Business Report

Project – Machine Learning

Created by Amit Jain

Table of Contents

List of Figure

1. Election Data – Machine Learning

Introduction: This report explains the business requirements and provide the detailed solution based on the data provided for each problem statement. given in the assignment. Also, the purpose of this exercise is to execute various supervised learning techniques and ensemble techniques on the given data, combine all predictions and find out the model with best prediction or accuracy. In supervised learning techniques, there are clearly defined X and Y variables. Supervised Learning is used to predict either a continuous response (as in regression) or a categorical response (as in classification). Ensemble Learning techniques are machine learning models for combining predictions from multiple separate models. Both regression and classification can be done using Ensemble Learning. Combining all the individual predictions can be done using either voting or averaging.

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. "

Dataset for Problem 1: Election_Data.xlsx

To understand the problem, for News channel CNBE has given sample of 1525 voters survey randomly collected data in the Election_Data.xlsx, which have pattern of the voters liking, behaviors , and other related patterns.

Assumption:

Assume that the data follows a normal distribution. In reality, the normality assumption may not always hold if the sample size is small.

Step of understanding the data:

Import the data: Imported the data using Python notebooks and analyzed the effects of Education and Occupations over salary field.

This is how the data look like:

	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

Data dictionary and insights:

1. Data has 10 fields/columns:
2. vote: Party choice: Conservative or Labour
3. age: in years
4. economic.cond.national: Assessment of current national economic conditions, 1 to 5.
5. economic.cond.household: Assessment of current household economic conditions, 1 to 5.
6. Blair: Assessment of the Labour leader, 1 to 5.
7. Hague: Assessment of the Conservative leader, 1 to 5.
8. Europe: an 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Eurosceptic' sentiment.
9. political.knowledge: Knowledge of parties' positions on European integration, 0 to 3.
10. gender: female or male.
11. 'Unnamed: 0' this is just a serial number field and can be dropped.
12. Column "vote" and "gender" are in object data type and rest all in Numeric format.

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

We have loaded the data into election_df data frame and checked for the shape of the data: Data have 1525 Rows and 10 Columns.

Check column values:

```
#   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Unnamed: 0               1525 non-null   int64
 1   vote                     1525 non-null   object
 2   age                      1525 non-null   int64
 3   economic.cond.national   1525 non-null   int64
 4   economic.cond.household  1525 non-null   int64
 5   Blair                    1525 non-null   int64
 6   Hague                    1525 non-null   int64
 7   Europe                   1525 non-null   int64
 8   political.knowledge      1525 non-null   int64
 9   gender                   1525 non-null   object
dtypes: int64(8), object(2)
```

From the above results we can see that there is no missing value present in the given dataset.

Data description:

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

Insights:

1. Age : with Minimum of 24 and Max of 93 age, fields looks good and no unexpected data from description
2. Rest all fields are just a Discrete data with class representation of numbers between 1 to 3 or 0 to 3 etc. No unexpected data

From the above table, we can see that we have both categorical and continuous data. For categorical data we have vote and gender and for continuous data we have age, Blair, Hague, Europe, political knowledge, economic.cond.national, economic.cond.household. vote will be target variable. From the summary of the dataset, we can see that mean of the age is the highest, followed by Europe, while the mean for political knowledge is least among all. The variables economic.cond.national and Blair share almost the similar mean whereas economic.cond.household has slightly lower mean than economic.cond.national. Similarly, age has the highest standard deviation whereas economic.cond.national has the lowest standard deviation, followed by economic.cond.household that has slightly higher standard deviation.

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

Univariate analysis:

Univariate Analysis should be done for each data columns, to understand the data pattern within each column. There are multiple graphs analysis available for different data types of columns. We usually go for Histogram ,Boxplot and QQplot for Data , which are Numeric in nature. And Count Plot, Pie Chart for data , which are Categorical in nature. Graph Usage:

Histogram gives beautiful distribution chart for the data.

Boxplot is used for median and outliers of the data and where majority of the data present.

Count Plot shows the Counts of each data segments for that column

Pie chart shows the proportion of the fields in complete data set.

Count Plot for Categorical Fields:

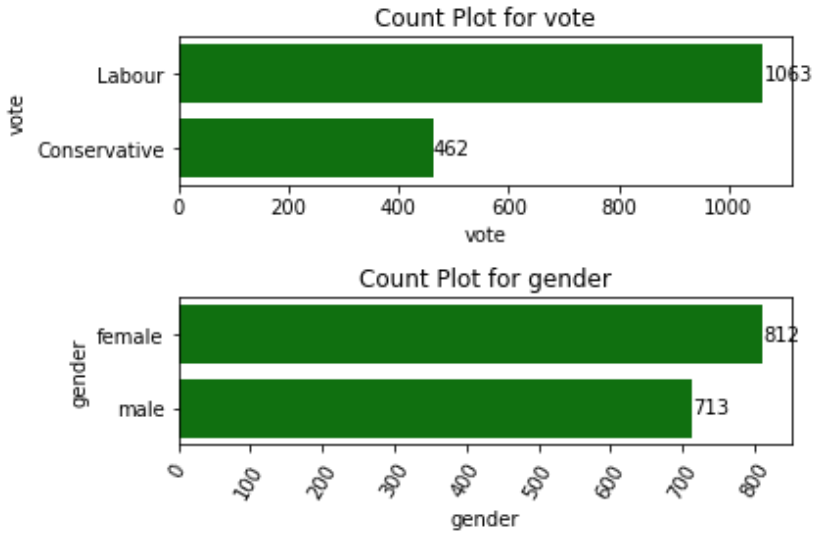


Fig 1 : Count Plot

```
Field name is  VOTE : and Count for this categories are  2
Conservative     462
Labour          1063
Name: vote, dtype: int64
```

```
Field name is  GENDER : and Count for this categories are  2
male      713
female    812
Name: gender, dtype: int64
```

Hist Plot and Box Plot:

