

Machine Learning Project Report

Nandha Keshore Utti

PG-DSBA Online

March' 22

Date: 25/09/2022

Contents

Problem 1	5
1.1. Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it. (4 Marks)	5
1.2. Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.	10
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).	19
1.4. Apply Logistic Regression and LDA (linear discriminant analysis)	21
1.5. Apply KNN Model and Naïve Bayes Model. Interpret the results.	22
1.6. Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting.	23
1.7. Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized.	28
1.8. Based on these predictions, what are the insights?	40
Problem 2	41
2.1. Find the number of characters, words, and sentences for the mentioned documents.	41
2.2. Remove all the stopwords from all three speeches.	42
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)	46
2.4. Plot the word cloud of each of the speeches of the variable. (after removing the stopwords).	46

List of figures

Fig.1 – Problem-1: Hist plot of all numeric variables	10
Fig.2 – Problem-1: Box plot of all numerical variables	11
Fig.3 – Problem-1: Count plots of categorical variables	12
Fig.4 – Problem-1: Count plot of Blair	13
Fig.5 – Problem-1: Count plot of Hague	13
Fig.6 – Problem-1: Pair plot of all numeric variables	14
Fig.7 – Problem-1: Heatmap of all numeric variables	15
Fig.8 – Problem-1: Count plot of vote for gender	16
Fig.9 – Problem-1: Count plot of vote for economic.cond.national	16
Fig.10 – Problem-1: Count plot of vote for economic.cond.household	17
Fig.11 – Problem-1: Count plot of vote for Europe	18
Fig.12 – Problem-1: Count plot of vote for political.knowledge	19
Fig.13 – Problem-1: Train and Test data set shapes	20
Fig.14 – Problem-1: Train and Test data set split percentages	20
Fig.15 – Problem-1: Logistic Regression model after fitting on the train data	21
Fig.16 – Problem-1: LDA model after fitting on the train data	22

Fig.17 – Problem-1: Naïve Bayes model after fitting on the train data	22
Fig.18 – Problem-1: KNN model after fitting on the train data	23
Fig.19 – Problem-1: Bagging model after fitting on the train data	23
Fig.20 – Problem-1: Boosting model after fitting on the train data	24
Fig.21 – Problem-1: GridsearchCV Logistic Regression model after fitting on the train data	25
Fig.22 – Problem-1: Best parameters and estimator of GridsearchCV Logistic Regression model	25
Fig.23 – Problem-1: GridsearchCV LDA model after fitting on the train data	25
Fig.24 – Problem-1: Best parameters and estimator of GridsearchCV LDA model	26
Fig.25 – Problem-1: GridsearchCV Naïve Bayes model after fitting on the train data	26
Fig.26 – Problem-1: Best parameters and estimator of GridsearchCV Naïve Bayes	
Fig.27 – Problem-1: GridsearchCV KNN model after fitting on the train data	26
Fig.28 – Problem-1: Best parameter and estimator of GridsearchCV KNN model	26
Fig.29 – Problem-1: GridsearchCV Bagging model after fitting on the train data	27
Fig.30 – Problem-1: Best parameter and estimator of GridsearchCV Bagging model	27
Fig.31 – Problem-1: GridsearchCV Boosting model after fitting on the train data	27
Fig.32 – Problem-1: Best parameter and estimator of GridsearchCV Boosting model	27
Fig.33 – Problem-1: Confusion matrix of train data set from Logistic Regression model	28
Fig.34 – Problem-1: ROC curve of train data set from Logistic Regression model	28
Fig.35 – Problem-1: Confusion matrix of test data set from Logistic Regression model ...	29
Fig.36 – Problem-1: ROC curve of test data set from Logistic Regression model	29
Fig.37 – Problem-1: Confusion matrix of train data set from LDA model	30
Fig.38 – Problem-1: ROC curve of train data set from LDA model	30
Fig.39 – Problem-1: Confusion matrix of test data set from LDA model ...	30
Fig.40 – Problem-1: ROC curve of test data set from LDA model	31
Fig.41 – Problem-1: Confusion matrix of train data set from Naïve Bayes model	31
Fig.42 – Problem-1: ROC curve of train data set from Naïve Bayes model	32
Fig.43 – Problem-1: Confusion matrix of test data set from Naïve Bayes model ...	32
Fig.44 – Problem-1: ROC curve of test data set from Naïve Bayes model	33
Fig.45 – Problem-1: Confusion matrix of train data set from KNN model	33
Fig.46 – Problem-1: ROC curve of train data set from KNN model	34
Fig.47 – Problem-1: Confusion matrix of test data set from KNN model ...	34
Fig.48 – Problem-1: ROC curve of test data set from KNN model	35
Fig.49 – Problem-1: Confusion matrix of train data set from Bagging model	35
Fig.50 – Problem-1: ROC curve of train data set from Bagging model	36
Fig.51 – Problem-1: Confusion matrix of test data set from Bagging model ...	36
Fig.52 – Problem-1: ROC curve of test data set from Bagging model	37
Fig.53 – Problem-1: Confusion matrix of train data set from Boosting model	37
Fig.54 – Problem-1: ROC curve of train data set from Boosting model	38
Fig.55 – Problem-1: Confusion matrix of test data set from Boosting model	38
Fig.56 – Problem-1: ROC curve of test data set from Boosting model	39
Fig.57 – Problem-2: Sample Raw text of Roosevelt_1941 speech	41
Fig.58 – Problem-2: Sample Raw text of Kennedy_1961 speech	42
Fig.59 – Problem-2: Sample Raw text of Nixon_1973 speech	42

List of tables

Table.1 – Problem-1: Data loaded with first five records	5
Table.2 – Problem-1: Data loaded with first five records after dropping Unnamed: 0 column	5
Table.3 – Problem-1: Data information table after dropping Unnamed: 0 column	6
Table.4 – Problem-1: Data types table after dropping Unnamed: 0 column	6
Table.5 – Problem-1: Data description table after dropping Unnamed: 0 column	7
Table.6 – Problem-1: Duplicated records	9
Table.7 – Problem-1: Skewness table	10
Table.8 – Problem-1: Sample data frame after encoding the target variable	19
Table.9 – Problem-1: Sample data frame after encoding the gender variable	20
Table.10 – Problem-1: Sample dataset after scaling the age variable	21
Table.11 – Problem-1: Classification report of train dataset of Logistic Regression model	27
Table.12 – Problem-1: Classification report of test dataset of Logistic Regression model	27
Table.13 – Problem-1: Classification report of train dataset of LDA model	29
Table.14 – Problem-1: Classification report of test dataset of LDA model	30
Table.15 – Problem-1: Classification report of train dataset of Naïve Bayes model	31
Table.16 – Problem-1: Classification report of test dataset of Naïve Bayes model	32
Table.17 – Problem-1: Classification report of train dataset of KNN model	33
Table.18 – Problem-1: Classification report of test dataset of KNN model	34
Table.19 – Problem-1: Classification report of train dataset of Bagging model	35
Table.20 – Problem-1: Classification report of test dataset of Bagging model	36
Table.21 – Problem-1: Classification report of train dataset of Boosting model	37
Table.22 – Problem-1: Classification report of test dataset of Boosting model	38
Table.23 – Problem-1: Performance metrics of Labour class	39
Table.24 – Problem-1: Performance metrics of Conservative class	40

Problem 1

Problem Statement:

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.

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

Exploratory Data Analysis:

➤ Data description:

Reading the data file and loading first five records:

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

Table. 01

Sample Dataset after dropping 'Unnamed: 0':

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

Table. 02

Dataset information:

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

Table. 03

Interpretation:

- There are no null values.
- Total 1525 records and 9 features are in the given dataset.
- There are 8 duplicated records.

Data types:

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

Table. 04

Interpretation:

- There are 7 numeric features and 2 object features.

Dataset description:

	count	mean	std	min	25%	50%	75%	max
age	1525.0	54.18	15.71	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	3.25	0.88	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	3.14	0.93	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	3.33	1.17	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	2.75	1.23	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	6.73	3.30	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	1.54	1.08	0.0	0.0	2.0	2.0	3.0

Table. 05

➤ Insights:

Let us interpret by each feature individually.

Numeric features:

- 1) 'Age':
 - No variation in these features.
 - Minimum voting age is 24 and maximum is 93.
 - Average age is ~54.
- 2) 'economic.cond.national':
 - Current national economic condition is rated from 1 to 5.
 - Average rating is 3.25 which shows national economic condition is good.
- 3) 'economic.cond.household':
 - Current household economic condition also rated on scale of 1 to 5.
 - Average rating is 3.14 which shows household economic condition is also good.
 - On comparing both national and household, national economic condition is slightly better than household.
- 4) 'Blair':
 - Assessment of 'Labour' party leaders is rated from 1 to 5.
 - Average rating is 3.33 which is good.
- 5) 'Hague':
 - Assessment of 'Conservative' party leaders is rated from 1 to 5.
 - Average rating is 2.73 which is not good.
 - On comparing both Labour and Conservative parties, Labour party is having better rating than Conservative in this particular survey.
- 6) 'Europe':

-An 11-point scale that measures respondents' attitudes toward European integration

-High scores represent 'Eurosceptic' sentiment.

-Average is 6.73 which indicates respondents' attitudes towards European integration is neutral.

7) 'political.knowledge':

-This feature is regarding knowledge of parties' positions on European integration, on a scale of 0 to 3.

-Majority of the voters are aware of parties' stand on European integration.

-There is slight variation in this feature.

Object features:

8) 'gender':

-Survey involved ~53% of the females and ~47% of the males.

9) 'vote':

-As per the survey, ~70%, ~30% of the people have chosen 'Labour' and 'Conservative' parties respectively.

- CNBE channel survey is showing 'Labour' party as the clear winner.

➤ Data pre-processing:

Null values detection and treatment:

- There are no null values in all the features and null treatment is not required.

Duplicated records check and analysis:

- There are 8 duplicated records as shown below:

	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

Table. 06

- It can be inferred that duplicated records are significant to keep for analysis.

