details such as sex and approximate age of the voter. These values can be considered while statistical analysis of collected data to reduce response bias as much as possible.

Problem 2:

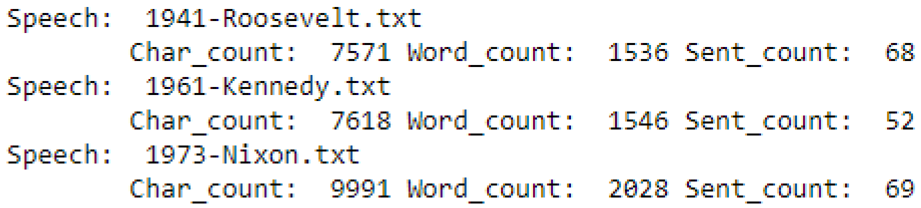
Text Mining

Problem statement:

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

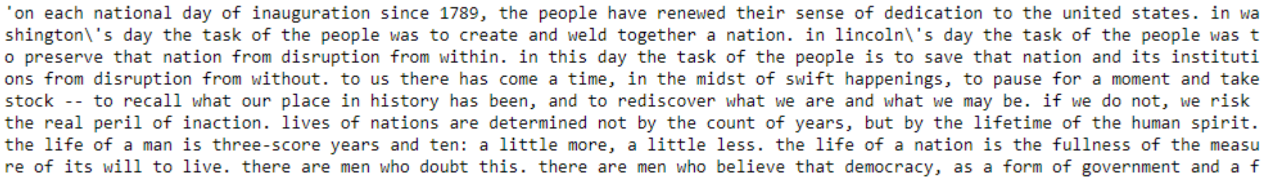
2.1 Find the number of characters, words and sentences for the mentioned documents. (Hint: use .words(), .raw(), .sent() for extracting counts)



```
Speech:  1941-Roosevelt.txt
        Char_count:  7571 Word_count:  1536 Sent_count:  68
Speech:  1961-Kennedy.txt
        Char_count:  7618 Word_count:  1546 Sent_count:  52
Speech:  1973-Nixon.txt
        Char_count:  9991 Word_count:  2028 Sent_count:  69
```

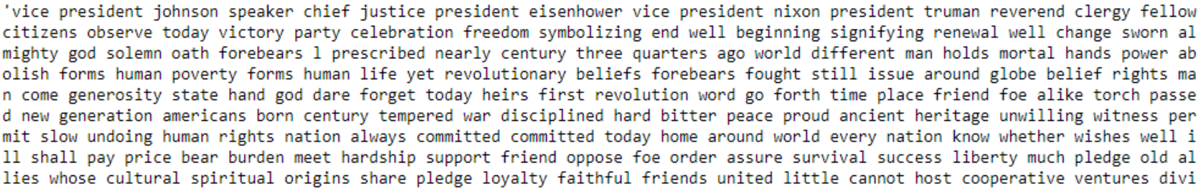
2.2 Remove all the stop words from the three speeches.

Before removing stop words and punctuations, converting the text to lower case.



```
'on each national day of inauguration since 1789, the people have renewed their sense of dedication to the united states. in wa
shington\'s day the task of the people was to create and weld together a nation. in lincoln\'s day the task of the people was t
o preserve that nation from disruption from within. in this day the task of the people is to save that nation and its instituti
ons from disruption from without. to us there has come a time, in the midst of swift happenings, to pause for a moment 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 not, we risk
the real peril of inaction. lives of nations are determined not by the count of years, but by the lifetime of the 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 fullness of the measu
re of its will to live. there are men who doubt this. there are men who believe that democracy, as a form of government and a f
```

Created a dataset with the three speeches and removed numbers, punctuations and stop words.

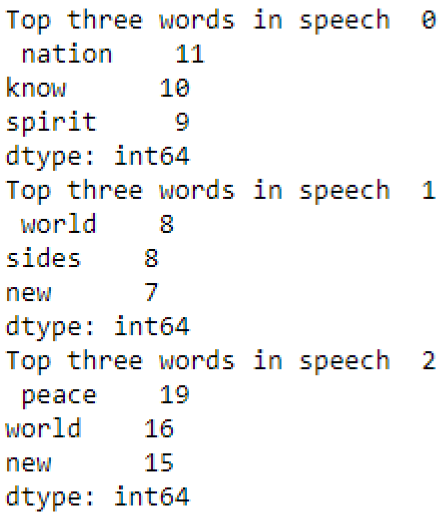


```
'vice president johnson speaker chief justice president eisenhower vice president nixon president truman reverend clergy fellow
citizens observe today victory party celebration freedom symbolizing end well beginning signifying renewal well change sworn al
mighty god solemn oath forebears l prescribed nearly century three quarters ago world different man holds mortal hands power ab
olish forms human poverty forms human life yet revolutionary beliefs forebears fought still issue around globe belief rights ma
n come generosity state hand god dare forget today heirs first revolution word go forth time place friend foe alike torch passe
d new generation americans born century tempered war disciplined hard bitter peace proud ancient heritage unwilling witness per
mit slow undoing human rights nation always committed committed today home around world every nation know whether wishes well i
ll shall pay price bear burden meet hardship support friend oppose foe order assure survival success liberty much pledge old al
lies whose cultural spiritual origins share pledge loyalty faithful friends united little cannot host cooperative ventures divi
```