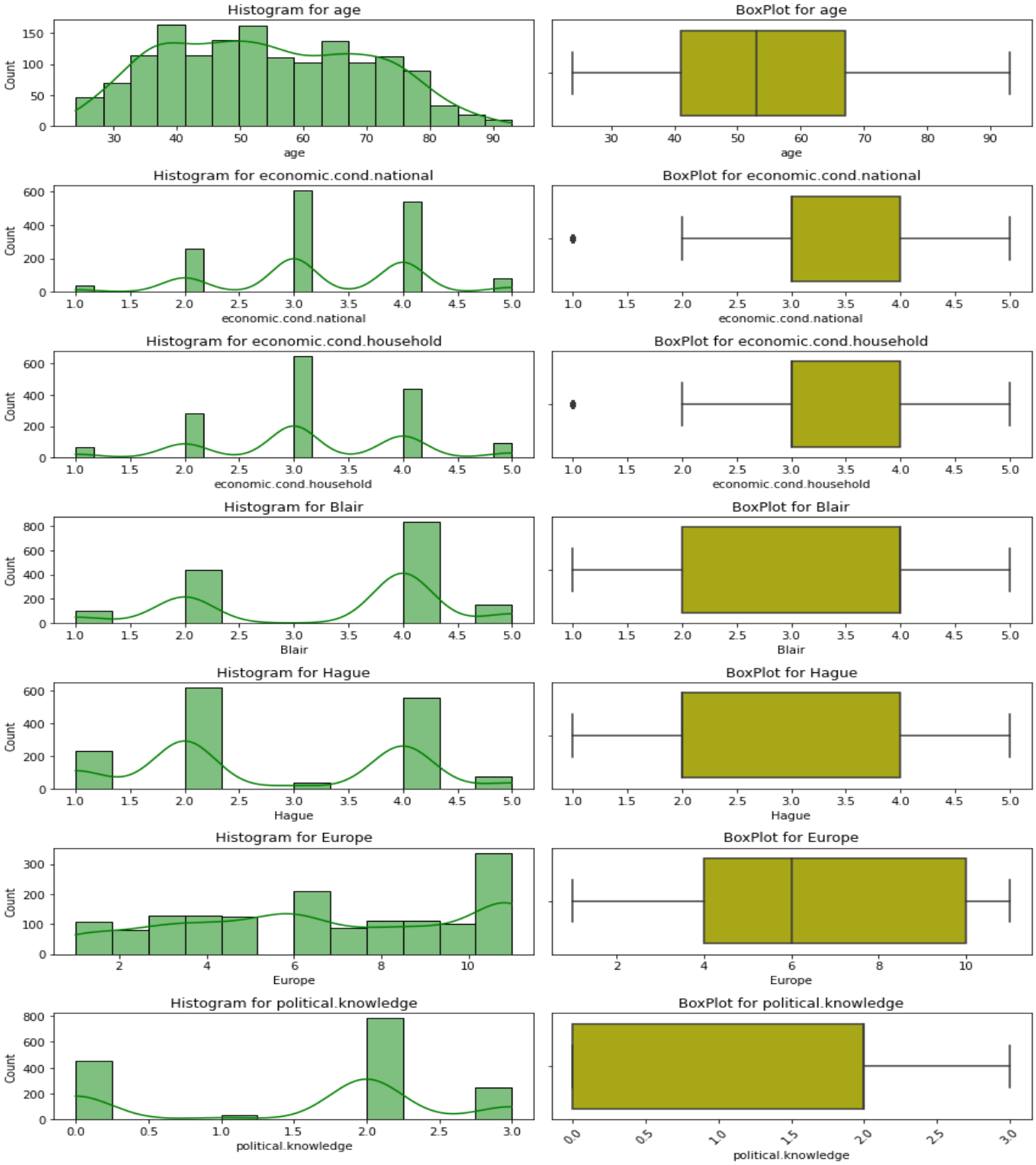


Fig 2: HistPlot and BoxPlot for CNBC News data

Skewness:

```
Hague                       0.152100
age                         0.144621
Europe                     -0.135947
economic.cond.household    -0.149552
economic.cond.national     -0.240453
political.knowledge        -0.426838
Blair                      -0.535419
dtype: float64
```

Observations:

1. data is not 100% normally distributed, but Still I would consider following fields are normally distributed, because its less skewed

```
Hague                       0.152100
age                         0.144621
Europe                     -0.135947
economic.cond.household    -0.149552
economic.cond.national     -0.240453
```

Whereas these fields are slightly Left skewed:

```
political.knowledge        -0.426838
Blair                      -0.535419
```

2. "economic.cond.household" and "economic.cond.national" had outliers for single specific Class of those field.

Count Plot for Independent Numeric field:

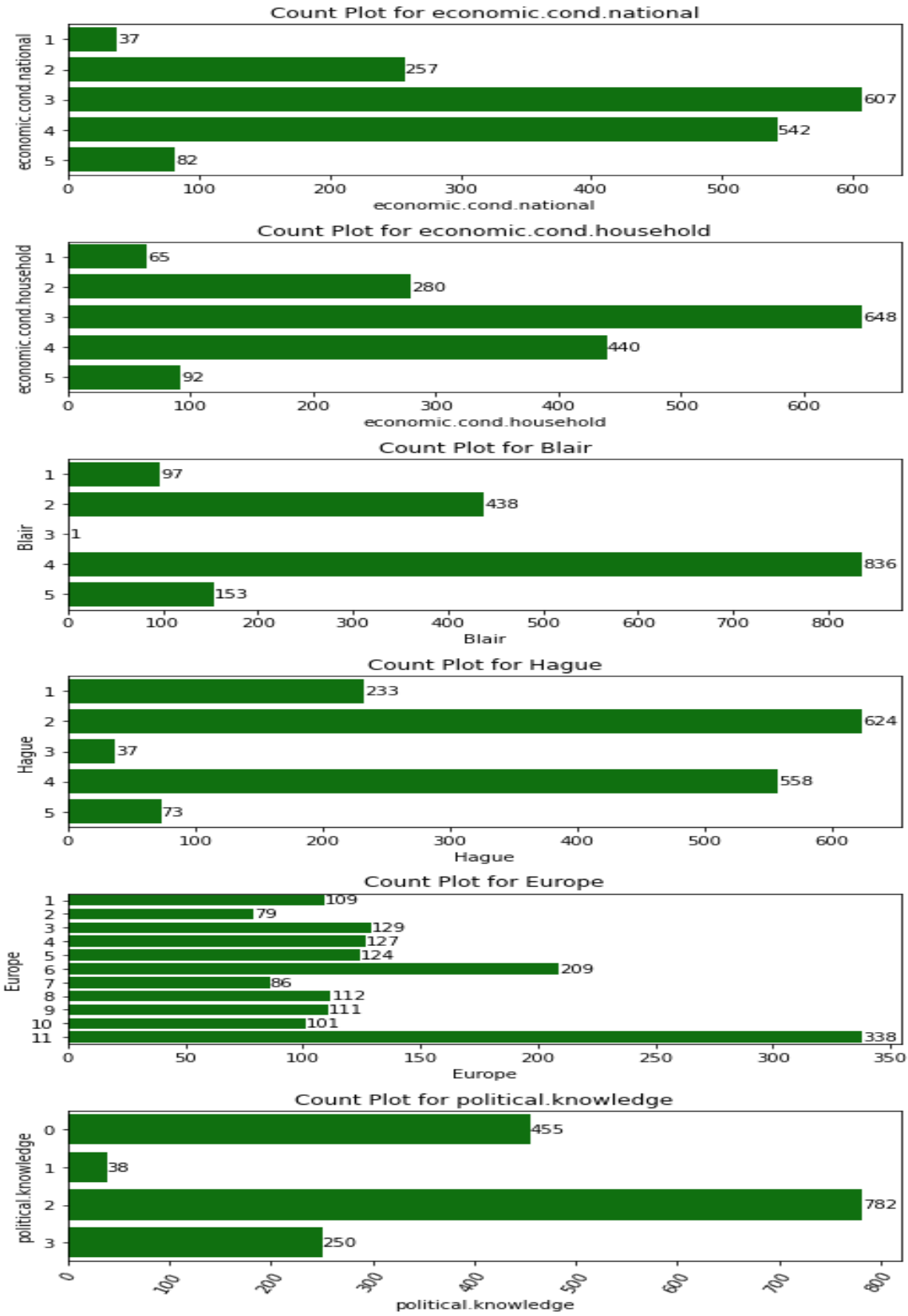


Fig 3: Count Plot for CNBC News data

Insights:

1. VOTE : Count for this categories are 2 and Following are the distribution counts for this columns:

```
Conservative     462
Labour          1063
```

Looks like data is not distributed for this Categories, as there are More Voters who likes "Labour" parties then who like "Conservative"

2. GENDER : and Count for this categories are 2 and Following are the distribution counts for this columns:

```
male      713
female    812
```

Data seems normally distributed, because counts for both categories, male and female is about same

3. ECONOMIC.COND.NATIONAL : and Count for this categories are 5 and Following are the distribution counts for this columns:

```
1      37
5      82
2     257
4     542
3     607
```

Current national Economic condition field is not distributed properly, there are more Samples of Categories 3 and 4 and very less in Category 1 and 5. So this seems right in current environment, when we have "very poor" and "very Rich" people are very less in numbers, whereas many people fall in category of "middle income"

4. ECONOMIC.COND.HOUSEHOLD : and Count for this categories are 5 and Following are the distribution counts for this columns:

```
1      65
5      92
2     280
4     440
3     648
```

Current household economic conditions are also have similar pattern as ECONOMIC.COND.NATIONAL and there are more Samples of Categories 3 and 4 and very less in Category 1 and 5. So this seems right in current environment, when we have "very poor" and "very Rich" people are very less in numbers, whereas many people fall in category of "middle income"

5. Field name is BLAIR : and Count for this categories are 5 and Following are the distribution counts for this columns:

```
3       1
1      97
5     153
```

```
2     438
4     836
```

Data is not distributed among all categories

6. Field name is HAGUE : and Count for this categories are 5 and Following are the distribution counts for this columns:

```
3      37
5      73
1     233
4     558
2     624
```

Data is not distributed among all categories

7. Field name is EUROPE : and Count for this categories are 11 and Following are the distribution counts for this columns:

```
2       79
7       86
10     101
1      109
9      111
8      112
5      124
4      127
3      129
6      209
11     338
```

Data seems normal , High scores represent ‘Eurosceptic’ sentiment.

8. Field name is POLITICAL.KNOWLEDGE : and Count for this categories are 4 and Following are the distribution counts for this columns:

```
1      38
3     250
0     455
2     782
```

Most of the people have Moderate knowledge on Politics, ans which seems Normal in nature

Bivariate Analysis:

Following is the Box Plot for each Numeric Field.

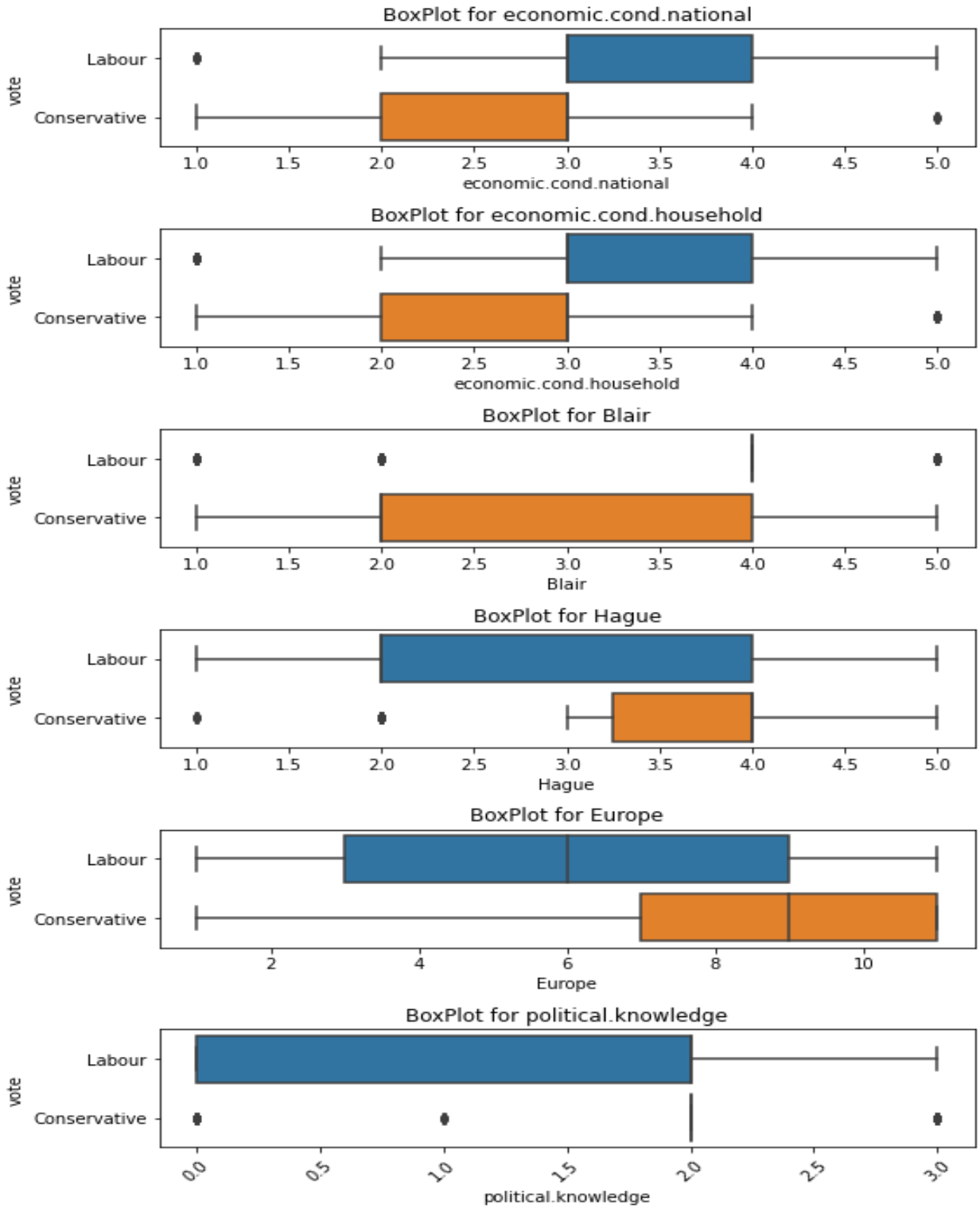


Fig 4: Box Plot with Hue on Dependent var.

Multivariate Analysis:

There are multiple ways for checking multivariate analysis .

Pair Plot: it gives comparison for each field of the data set

Heat Map: we can generate Heat map for the co-relations of the data elements. It gives a beautiful color-coded comparison of the data . and Strong to Light colors shows the Strong to low relationships between Fields. Co-relation chart Metrics can also be built, which shows the co-relation numbers in a Tabular format.