Anomalies' check:

- No anomalies are observed.

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

➤ Data visualization:

Univariate analysis:

- Let's visualize all the numeric columns using hist plot and check the distribution nature of the features.

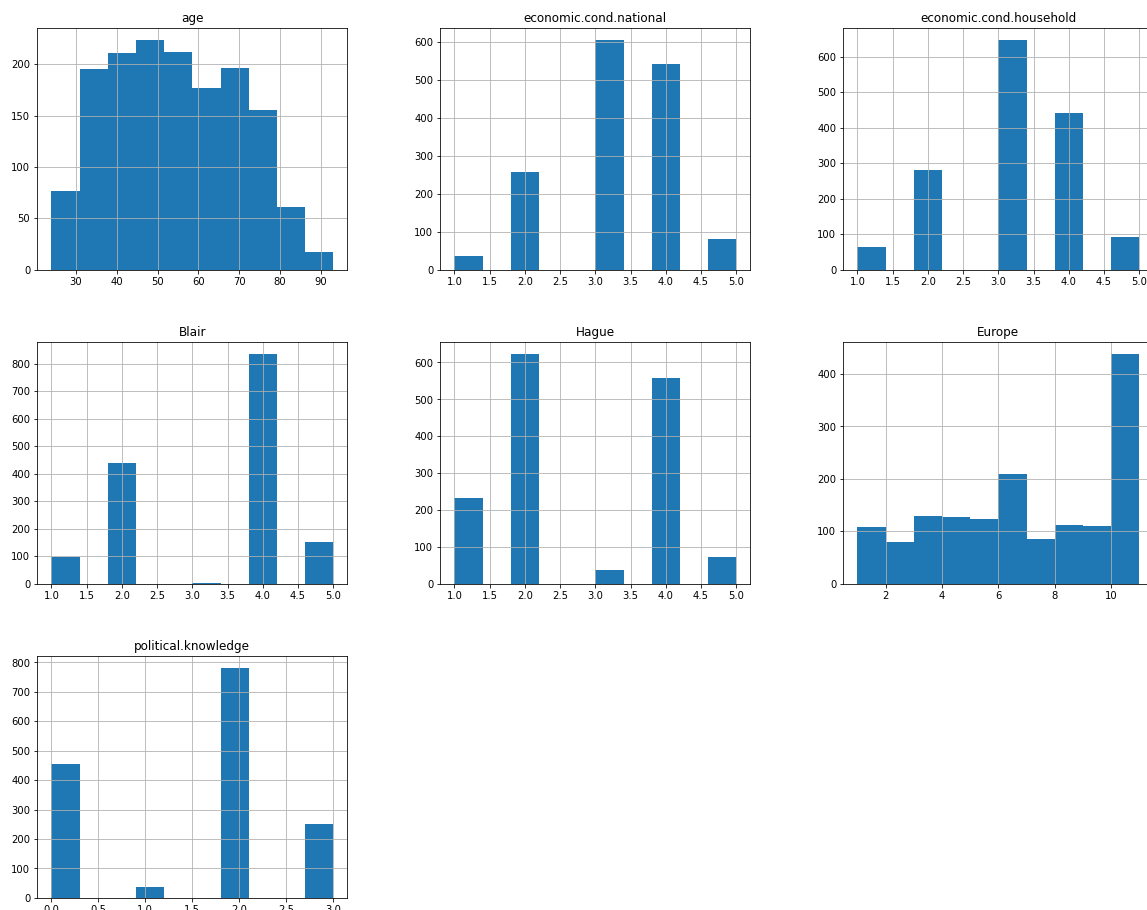


Fig. 01

Checking skewness:

age	0.14
economic.cond.national	-0.24
economic.cond.household	-0.15
Blair	-0.54
Hague	0.15
Europe	-0.14
political.knowledge	-0.43
dtype:	float64

Table. 07

Interpretation:

- All the features are normally distributed.

Outliers check:

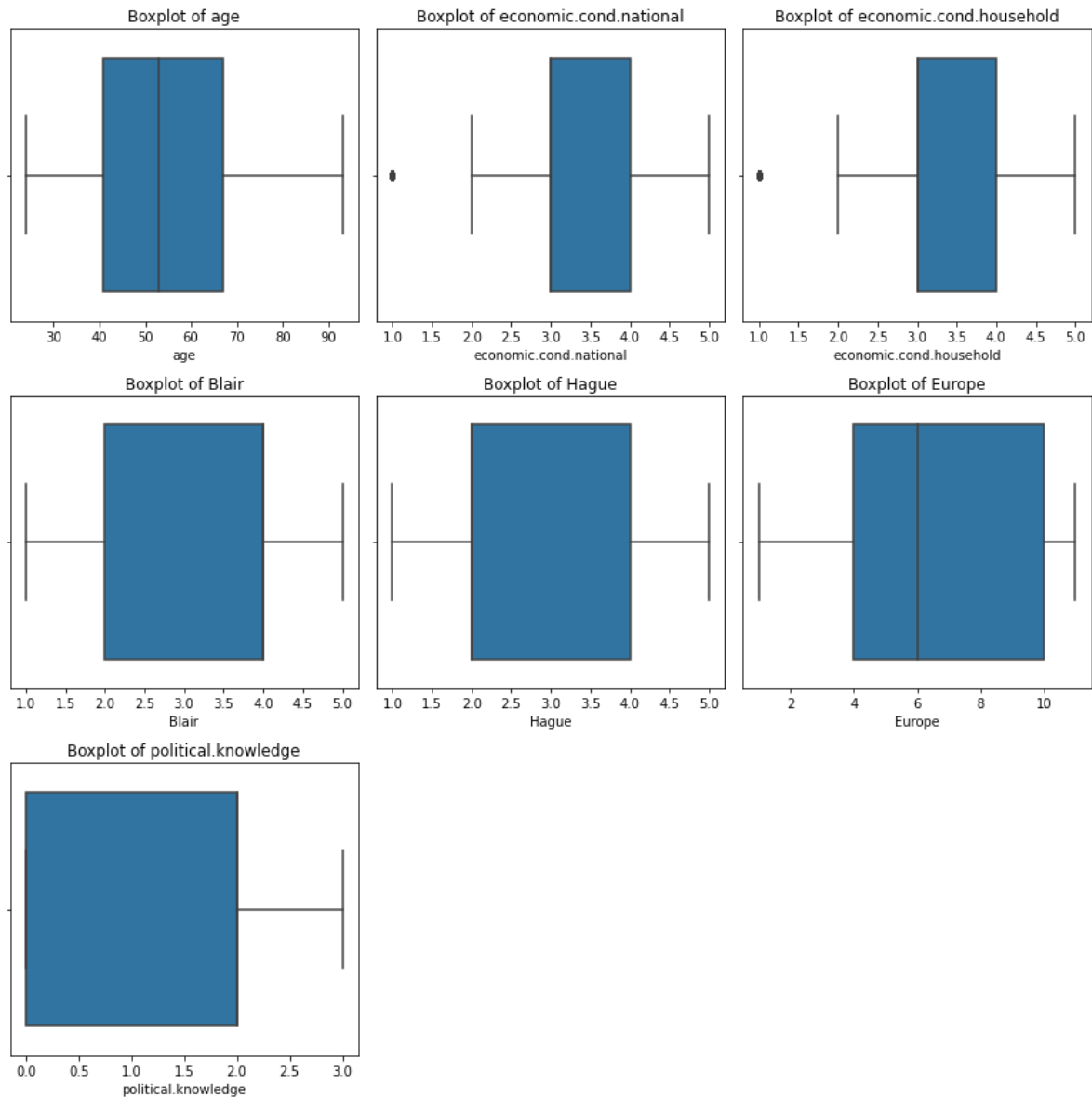


Fig. 02

Interpretation:

- Outliers are present only in two features i.e., 'economic.cond.national', 'economic.co nd.household'.
- But these are discrete numeric features having significant in keeping every value.
- So outlier treatment is not required for this dataset and outlier proportion also cannot be discussed for this dataset.

Let us visualize the categorical variables and its classes: (by using count plot)

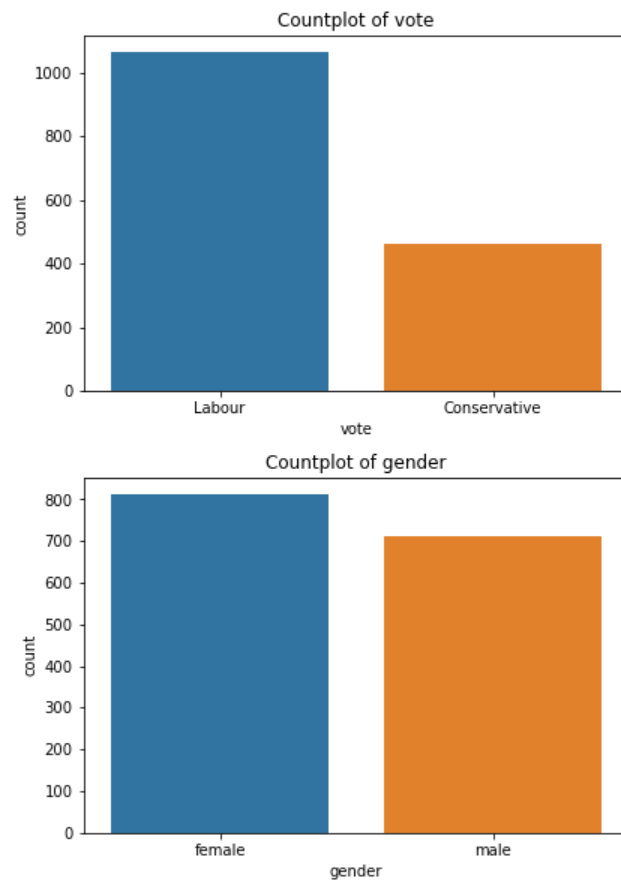


Fig. 03

Interpretation:

- Survey involved ~53% of the females and ~47% of the males.
- As per the survey, ~70%, ~30% of the people have chosen 'Labour' and 'Conservative' parties respectively.
- CNBE channel survey is showing 'Labour' party as the clear winner.