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)

Most Frequent words in the three speeches:

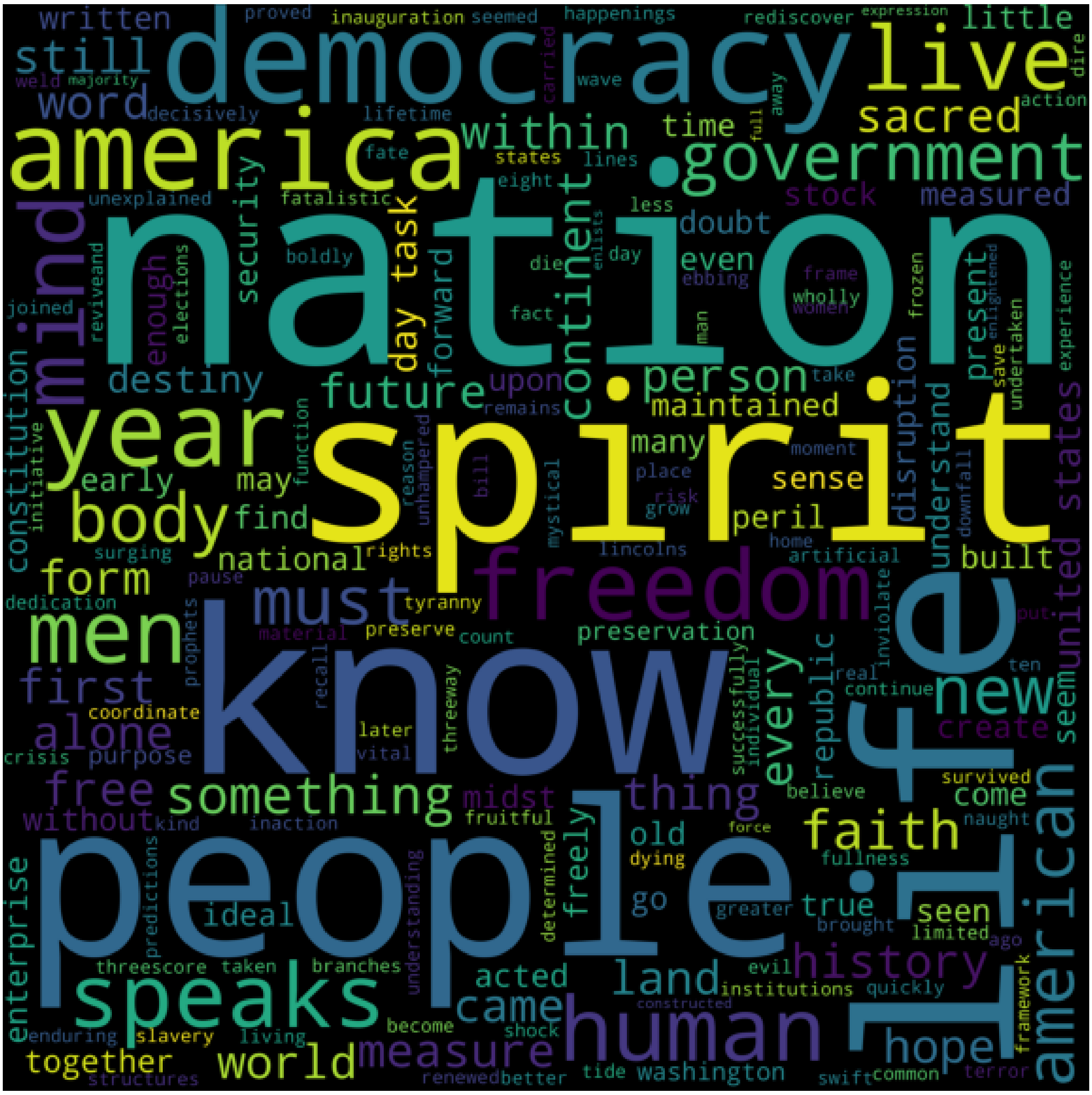
Speech 0: 1941-Roosevelt.txt
Speech 1: 1961-Kennedy.txt
Speech 2: 1973-Nixon.txt