Pair Plot

Fig 5: Pair Plot for CNBC data

There is no clear separation of the target Column Vote data is not easily separable by Straight Line, which give us a clear picture, we need to perform further analysis for predicting the classes

Heat Map:

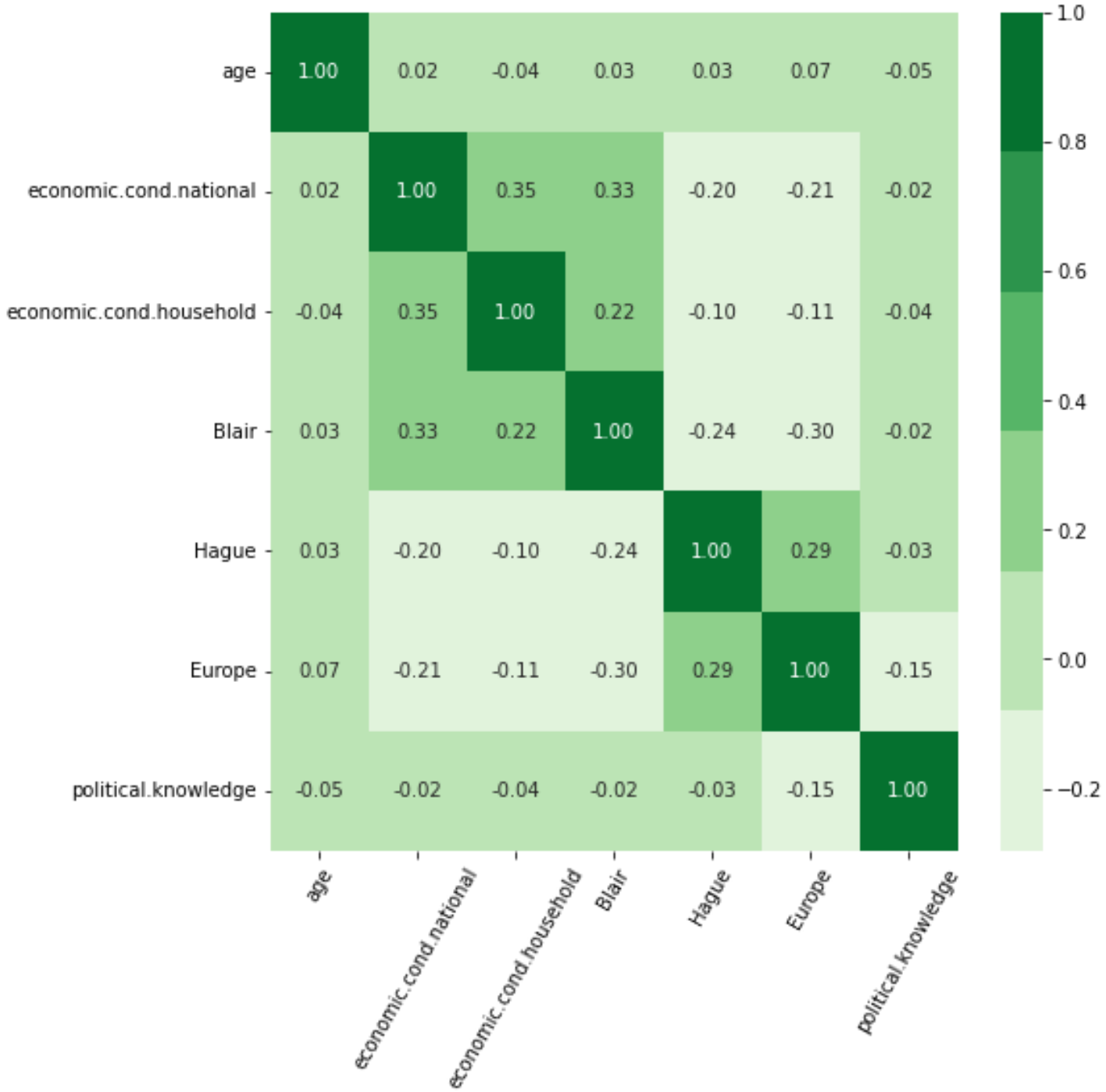


Fig 6: Heat Map for CNBC News data

Co-relation data

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
age	1.000000	0.018567	-0.041587	0.030218	0.034626	0.068880	-0.048490
economic.cond.national	0.018567	1.000000	0.346303	0.326878	-0.199766	-0.209429	-0.023624
economic.cond.household	-0.041587	0.346303	1.000000	0.215273	-0.101956	-0.114885	-0.037810
Blair	0.030218	0.326878	0.215273	1.000000	-0.243210	-0.296162	-0.020917
Hague	0.034626	-0.199766	-0.101956	-0.243210	1.000000	0.287350	-0.030354
Europe	0.068880	-0.209429	-0.114885	-0.296162	0.287350	1.000000	-0.152364
political.knowledge	-0.048490	-0.023624	-0.037810	-0.020917	-0.030354	-0.152364	1.000000

Insights:

1. There is No Strong co-relation between any of the Numeric field or column and looks
2. There is slight Positive relation with "economic.cond.national" and "economic.cond.household" , which is normal, means if Citizen from Sample are wealthy, nation (only Sample) will be wealthy too.
3. there are also slight Positive relation with "economic.cond.national" and "Blair" which means if nation economic condition is good, people tend to choose labour Parties.

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:

We have performed One Hot Encoding for Categorical field as Vote and Gender And this is how the new Sample look like:

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

New Column List is as follows:

```
Index(['Unnamed: 0', 'age', 'economic.cond.national',
       'economic.cond.household', 'Blair', 'Hague', 'Europe',
       'political.knowledge', 'vote_Labour', 'gender_male'],
      dtype='object')
```

Scaling of the data

After performing One Hot encoding this is how the data look like:

```
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   age                      1525 non-null   int64
 1   economic.cond.national   1525 non-null   int64
 2   economic.cond.household  1525 non-null   int64
 3   Blair                    1525 non-null   int64
 4   Hague                    1525 non-null   int64
 5   Europe                   1525 non-null   int64
 6   political.knowledge      1525 non-null   int64
 7   vote_Labour              1525 non-null   uint8
 8   gender_male              1525 non-null   uint8
```

We also analyze the data , with the description of it:

	count	mean	std	min	25%	50%	75%	max
age	1525.0	54.182295	15.711209	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	3.245902	0.880969	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	3.140328	0.929951	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	3.334426	1.174824	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	2.746885	1.230703	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	6.728525	3.297538	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	1.542295	1.083315	0.0	0.0	2.0	2.0	3.0
vote_Labour	1525.0	0.697049	0.459685	0.0	0.0	1.0	1.0	1.0
gender_male	1525.0	0.467541	0.499109	0.0	0.0	0.0	1.0	1.0

Check for the data variance:

```
age                      246.842075
economic.cond.national     0.776107
economic.cond.household    0.864810
Blair                      1.380212
Hague                      1.514631
Europe                    10.873759
political.knowledge        1.173571
dtype: float64
```

Since variance has a big difference, we need to perform Scaling on the data.