Count plot of 'Blair':

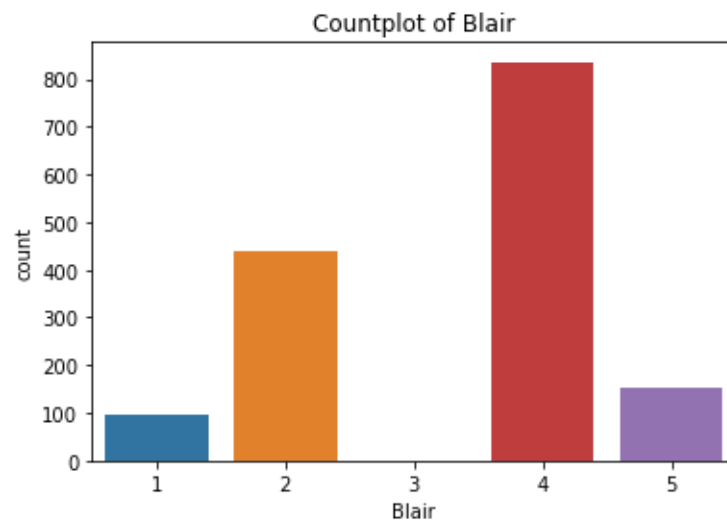


Fig. 04

Interpretatio:

- 4 and 5 rating are in dominating count for 'Labour' party leader which is a good sign.

Count plot of 'Hague':

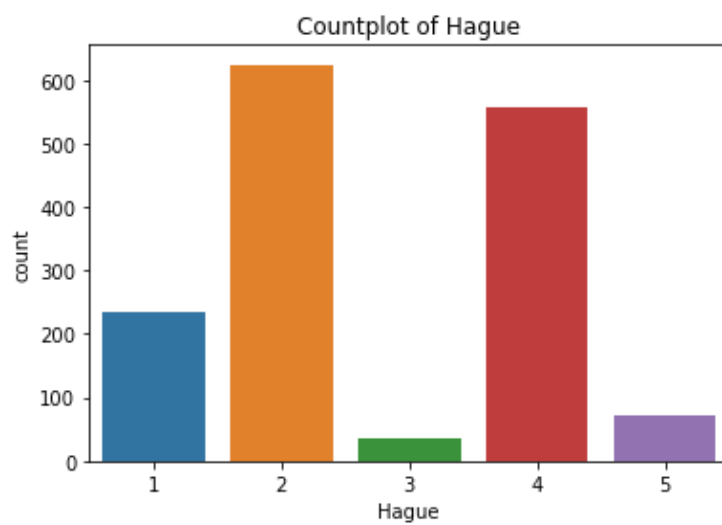


Fig. 05

Interpretation:

- 1 and 2 rating are in dominating count for 'Conservative' party leader which is not a good sign.

Bivariate analysis:

- Let's plot the pair plot and heatmap to check correlation b/w the data features

Pair plot:

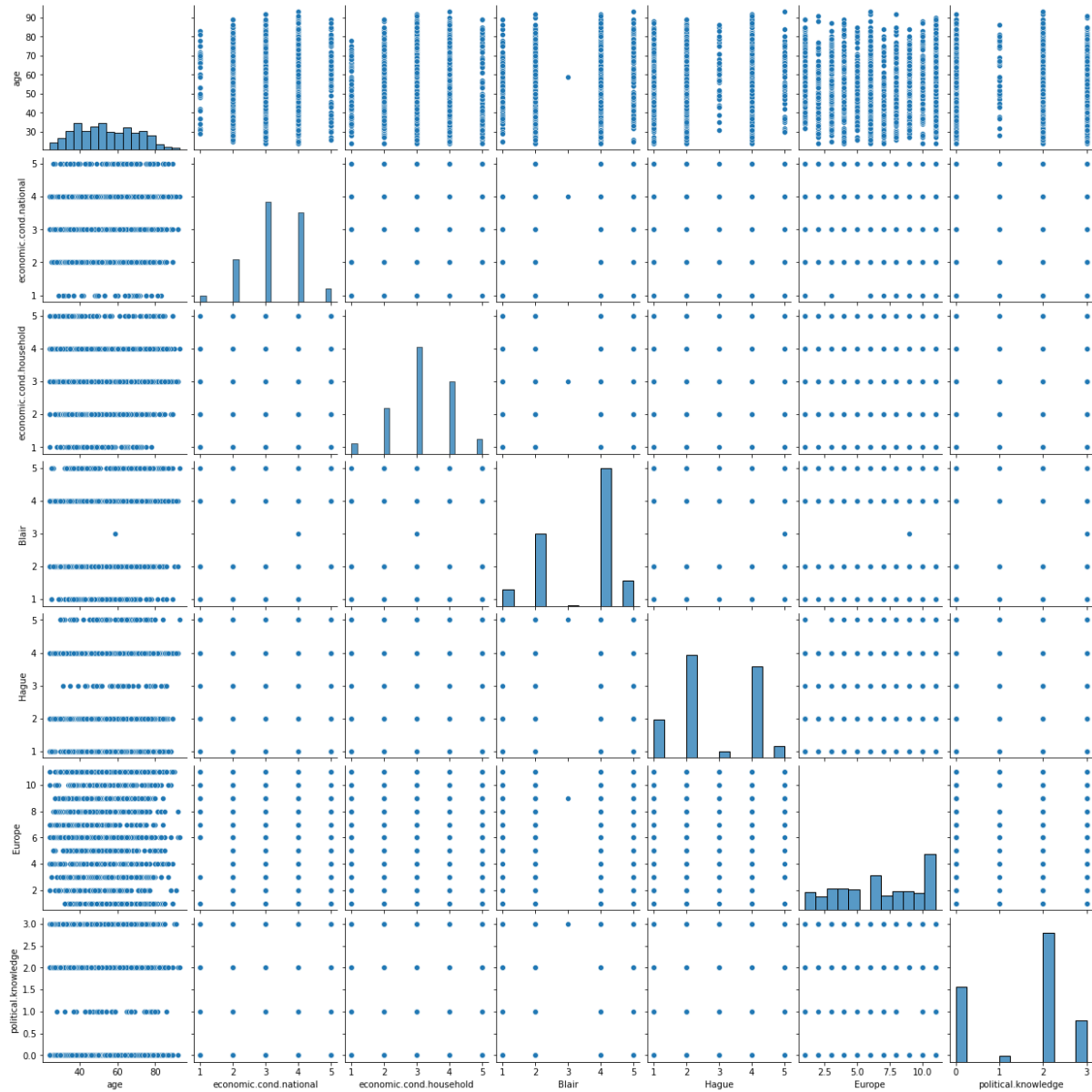


Fig. 06

Heatmap:

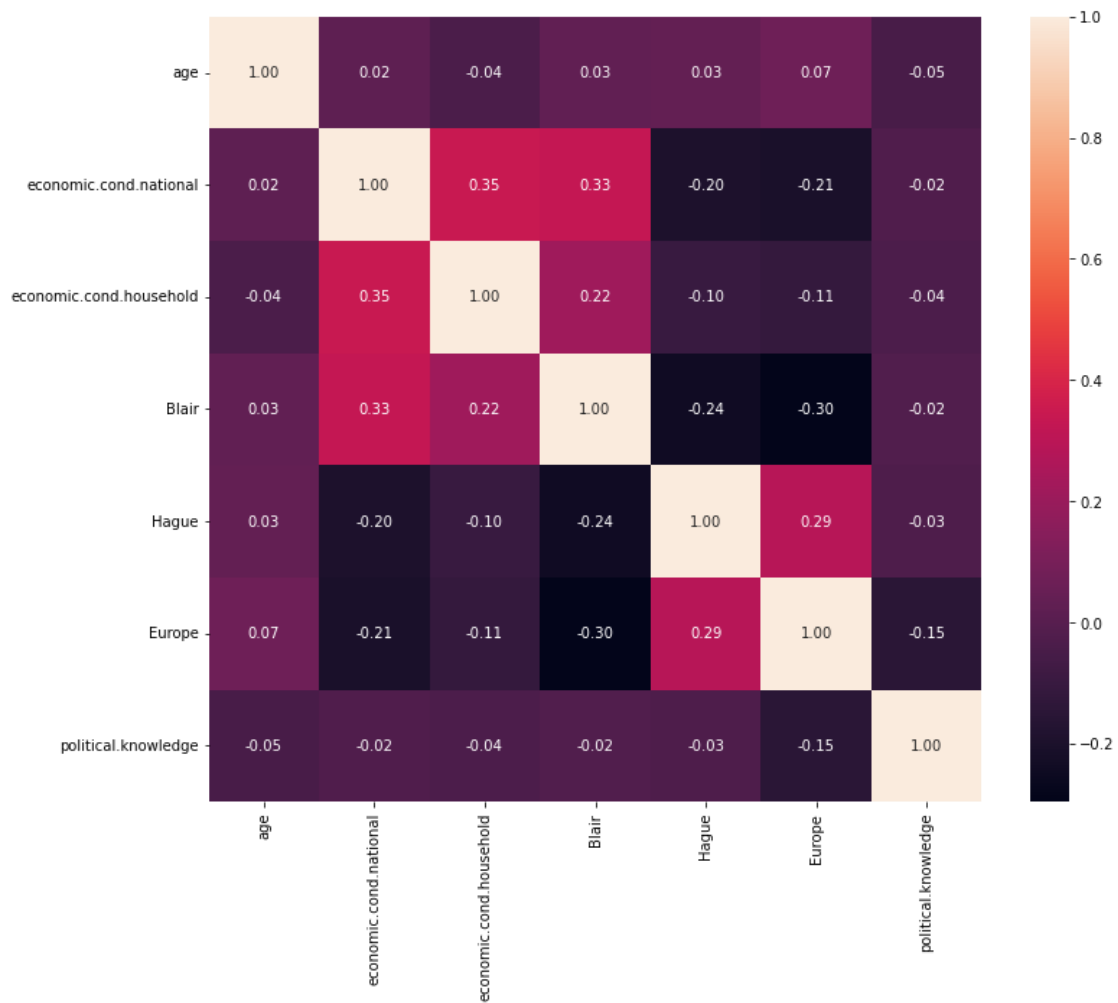


Fig. 07

Interpretation (From both pairplot and heatmap):

- There is no correlation among all the features.
- Except age feature, all the numeric features are discrete in nature. That's can be the reason for not observing any correlation among the numeric features available.