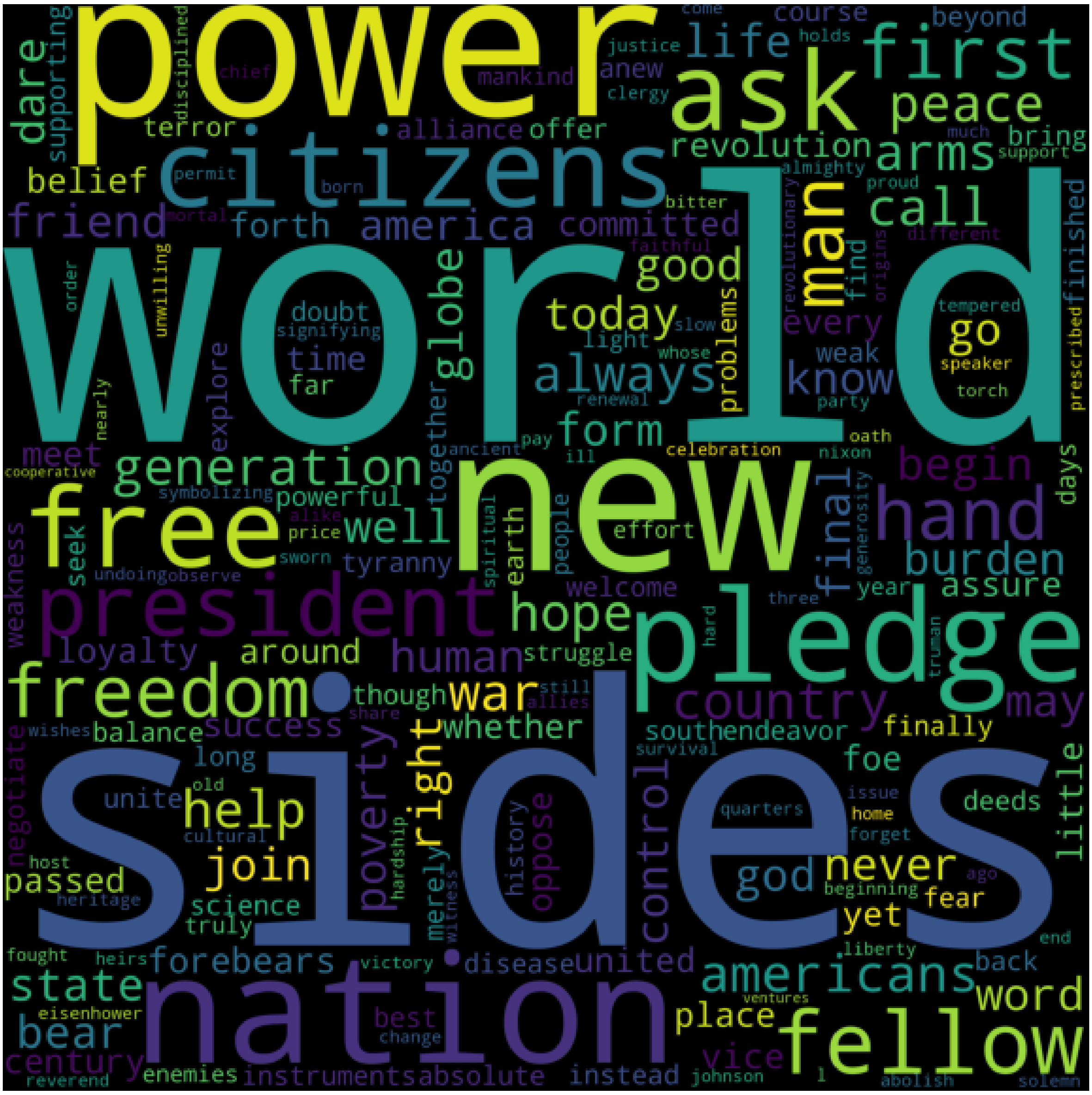
```
Top three words in speech  0
 nation    11
know      10
spirit     9
dtype: int64
Top three words in speech  1
 world    8
sides    8
new      7
dtype: int64
Top three words in speech  2
 peace    19
world    16
new      15
dtype: int64
```