We will perform ZScore scaling mechanism on the data , after performing scaling, this is how the data ook like:

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	vote_Labour	gender_male
0	-0.711973	-0.279218	-0.150948	0.566716	-1.419886	-1.434426	0.422643	1	0
1	-1.157661	0.856268	0.924730	0.566716	1.018544	-0.524358	0.422643	1	1
2	-1.221331	0.856268	0.924730	1.418187	-0.607076	-1.131070	0.422643	1	1
3	-1.921698	0.856268	-1.226625	-1.136225	-1.419886	-0.827714	-1.424148	1	0
4	-0.839313	-1.414704	-1.226625	-1.987695	-1.419886	-0.221002	0.422643	1	1

This is the new description of the data after performing scaling:

	count	mean	std	min	25%	50%	75%	max
age	1525.0	1.260922e-16	1.000328	-1.921698	-0.839313	-0.075276	0.816100	2.471512
economic.cond.national	1525.0	2.545141e-16	1.000328	-2.550189	-0.279218	-0.279218	0.856268	1.991754
economic.cond.household	1525.0	-4.551550e-16	1.000328	-2.302303	-0.150948	-0.150948	0.924730	2.000408
Blair	1525.0	4.322954e-16	1.000328	-1.987695	-1.136225	0.566716	0.566716	1.418187
Hague	1525.0	-1.560864e-16	1.000328	-1.419886	-0.607076	-0.607076	1.018544	1.831354
Europe	1525.0	-3.619691e-16	1.000328	-1.737782	-0.827714	-0.221002	0.992422	1.295778
political.knowledge	1525.0	-6.921968e-16	1.000328	-1.424148	-1.424148	0.422643	0.422643	1.346038
vote_Labour	1525.0	6.970492e-01	0.459685	0.000000	0.000000	1.000000	1.000000	1.000000
gender_male	1525.0	4.675410e-01	0.499109	0.000000	0.000000	0.000000	1.000000	1.000000

Data Split: Split the data into train and test (70:30)

We Split the data into independent Variables and Dependent variables , into X and Y set, And then created training and testing set of data with 70-30 percent ratio.

```
Shape for x_train is (1067, 8)
Shape for x_test is  (458, 8)
Shape for y_train  is  (1067, 1)
Shape for y_test is  (458, 1)
```

```
X_train.head()
```

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
1453	0.497751	-0.279218	-0.150948	-1.136225	-0.607076	1.295778	0.422643	0
275	-0.329955	-0.279218	-0.150948	-1.136225	-0.607076	0.385710	-1.424148	0
1130	1.261787	0.856268	0.924730	0.566716	1.018544	0.082354	-1.424148	1
1153	0.179402	-1.414704	-0.150948	0.566716	-0.607076	-0.221002	0.422643	0
1172	-1.921698	0.856268	2.000408	0.566716	1.018544	-0.221002	-1.424148	1

```
y_train.head()
```

	vote_Labour
1453	1
275	0
1130	1
1153	1
1172	0

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

Logistic Regression

Logistic regression is a linear model for classification rather than regression. It is also known as logit regression. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.

Note: Regularization is applied by default, which is common in machine learning but not in statistics. Another advantage of regularization is that it improves numerical stability. No regularization amounts to setting C to a very high value.

Solver: solver is the algorithm to use in the optimization problem. The different types of solvers available are newton-cg, lbfgs, liblinear, sag and saga where lbfgs is the default one. To choose a solver, we might want to consider the following aspects:

1. For small datasets, 'liblinear' is a good choice,
2. whereas 'sag' and 'saga' are faster for large ones;
3. For multiclass problems, only 'newton-cg', 'sag', 'saga' and 'lbfgs' handle multinomial loss;
4. 'liblinear' is limited to one-versus-rest schemes. Since we are dealing with multi class problem

in this dataset where our target variable 'vote' is having two classes, we choose solver to be liblinear Solver.

max_iter: Maximum number of iterations taken for the solvers to converge. Default is 100, but here 10000 is given for better accuracy and more learning by the algorithm.

penalty: Penalized logistic regression imposes a penalty to the logistic model for having too many variables. This results in shrinking the coefficients of the less contributive variables towards zero. This is also known as regularization. The various values given for penalty parameter are none, l2, l1, elasticnet out of which l2 is the default choice

We have used Grid search for choosing the best parameter list and following options were given forchoosing the best param:

#Applying GridSearchCV

grid={'penalty':['l1','l2','none'],

'solver':['lbfgs', 'liblinear'],

'tol':[0.01,0.001]}

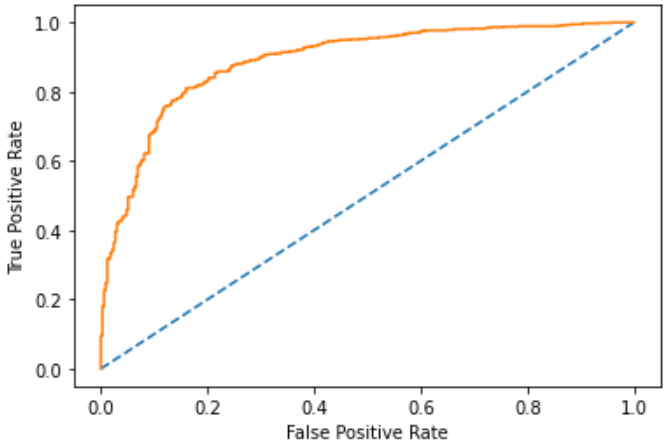
Following are the best parameters were calculated using Grid search calculations:

```
{'penalty': 'l2', 'solver': 'liblinear', 'tol': 0.01}
```

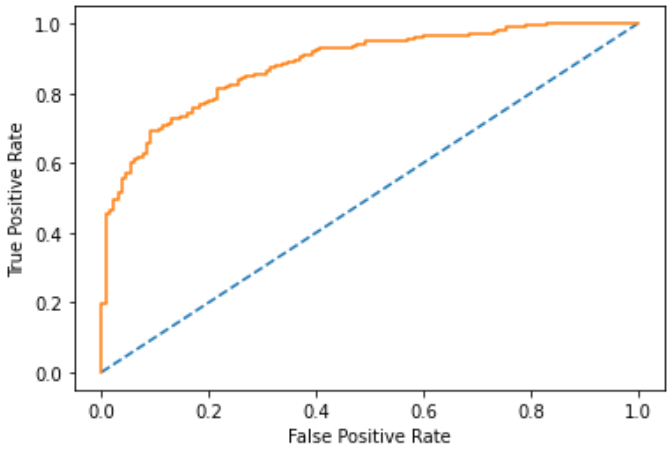
```
LogisticRegression(max_iter=100000, n_jobs=2, solver='liblinear', tol=0.01)
```

Fit and perform Model with above listed data, and following are the results:

```
AUC for training data for Logit Model is : 0.889
```




```
AUC for testing data for Logit Model is : 0.882
```




```
AUC Curve for Test data for LOGIT
```

Fig: 7 AUC Curve for Train and Test data for LOGIT Model

Classification report for Train data is as follows:

	precision	recall	f1-score	support
0	0.77	0.69	0.73	332
1	0.87	0.91	0.89	735
accuracy			0.84	1067
macro avg	0.82	0.80	0.81	1067
weighted avg	0.84	0.84	0.84	1067

Classification report for Test data is as follows:

	precision	recall	f1-score	support
0	0.70	0.65	0.68	130
1	0.87	0.89	0.88	328
accuracy			0.82	458
macro avg	0.78	0.77	0.78	458
weighted avg	0.82	0.82	0.82	458