Count plot of 'vote' for 'gender':

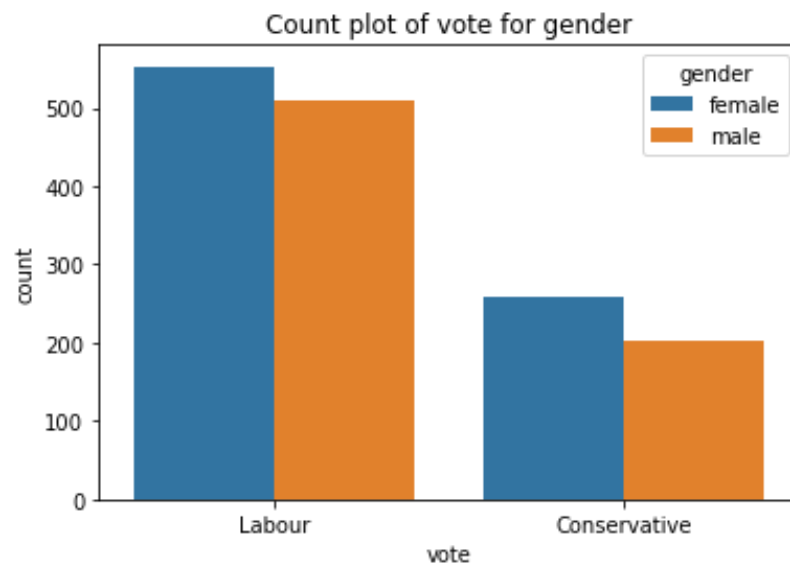


Fig. 08

Interpretation:

- Females have major voter share for the both of the parties.

Count plot of 'vote' for 'economic.cond.national':

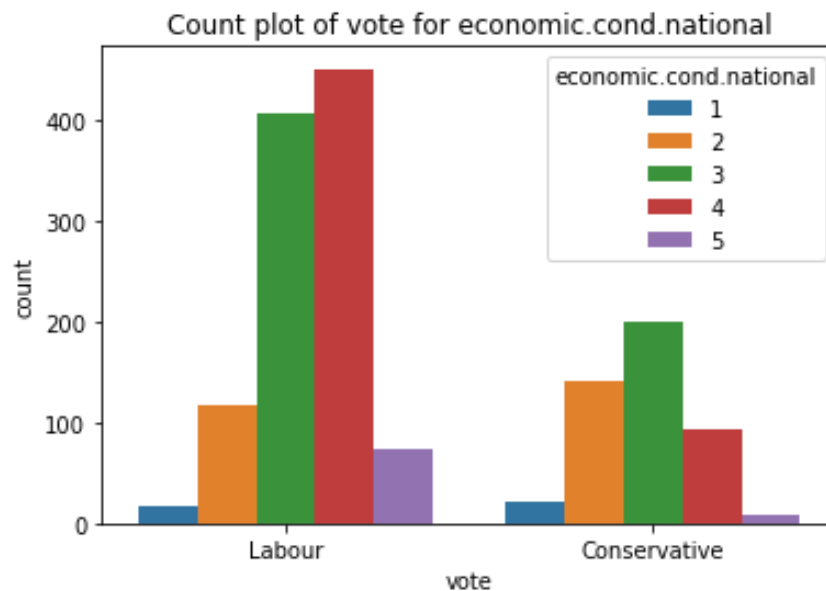


Fig. 09

Interpretation:

- Voters have chosen 'Labour' party keeping current national economic condition in mind.

- This feature could be the reason for big difference b/w overall vote share between both the parties.

Count plot of 'vote' for 'economic.cond.household':

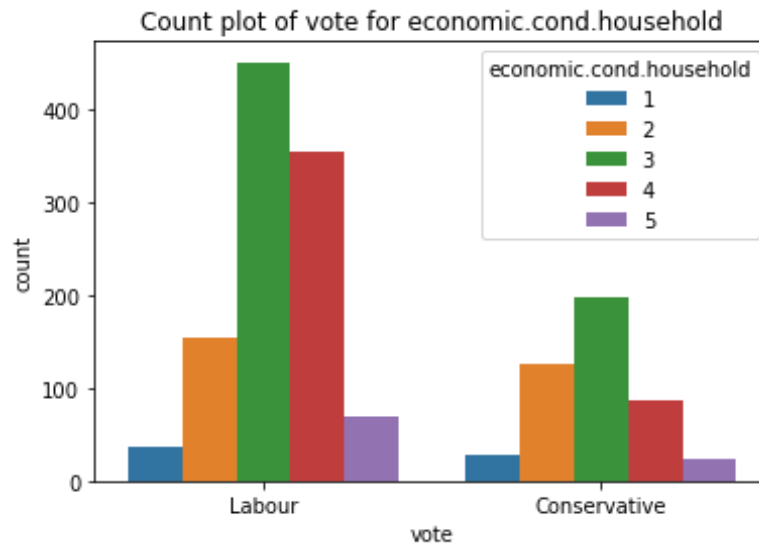


Fig. 10

Interpretation:

- Voters have chosen 'Labour' party keeping current household economic condition in mind.
- This feature also behaving same as 'economic.cond.national'.
- Current economical conditions' assessment of both national and household are playing very key role, the same on which 'Labour' party is good at.

Count plot of 'vote' for 'Europe':

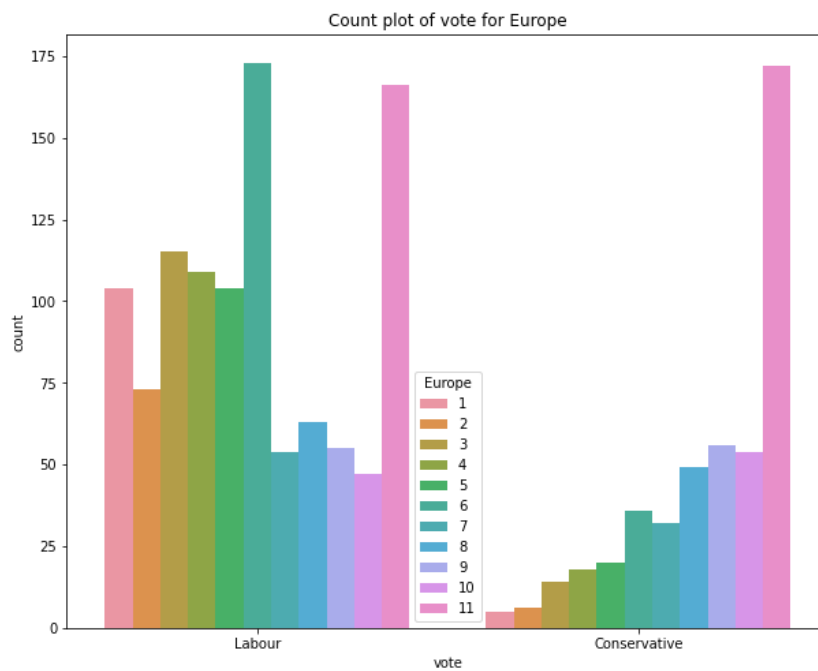


Fig. 11

Interpretation:

- Basically, 'Europe' feature is an 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Euroseptic' sentiment.
- From above plot, we can see difference from 1 to 6 scores b/w both the parties, i.e., low rating range.
- Low scores represent non-Euroseptic sentiment and High scores represent Euroseptic sentiment.
- Clearly, Labour party is favoring towards European integration and its increasing power. This aspect also resulting more vote share for the Labour party.

Count plot of 'vote' for 'Political.Knowledge':

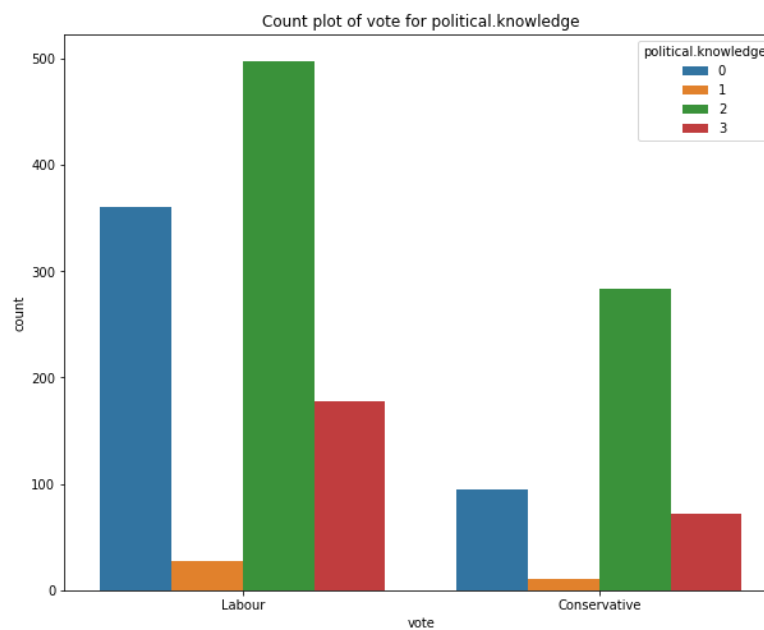


Fig. 12

Interpretation:

- Both of the parties are sharing the same amount of knowledge towards European integration.