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

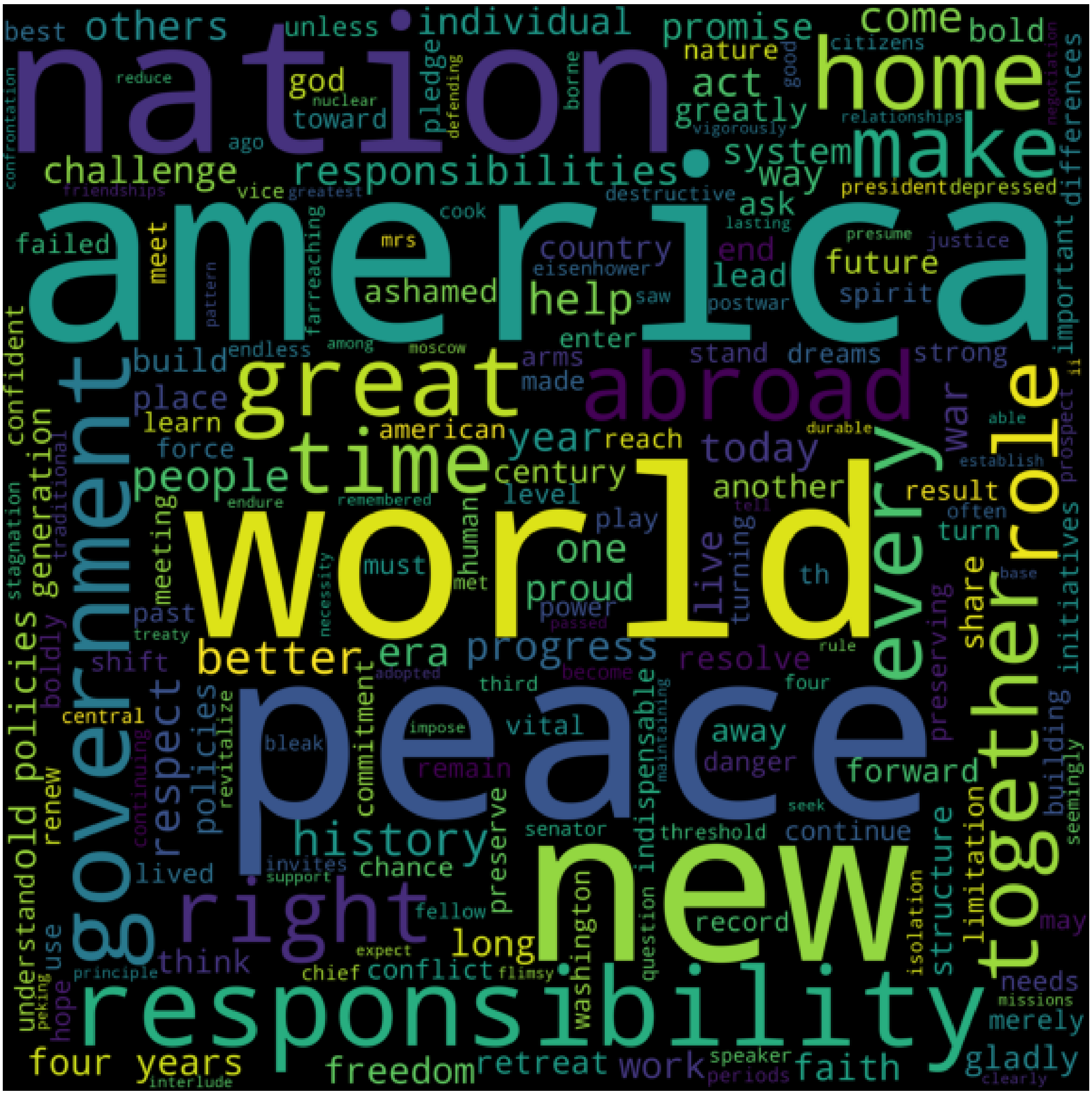
Word Cloud for 1941-Roosevelt.txt



Word cloud for 1961-Kennedy.txt



Word cloud for 1973-Nixon.txt


america
world
peace
nation
new
responsibility
government
great
time
abroad
right
home
make
together
every
role
responsibilities
challenge
history
best
others
unless
individual
promise
come
bold
god
pledge
nature
act
greatly
citizens
system
way
toward
ago
nuclear
reduce
defending
borne
good
relationships
vigorously
differences
negotiation
confrontation
friendships
vice
greatest
destructive
ask
president
depressed
cook
lasting
presume
justice
failed
meet
mrs
farreaching
country
eisenhower
end
lead
future
help
ashamed
saw
postwar
spirit
important
pattern
enter
endless
among
moscow
arms
stand
dreams
strong
build
made
place
learn
confident
american
year
reach
today
durable
able
force
people
war
establish
prospect
ii
century
level
another
result
often
endure
remembered
tell
turn
generation
traditional
stagnation
meeting
human
play
live
turning
must
one
met
th
share
initiatives
base
proud
preserving
power
necessity
past
passed
treaty
rule
better
era
progress
resolve
shift
boldly
policies
adopted
become
third
four
building
central
commitment
indispensable
revitalize
impose
vital
away
respect
bleak
remain
danger
forward
maintaining
continuing
renew
seek
seemingly
senator
threshold
continue
structure
limitation
preserve
question
invites
chance
support
lived
understand
old
washington
isolation
fellow
record
may
use
expect
long
think
principle
chief
conflict
flimsy
needs
peking
hope
missions
merely
four years
freedom
retreat
work
speaker
faith
gladly
interlude
periods
clearly