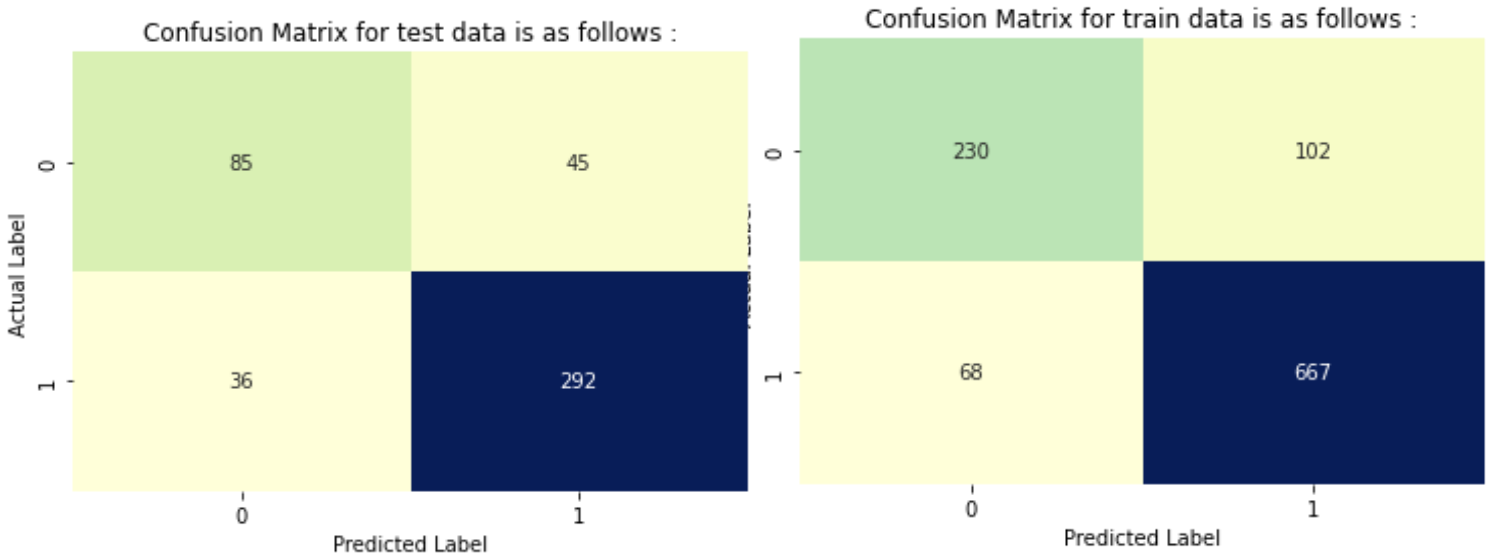


Fig 8: Confusion Matrix for Train and Test data for LOGIT Model

Feature importance:

```
Feature: 0, Score: -0.31987
Feature: 1, Score: 0.29475
Feature: 2, Score: 0.14734
Feature: 3, Score: 0.66726
Feature: 4, Score: -1.01605
Feature: 5, Score: -0.77284
Feature: 6, Score: -0.51500
Feature: 7, Score: 0.30371
```

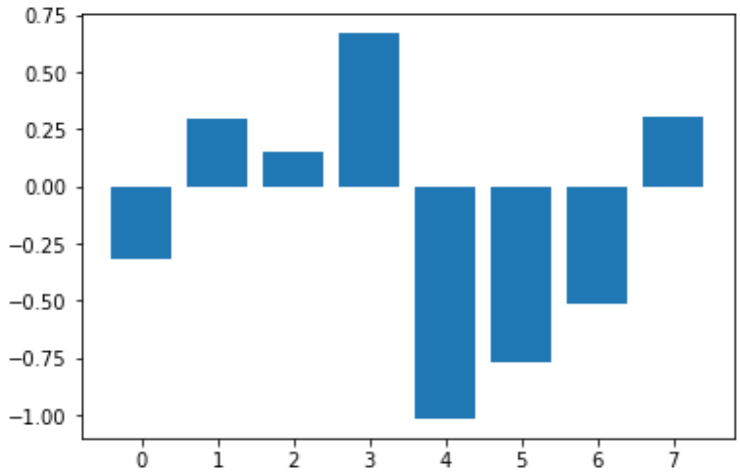


Fig 9: Feature importance bar plot

By looking at the feature importance bar plot, we can say that Column with position 3 ,4 and 5, which are Blair, Hague and Europe have important role to play in predictions

Perform LDA

LDA can be derived from simple probabilistic models which model the class conditional distribution of the data P(X|y=k) for each class k .Predictions can then be obtained by using Bayes' rule:

P(y=k|X)=P(X|y=k)P(y=k)P(X)=P(X|y=k)P(y=k)∑lP(X|y=l)·P(y=l) and we select the class k which maximizes this conditional probability.

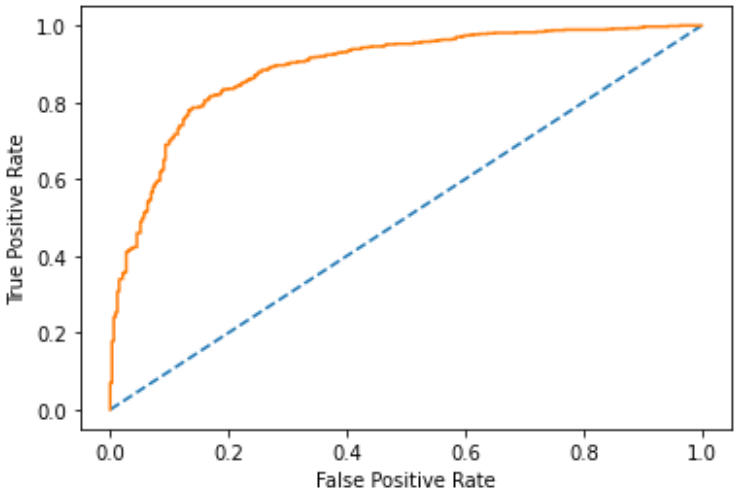
More specifically, for linear discriminant analysis, P(X|y) is modeled as a multivariate Gaussian distribution with density:

P(X|y=k)=1(2π)d/2|Σk|1/2exp(−12(X−μk)tΣ−1k(X−μk)) where d is the number of features.

source: scikit-learn

We performed LDS and following results were drawn out of model:

AUC for training data for LDA Model is : 0.889



AUC for testing data for LDA Model is : 0.884

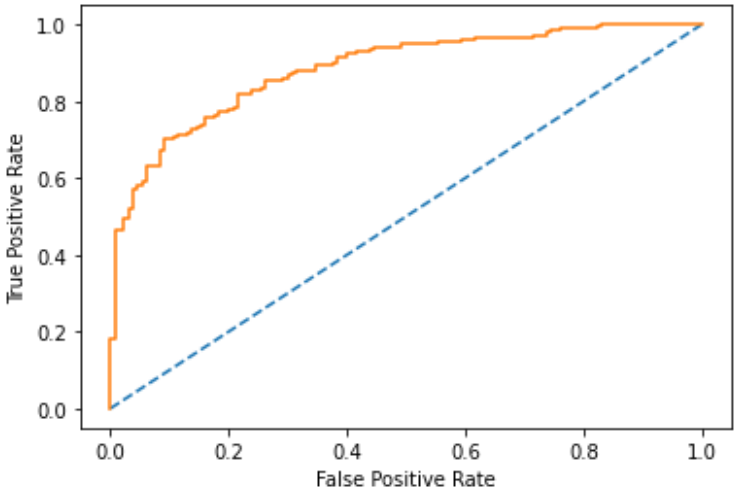


Fig 10: AUC Curve for Train and Test data for LDA Model

Classification report for Train data is as follows:

	precision	recall	f1-score	support
0	0.76	0.70	0.73	332
1	0.87	0.90	0.88	735
accuracy			0.84	1067
macro avg	0.81	0.80	0.81	1067
weighted avg	0.83	0.84	0.84	1067

```
Classification report for Test data is as follows:
              precision    recall  f1-score   support

           0       0.69      0.66      0.67       130
           1       0.87      0.88      0.87       328

    accuracy                           0.82       458
   macro avg       0.78      0.77      0.77       458
weighted avg       0.82      0.82      0.82       458
```