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

Data Encoding:

Sample data frame after encoding the target variable 'Vote':

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

Table. 08

- Note: 0- Conservative; 1- Labour

Sample data frame after encoding the categorical variable 'gender':

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

Table. 09

Note: 0- female; 1- male

- Shape of encoded data frame is (1525, 9)

Data Splitting:

- Let us split the data into train and test set in 70:30 ratio.
- Shapes of splitted train and test sets are as shown below:

```
Shape of X_train is (1067, 8)
Shape of y_train is (1067, 1)
Shape of X_test is (458, 8)
Shape of y_test is (458, 1)
```

Fig. 13

- Let us check ratio of data after splitting

```
Split percent of X_train is 69.97
Split percent of y_train is 69.97
Split percent of X_test is 30.03
Split percent of y_test is 30.03
```

Fig. 14

- We can see that data successfully splitted into 70:30 train/test ratio.

Data Scaling:

- Data scaling is not required for this dataset except for KNN model.

Reason:

- The ML algorithms require the scaling if the values of the features are not closer to each other. If the values are close, there are chances for the algorithm to get trained well and faster instead of the data set where the data points or features values have high differences with each other will take more time to understand the data and the accuracy will be lower.
- So, if the data in any conditions has data points far from each other, scaling is a technique to make them closer to each other or in simpler words, we can say that the scaling is used for making data points generalized so that the distance between them will be lower.

- For the given dataset, values are closer to each other, there are no significant outliers. So, scaling is not required for this dataset.
- We will use scaled data for KNN model

Sample Train data set after scaling the 'Age' variable:

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
1493	-1.266219	3	1	4	2	6	2	0
1431	-0.757323	3	4	4	4	3	2	0
235	-1.138995	4	4	4	2	7	2	0
1078	0.642142	4	3	2	1	4	2	1
735	-1.075383	4	4	4	2	2	3	1

Table. 10

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

Logistic Regression:

- Let us fit the train dataset by using 'Logistic Regression' model.

```
LogisticRegression
LogisticRegression(class_weight='balanced', max_iter=10000, solver='sag')
```

Fig. 15

Hyperparameters:

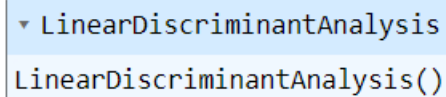
- Class_weight:
It adjusts weights inversely proportional to class frequencies in the input data
- Solver:
For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga' are faster for large ones. 'liblinear' is limited to one-versus-rest schemes. As we have two classes in the target variable and also have enough rows in the dataset, we can use 'sag' or 'saga' solvers. Let us choose 'sag' solver for our model.
- max_iter:
Optimal number of iterations would be 10000 for this solver and for our dataset to avoid overfitting of the model.

Accuracy scores and validity:

- For train dataset, 81.25%
- For test dataset, 82.53%
- Model is valid.

Linear discriminant analysis:

- Let us fit the train dataset by using 'LDA' model.



```
▼ LinearDiscriminantAnalysis
LinearDiscriminantAnalysis()
```

Fig. 16

Hyperparameters:

- Solver: 'svd' is chosen here as a default solver for our dataset.
- Other hyperparameter also can be chosen as default for our dataset.

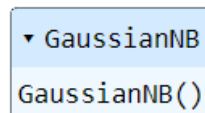
Accuracy scores and validity:

- For train dataset, 82.56%
- For test dataset, 84.49%
- Model is valid.

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

Naïve Bayes:

- Let us fit the train dataset by using 'Naive Bayes' model.



```
▼ GaussianNB
GaussianNB()
```

Fig. 17

Hyperparameters:

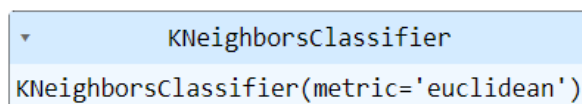
- 'priors': Prior probabilities specification is not required for our dataset and it is chosen as default.
- 'var_smoothing': This artificially adds a user-defined value to the distribution's variance. We are choosing default value as 1e-9.

Accuracy scores and validity:

- For train dataset, 82.19%
- For test dataset, 84.71%
- Model is valid.

K-Nearest Neighbor:

- Let us fit the train dataset by using 'KNN' model.



```
▼ KNeighborsClassifier
KNeighborsClassifier(metric='euclidean')
```

Fig. 18

Hyperparameters:

- 'n_neighbors': 5 nearest neighbours are chosen for our model, choosing more will create overfitting.
- 'weights': Uniform weight is chosen so that all points in each neighbourhood are weighted equally.
- 'metric': Euclidean distance is chosen to find the nearest neighbours as a base metric.
- 'leaf_size': 30 samples is chosen as min samples for each tree. This number should be sufficient for our problem.

Accuracy scores and validity:

- For train dataset, 85.94%
- For test dataset, 83.18%
- Model is valid.

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

Bagging:

- Let us fit the train dataset by using 'Bagging classifier' model.

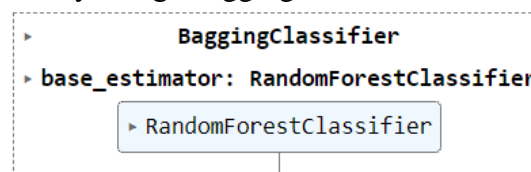


Fig. 19

Hyperparameters:

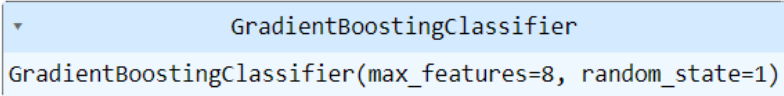
- 'base_estimator': Random Forest is chosen as base estimator.
- 'n_estimators': 100 no. of estimators are chosen for our model which is optimal number for our dataset.
- 'bootstrap': Setting bootstrap as true is to make sure samples are drawn with replacement.
- 'max_features': Let us take maximum no. of features, 8 and will fine tune later during gridsearch.
- 'random_state': It controls the random resampling of the original dataset (sample wise and feature wise).

Accuracy scores and validity:

- For train dataset, 97.18%
- For test dataset, 84.27%
- Model is overfit.

Boosting:

- Let us fit the train dataset by using 'Gradient Boosting classifier' model.



```
GradientBoostingClassifier(max_features=8, random_state=1)
```

Fig. 20

Hyperparameters:

- 'loss': 'log loss' function is chosen for our dataset which is actually good choice for classification with probabilistic output.
- 'learning_rate': Low learning rate gives us the good result, 0.1 to 0.3 is better range. So, 0.1 is chosen as learning rate for our model.
- 'n_estimators': Number of boosting stages to be performed each time is chosen as 100 which is ideal value.
- 'max_features': This is regarding number of features to consider when looking for the best split. Let us take maximum no. of features, 8 and will fine tune later during gridsearch.
- 'random_state': This is to controls the random seed given to each Tree estimator at each boosting iteration.

Accuracy scores and validity:

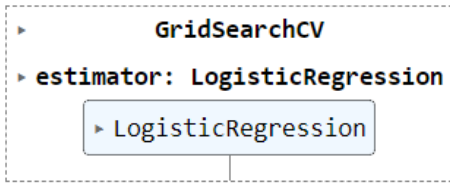
- For train dataset, 88.75%
- For test dataset, 83.84%
- Model is valid.

Model tuning by using GridSearchCV:

Note: Logical reasoning for selection of different parameters has been explained in above sections. Now let us select the best parameters by tuning.

Logistic Regression:

- Let us do the grid search on the train dataset.



```
GridSearchCV
  estimator: LogisticRegression
    LogisticRegression
```

Fig. 21

- Best parameters and estimator after grid search are as below:

```
{'class_weight': 'dict', 'max_iter': 10000, 'solver': 'sag', 'tol': 1e-05}
LogisticRegression(class_weight='dict', max_iter=10000, solver='sag', tol=1e-05)
```

Fig. 22

Accuracy scores and validity after grid search:

- For train dataset, 83.03%
- For test dataset, 84.93%
- Model is valid.
- Accuracy scores are increased ~2% for both the train and test datasets after tuning the model using grid search.

Linear discriminant analysis (LDA):

- Let us do the grid search on the train dataset.

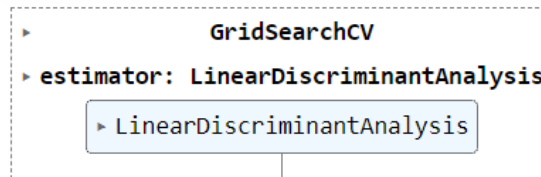


Fig. 23

- Best parameters and estimator after grid search are as below:

```
{'shrinkage': 'auto', 'solver': 'lsqr', 'tol': 0.0001}
LinearDiscriminantAnalysis(shrinkage='auto', solver='lsqr')
```

Fig. 24

Accuracy scores and validity after grid search:

- For train dataset, 82.84%
- For test dataset, 85.15%
- Model is valid.
- There is no significant improvement in accuracy scores after tuning by grid search.

Naïve Bayes:

- Let us do the grid search on the train dataset.

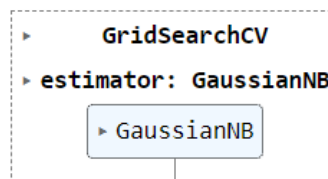


Fig. 25

- Best parameters and estimator after grid search are as below:

```
{'var_smoothing': 0.0008111308307896872}
GaussianNB(var_smoothing=0.0008111308307896872)
```

Fig. 26

Accuracy scores and validity after grid search:

- For train dataset, 82.09%
- For test dataset, 85.15%
- Model is valid.
- There is no significant improvement in accuracy scores after tuning by grid search.

KNN:

- Let us do the grid search on the train dataset.