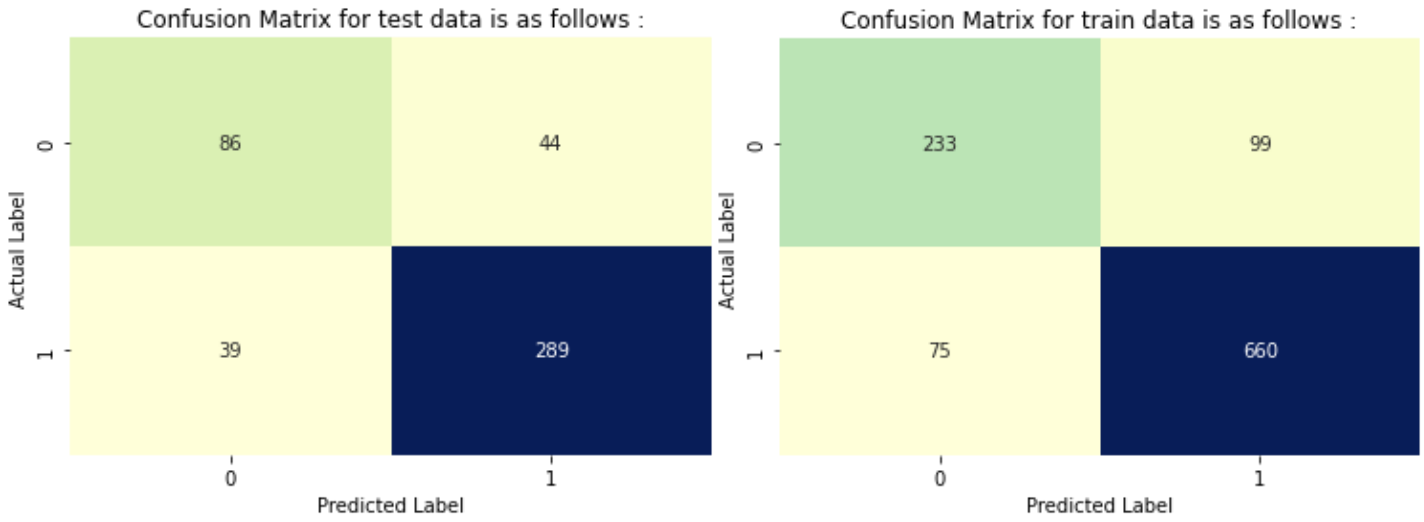


Fig 11: Confusion Matrix for Train and Test data for LDA Model

Features importance:

```
Feature: 0, Score: -0.40242
Feature: 1, Score: 0.30754
Feature: 2, Score: 0.13977
Feature: 3, Score: 0.82730
Feature: 4, Score: -1.18993
Feature: 5, Score: -0.85234
Feature: 6, Score: -0.61982
Feature: 7, Score: 0.24908
```

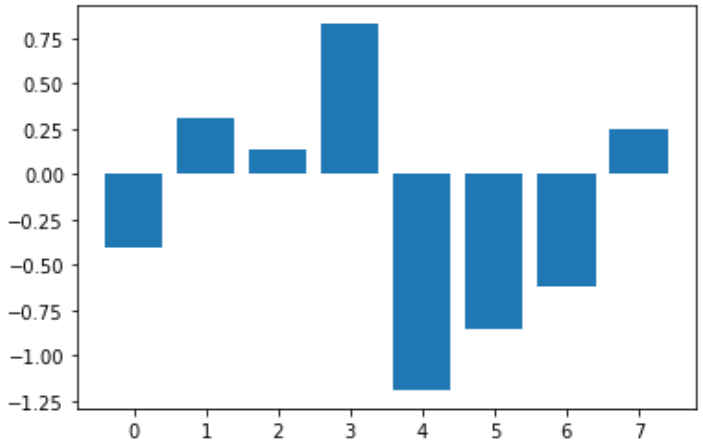


Fig 12: Feature importance bar plot

By looking at the feature importance bar plot, we can say that Column with position 3 ,4 and 5, which are Blair, Hague and Europe have important role to play in predictions

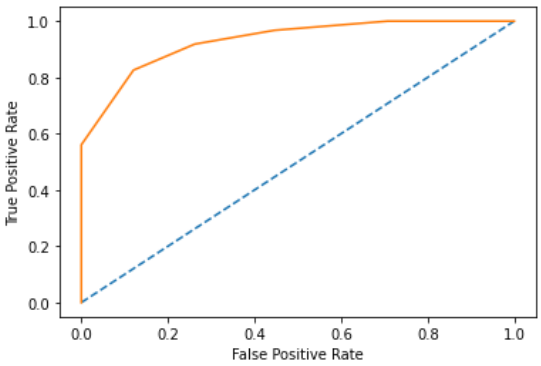
1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results

Perform KNN Model

Neighbors-based classification is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point.

After performing Model , following Model features were drawn:

```
AUC for training data for KNN Model is : 0.930
```




```
AUC for testing data for KNN Model is : 0.868
```

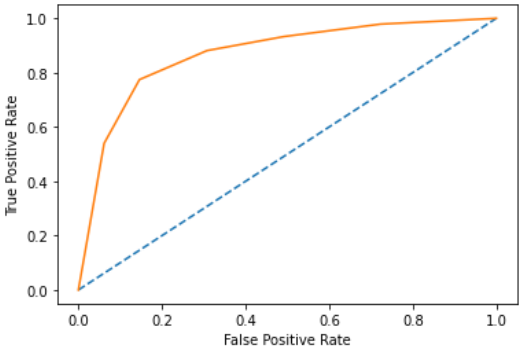


Fig 13: AUC Curve for Train and Test data for KNN Model

```
Classification report for Train data is as follows:
              precision    recall  f1-score   support

           0       0.80      0.74      0.77       332
           1       0.89      0.92      0.90       735

    accuracy                           0.86      1067
   macro avg       0.84      0.83      0.84      1067
weighted avg       0.86      0.86      0.86      1067


Classification report for Test data is as follows:
              precision    recall  f1-score   support

           0       0.70      0.69      0.69       130
           1       0.88      0.88      0.88       328

    accuracy                           0.83       458
   macro avg       0.79      0.79      0.79       458
weighted avg       0.83      0.83      0.83       458
```