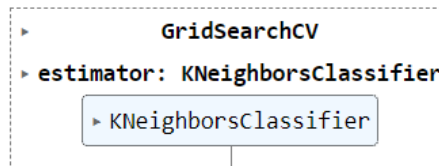


Fig. 27

- Best parameters and estimator after grid search are as below:

```
{'leaf_size': 2, 'metric': 'manhattan', 'n_neighbors': 26, 'weights': 'uniform'}
KNeighborsClassifier(leaf_size=2, metric='manhattan', n_neighbors=26)
```

Fig. 28

Accuracy scores and validity after grid search:

- For train dataset, 84.16%
- For test dataset, 85.15%
- Model is valid.
- There is no significant improvement in accuracy scores after tuning by grid search.

Bagging:

- Let us do the grid search on the train dataset.

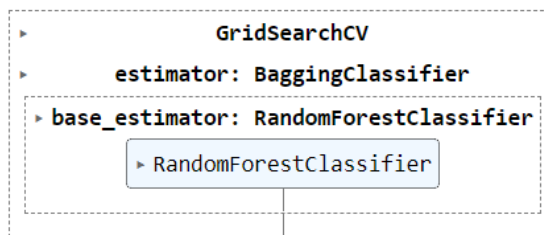


Fig. 29

- Best parameters and estimator after grid search are as below:

```
{'max_features': 8, 'max_samples': 0.2, 'n_estimators': 200}
BaggingClassifier(base_estimator=RandomForestClassifier(), max_features=8,
                  max_samples=0.2, n_estimators=200, random_state=1)
```

Fig. 30

Accuracy scores and validity after grid search:

- For train dataset, 87.06%

- For test dataset, 84.93%
- Model is valid.
- Overfitting issue has been rectified by tuning the model.

Boosting:

- Let us do the grid search on the train dataset.

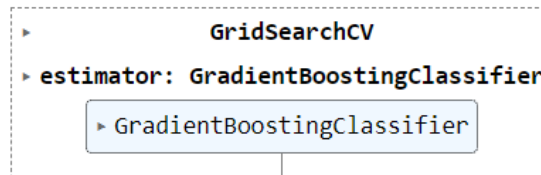


Fig. 31

- Best parameters and estimator after grid search are as below:

```
{'learning_rate': 0.05, 'max_depth': 3, 'max_features': 4, 'n_estimators': 100}
GradientBoostingClassifier(learning_rate=0.05, max_features=4, random_state=1)
```

Fig. 32

Accuracy scores and validity after grid search:

- For train dataset, 86.97%
- For test dataset, 85.80%
- Model is valid.
- There is no significant improvement in accuracy scores after tuning by grid search.

1.7.Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized.

Logistic Regression model:

- Train and test datasets are predicted using tuned Logistic Regression model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[211, 112],
       [ 69, 675]], dtype=int64)
```

Fig. 33

- Classification report table:

	precision	recall	f1-score	support
0	0.753571	0.653251	0.699834	323.000000
1	0.857687	0.907258	0.881777	744.000000
accuracy	0.830366	0.830366	0.830366	0.830366
macro avg	0.805629	0.780254	0.790805	1067.000000
weighted avg	0.826170	0.830366	0.826699	1067.000000

Table. 11

- Accuracy score is 83.03%
- ROC_AUC score is 0.877
- ROC curve:

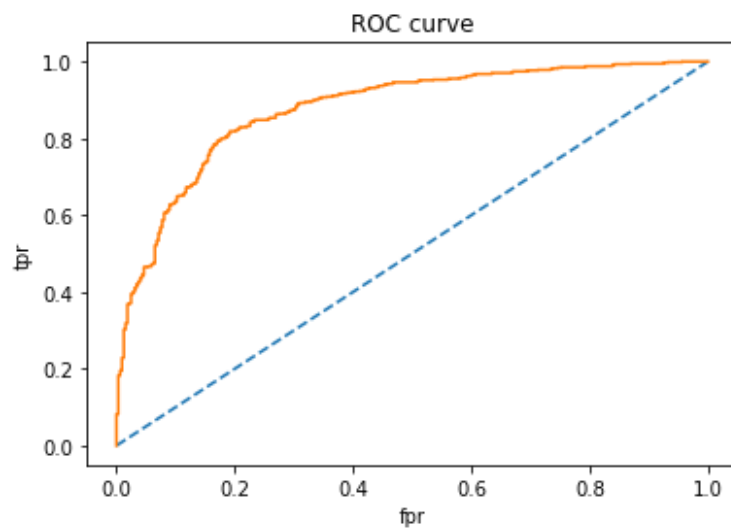


Fig. 34

Test dataset:

- Confusion matrix:

```
array([[ 94,  45],
       [ 24, 295]], dtype=int64)
```

Fig. 35

- Classification report:

	precision	recall	f1-score	support
0	0.796610	0.676259	0.731518	139.000000
1	0.867647	0.924765	0.895296	319.000000
accuracy	0.849345	0.849345	0.849345	0.849345
macro avg	0.832129	0.800512	0.813407	458.000000
weighted avg	0.846088	0.849345	0.845590	458.000000

Table. 12

- Accuracy score is 84.93%
- ROC_AUC score is 0.915
- ROC curve:

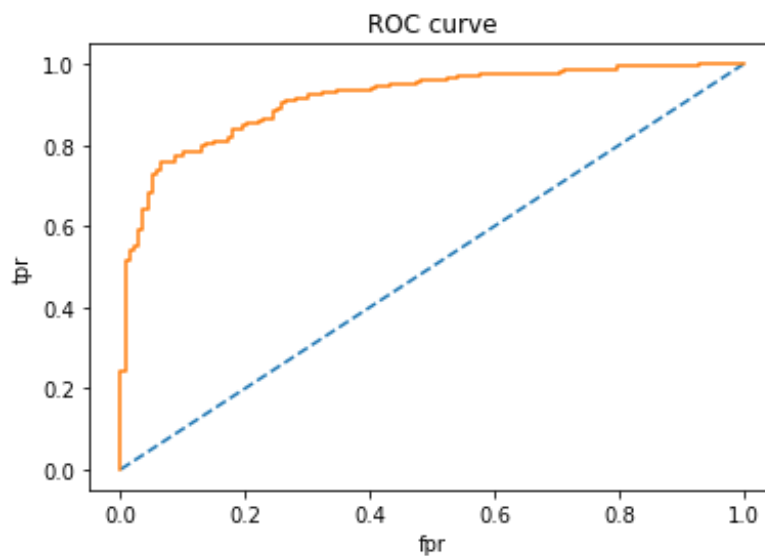


Fig. 36

Linear discriminant analysis:

- Train and test datasets are predicted using tuned LDA model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
[[218 105]
 [ 78 666]]
```

Fig. 37

- Classification report:

	precision	recall	f1-score	support
0	0.736486	0.674923	0.704362	323.000000
1	0.863813	0.895161	0.879208	744.000000
accuracy	0.828491	0.828491	0.828491	0.828491
macro avg	0.800150	0.785042	0.791785	1067.000000
weighted avg	0.825269	0.828491	0.826279	1067.000000

Table. 13

- Accuracy score is 82.84%
- ROC_AUC score is 0.877
- ROC curve:

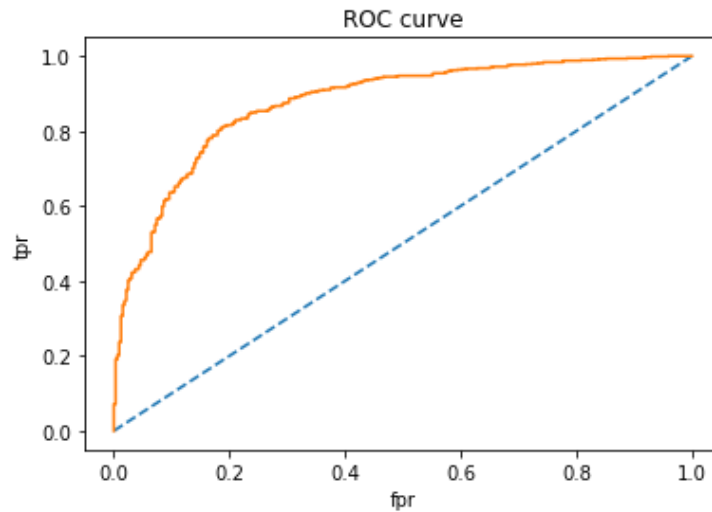


Fig. 38

Test dataset:

- Confusion matrix:

```
array([[100, 39],
       [ 29, 290]], dtype=int64)
```

Fig. 39

- Classification report:

	precision	recall	f1-score	support
0	0.775194	0.719424	0.746269	139.000000
1	0.881459	0.909091	0.895062	319.000000
accuracy	0.851528	0.851528	0.851528	0.851528
macro avg	0.828326	0.814258	0.820665	458.000000
weighted avg	0.849208	0.851528	0.849904	458.000000

Table. 14

- Accuracy score is 85.15%
- ROC_AUC score is 0.915
- ROC curve:

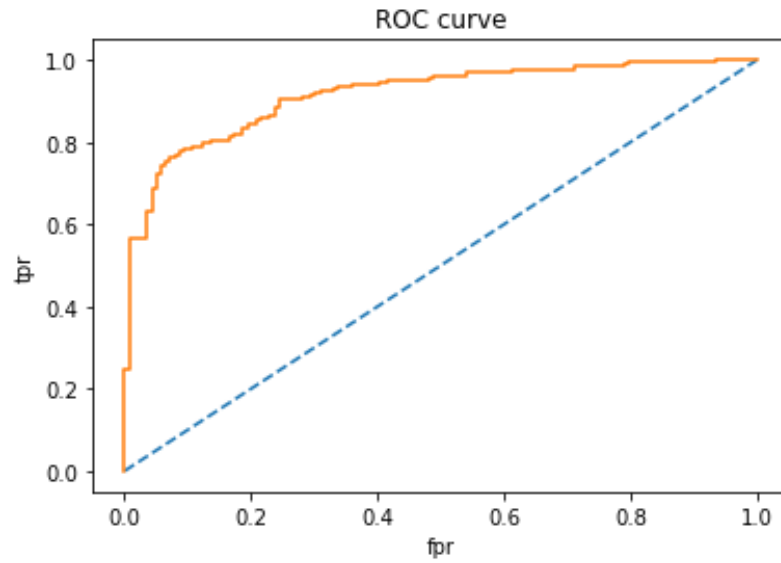


Fig. 40

Naïve Bayes:

- Train and test datasets are predicted using tuned Naïve Bayes model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
[[ 220 103]
 [ 88 656]]
```

Fig. 41

- Classification report:

	precision	recall	f1-score	support
0	0.714286	0.681115	0.697306	323.000000
1	0.864295	0.881720	0.872921	744.000000
accuracy	0.820993	0.820993	0.820993	0.820993
macro avg	0.789290	0.781417	0.785113	1067.000000
weighted avg	0.818885	0.820993	0.819759	1067.000000

Table. 15

- Accuracy score is 82.09%
- ROC_AUC score is 0.874
- ROC curve:

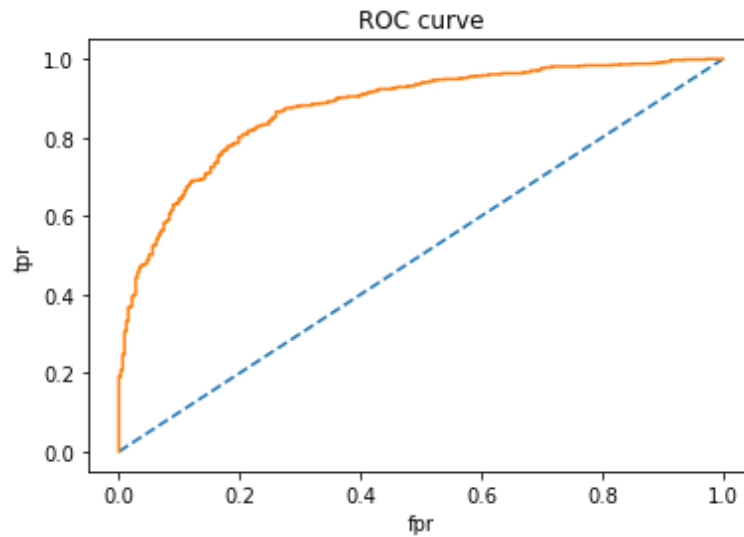


Fig. 42

Test dataset:

- Confusion matrix:

$$\begin{bmatrix} 101 & 38 \\ 30 & 289 \end{bmatrix}$$

Fig. 43

- Classification report:

	precision	recall	f1-score	support
0	0.770992	0.726619	0.748148	139.000000
1	0.883792	0.905956	0.894737	319.000000
accuracy	0.851528	0.851528	0.851528	0.851528
macro avg	0.827392	0.816287	0.821442	458.000000
weighted avg	0.849558	0.851528	0.850248	458.000000

Table. 16

- Accuracy score is 85.15%
- ROC_AUC score is 0.910
- ROC curve:

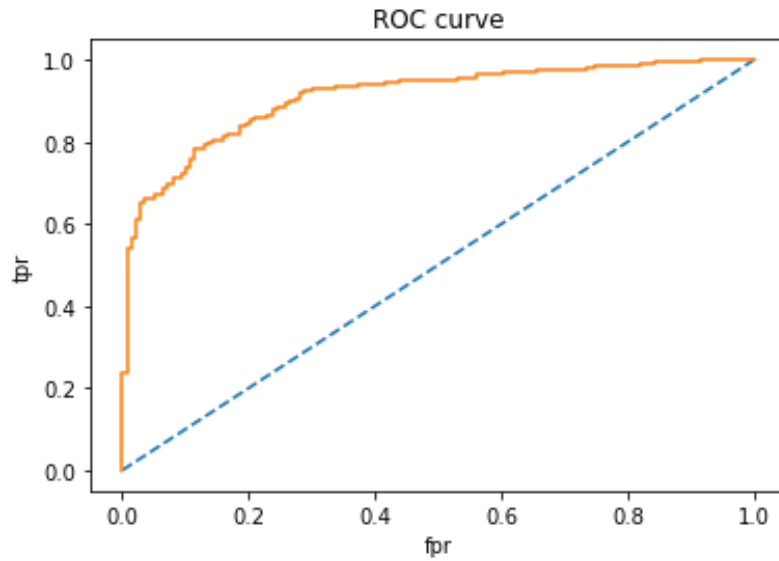


Fig. 44

K- Nearest Neighbors model:

- Train and test datasets are predicted using tuned KNN model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
[[ 228  95]
 [ 74 670]]
```

Fig. 45

- Classification report:

	precision	recall	f1-score	support
0	0.754967	0.705882	0.729600	323.000000
1	0.875817	0.900538	0.888005	744.000000
accuracy	0.841612	0.841612	0.841612	0.841612
macro avg	0.815392	0.803210	0.808803	1067.000000
weighted avg	0.839234	0.841612	0.840053	1067.000000

Table. 17

- Accuracy score is 84.16%
- ROC_AUC score is 0.907
- ROC curve:

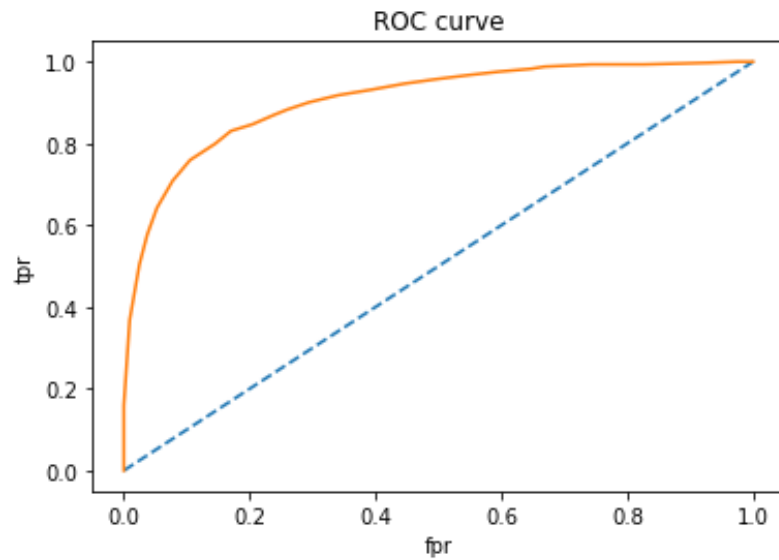


Fig. 46

Test dataset:

- Confusion matrix:

```
[[101  38]
 [ 30 289]]
```

Fig. 47

- Classification report:

	precision	recall	f1-score	support
0	0.770992	0.726619	0.748148	139.000000
1	0.883792	0.905956	0.894737	319.000000
accuracy	0.851528	0.851528	0.851528	0.851528
macro avg	0.827392	0.816287	0.821442	458.000000
weighted avg	0.849558	0.851528	0.850248	458.000000

Table. 18

- Accuracy score is 85.15%
- ROC_AUC score is 0.902
- ROC curve:

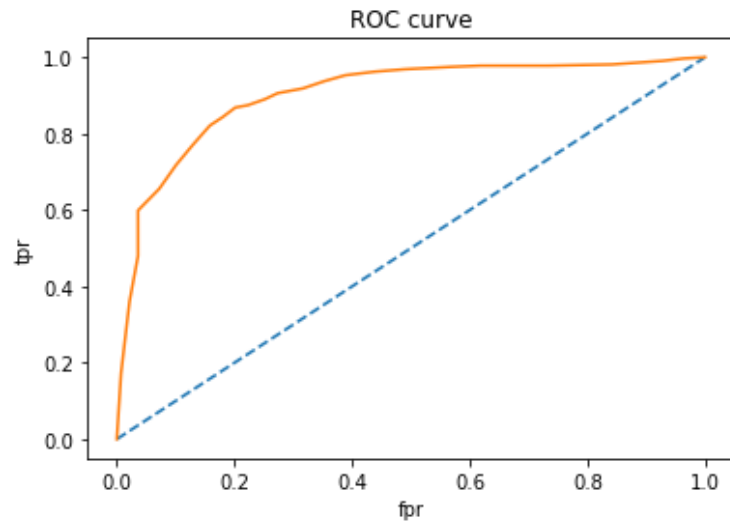


Fig. 48

Bagging:

- Train and test datasets are predicted using tuned Bagging classifier model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- **Confusion matrix:**

```
[[ 224  99]
 [ 39 705]]
```

Fig. 49

- **Classification report:**

	precision	recall	f1-score	support
0	0.851711	0.693498	0.764505	323.000000
1	0.876866	0.947581	0.910853	744.000000
accuracy	0.870665	0.870665	0.870665	0.870665
macro avg	0.864288	0.820540	0.837679	1067.000000
weighted avg	0.869251	0.870665	0.866551	1067.000000

Table. 19

- Accuracy score is 87.06%
- ROC_AUC score is 0.945
- ROC curve:

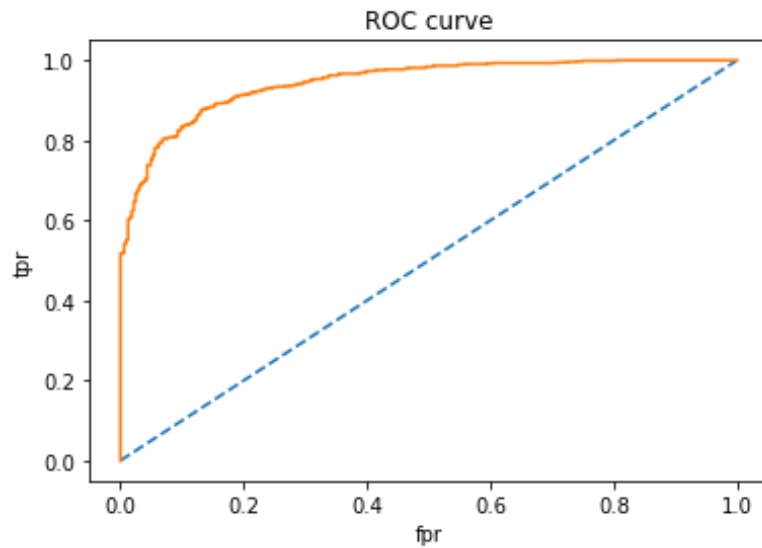


Fig. 50

Test dataset:

- Confusion matrix:

```
[[ 92  47]
 [ 22 297]]
```

Fig. 51

- Classification report:

	precision	recall	f1-score	support
0	0.807018	0.661871	0.727273	139.000000
1	0.863372	0.931034	0.895928	319.000000
accuracy	0.849345	0.849345	0.849345	0.849345
macro avg	0.835195	0.796452	0.811600	458.000000
weighted avg	0.846269	0.849345	0.844742	458.000000

Table. 20

- Accuracy score is 84.93%
- ROC_AUC score is 0.920
- ROC curve:

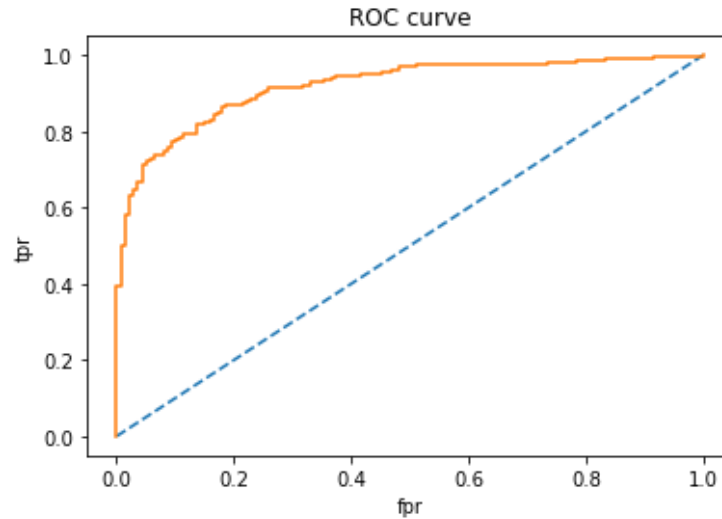


Fig. 52

Boosting:

- Train and test datasets are predicted using tuned Boosting model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
[[232  91]
 [ 48 696]]
```

Fig. 53

- Classification report:

	precision	recall	f1-score	support
0	0.828571	0.718266	0.769486	323.000000
1	0.884371	0.935484	0.909210	744.000000
accuracy	0.869728	0.869728	0.869728	0.869728
macro avg	0.856471	0.826875	0.839348	1067.000000
weighted avg	0.867479	0.869728	0.866913	1067.000000

Table. 21

- Accuracy score is 86.97%
- ROC_AUC score is 0.930
- ROC curve:

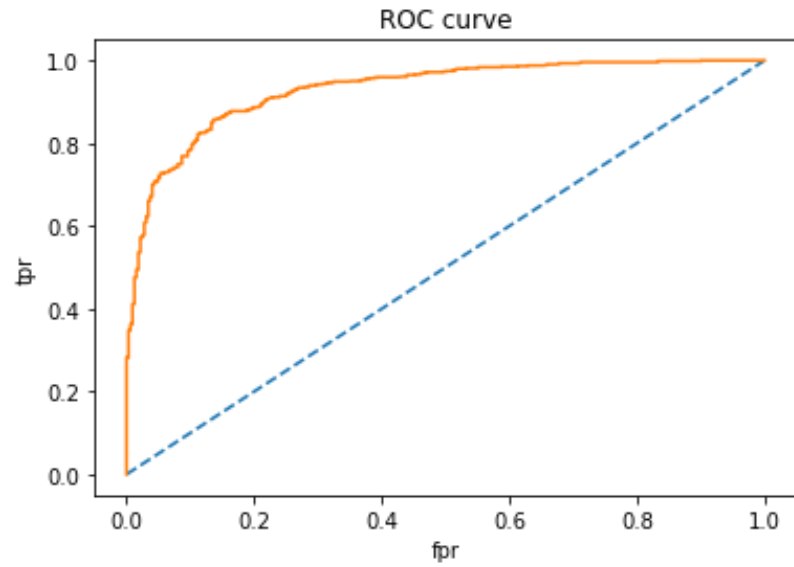


Fig. 54

Test dataset:

- Confusion matrix:

```
[[ 96  43]
 [ 22 297]]
```

Fig. 55

- Classification report:

	precision	recall	f1-score	support
0	0.813559	0.690647	0.747082	139.000000
1	0.873529	0.931034	0.901366	319.000000
accuracy	0.858079	0.858079	0.858079	0.858079
macro avg	0.843544	0.810841	0.824224	458.000000
weighted avg	0.855329	0.858079	0.854542	458.000000

Table. 22

- Accuracy score is 85.80%
- ROC_AUC score is 0.920
- ROC curve:

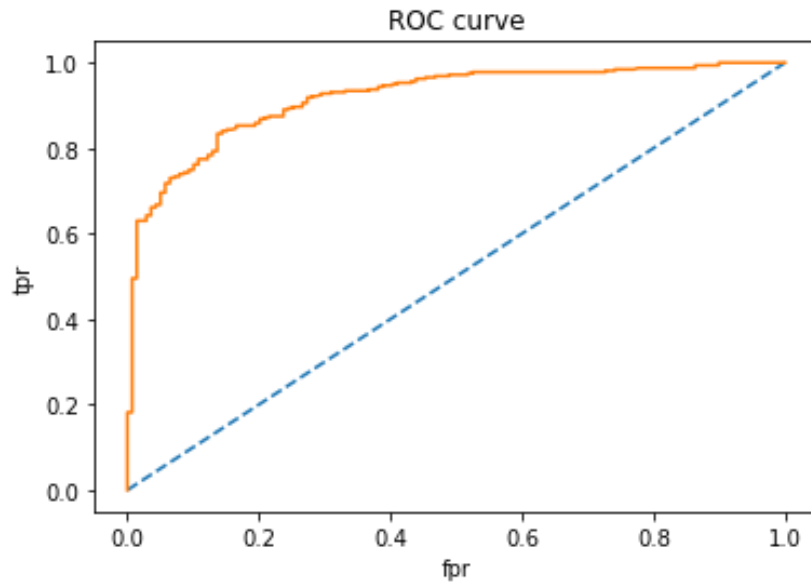


Fig. 56

- Let us tabulate the performance metrics for both the classes i.e., Labour and Conservative:

Note: Accuracy scores and AUC scores will be same, but precision, recall, F1 scores are only differ for both the classes.

Performance metrics table for Labour party:

	Accuracy	AUC	Precision	Recall	F1 Score
Logistic_Train	0.83	0.88	0.86	0.91	0.88
Logistic_Test	0.85	0.91	0.87	0.92	0.90
LDA_Train	0.83	0.88	0.86	0.90	0.88
LDA_Test	0.85	0.92	0.88	0.91	0.90
NB_Train	0.82	0.87	0.86	0.88	0.87
NB_Test	0.85	0.91	0.88	0.91	0.89
KNN_Train	0.84	0.91	0.88	0.90	0.89
KNN_Test	0.85	0.90	0.88	0.91	0.89
Bagging_Train	0.87	0.94	0.88	0.95	0.91
Bagging_Test	0.85	0.92	0.86	0.93	0.90
Boosting_Train	0.87	0.93	0.88	0.94	0.91
Boosting_Test	0.86	0.92	0.87	0.93	0.90

Table. 23

Performance metrics table for Conservative party:

	Accuracy	AUC	Precision	Recall	F1 Score
Logistic_Train	0.83	0.88	0.75	0.65	0.70
Logistic_Test	0.85	0.91	0.80	0.68	0.73
LDA_Train	0.83	0.88	0.74	0.67	0.70
LDA_Test	0.85	0.92	0.78	0.72	0.75
NB_Train	0.82	0.87	0.71	0.68	0.70
NB_Test	0.85	0.91	0.77	0.73	0.75
KNN_Train	0.84	0.91	0.75	0.71	0.73
KNN_Test	0.85	0.90	0.77	0.73	0.75
Bagging_Train	0.87	0.94	0.85	0.69	0.76
Bagging_Test	0.85	0.92	0.81	0.66	0.73
Boosting_Train	0.87	0.93	0.83	0.72	0.77
Boosting_Test	0.86	0.92	0.81	0.69	0.75

Table. 24

Conclusion:

- All the models are giving the better accuracy scores i.e., above ~80%
- Boosting and Bagging models have better AUC scores i.e., >0.90 compared to all other models.
- Bagging and Boosting models have better precision, recall, f1 scores compared to all other models.
- So, we can finalize Bagging classifier and Gradient boosting models as our final models for this survey problem.

Note: Class of interest for this problem is Conservative (~30%). So, conclusions are made based on performance metrics of conservative party.

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

Business insights:

Based on the EDA analysis,

- Economic conditions are playing very key role in the elections.
- People are aware of the both parties stand on economy of the both country and every household.
- Labour party is having better knowledge towards this economy aspect than other party which is why CNBE survey is giving Labour party and its' leader as a clear winner.
- Along with economy, European integration and Eurosceptic sentiments are also key aspects in deciding the winner.

- Even though both the parties have good knowledge about European integration, Labour party is more favoured towards the European integration.
- People are wanting unity among European nations, the same which Labour party also supporting.

Recommendations:

- Survey has included ~1500 of the people, it should be more to get clear results, at least in 10000-20000 range is ideal.
- Survey has been done two main aspects here, Economy and European integration. It should have included some more aspects such as Agricultural, Industrial policies. Separate schemes for poor people and their opinion on them. Different policies parties going to offer for entrepreneurs etc.
- Labour party should continue their economy policies and stand on European integration. This will help them to cash more voting percentage.
- Conservative party should concentration on their Economy policies. Their leader rating also should be improved or else party should change the proposed leaders.

THE END