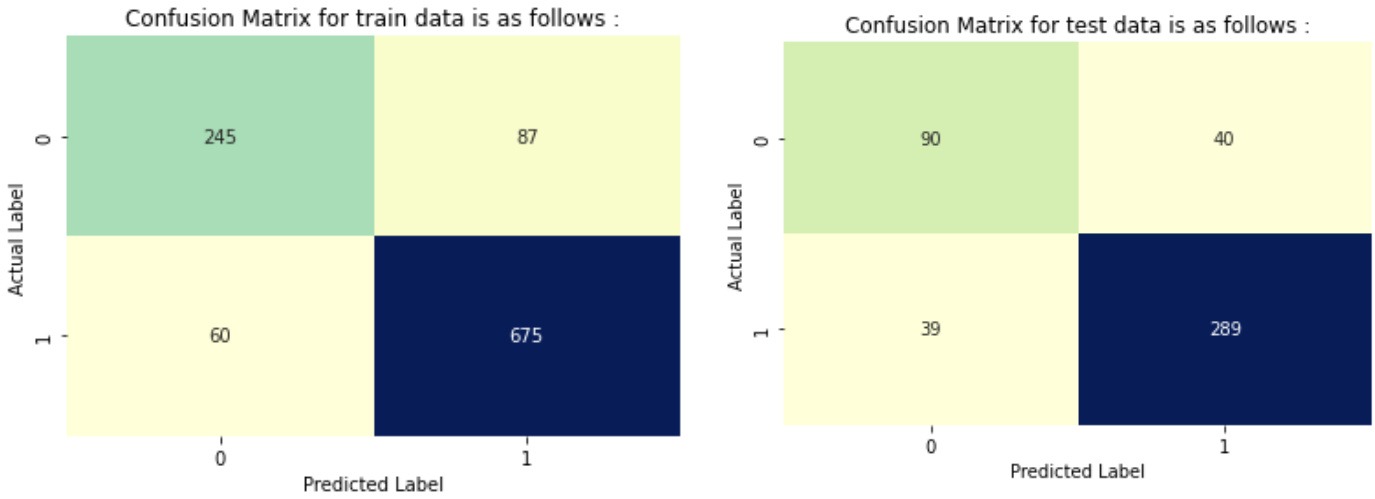


Fig 14: Confusion Matrix for Train and Test data for KNN Model

KNN with different K values

We have chosen multiple K value for K nearest neighbors, we have tried with 1 to 21 Odd K values . Following are the MSE were calculated with different values of K:

```
[0.22489082969432317,
 0.18558951965065507,
 0.17248908296943233,
 0.1746724890829694,
 0.17248908296943233,
 0.18122270742358082,
 0.18777292576419213,
 0.1746724890829694,
 0.17903930131004364,
 0.17903930131004364]
```

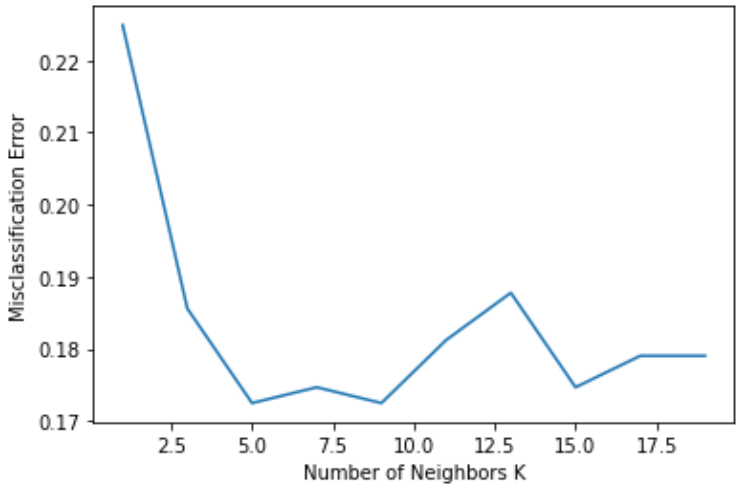
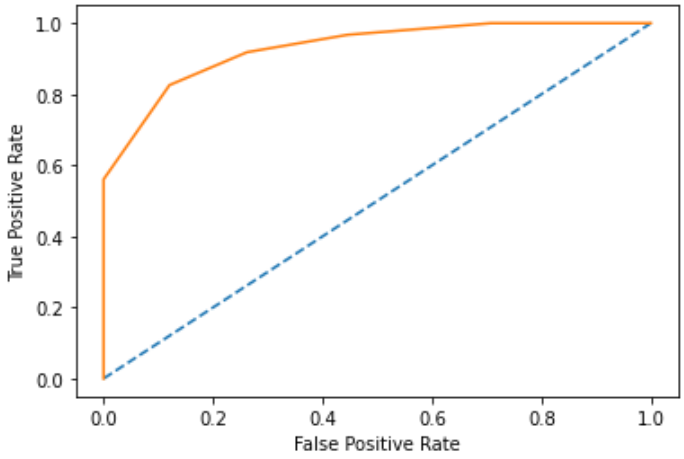


Fig 15: Performance Metrics for different K value KNN

We can clearly see that mis classification error is least for K neighbor as 5 and 7, So we will choose 5 neighbors for better computations.

Lets fit the model with training data once again and matrices again:

```
AUC for training data for KNN_5 Model is : 0.930
```




```
AUC for testing data for KNN_5 Model is : 0.868
```

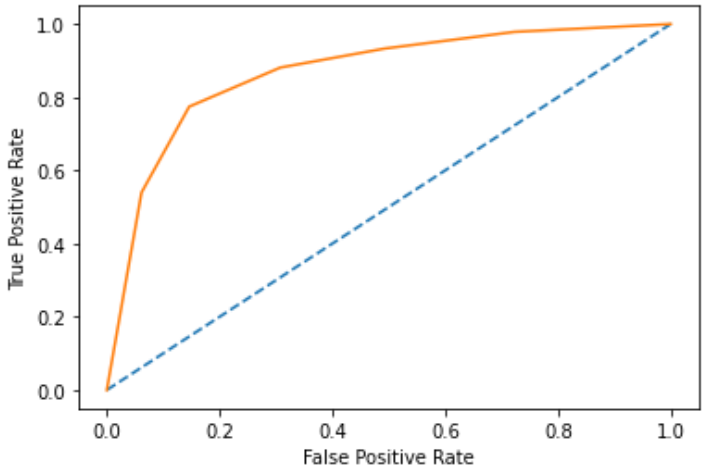


Fig 16: AUC Curve for Train and Test data for KNN, K=5 Model

```
Classification report for Train data is as follows:
              precision    recall  f1-score   support

           0       0.80      0.74      0.77       332
           1       0.89      0.92      0.90       735

    accuracy                           0.86      1067
   macro avg       0.84      0.83      0.84      1067
weighted avg       0.86      0.86      0.86      1067


Classification report for Test data is as follows:
              precision    recall  f1-score   support
```

0	0.70	0.69	0.69	130
1	0.88	0.88	0.88	328
accuracy			0.83	458
macro avg	0.79	0.79	0.79	458
weighted avg	0.83	0.83	0.83	458

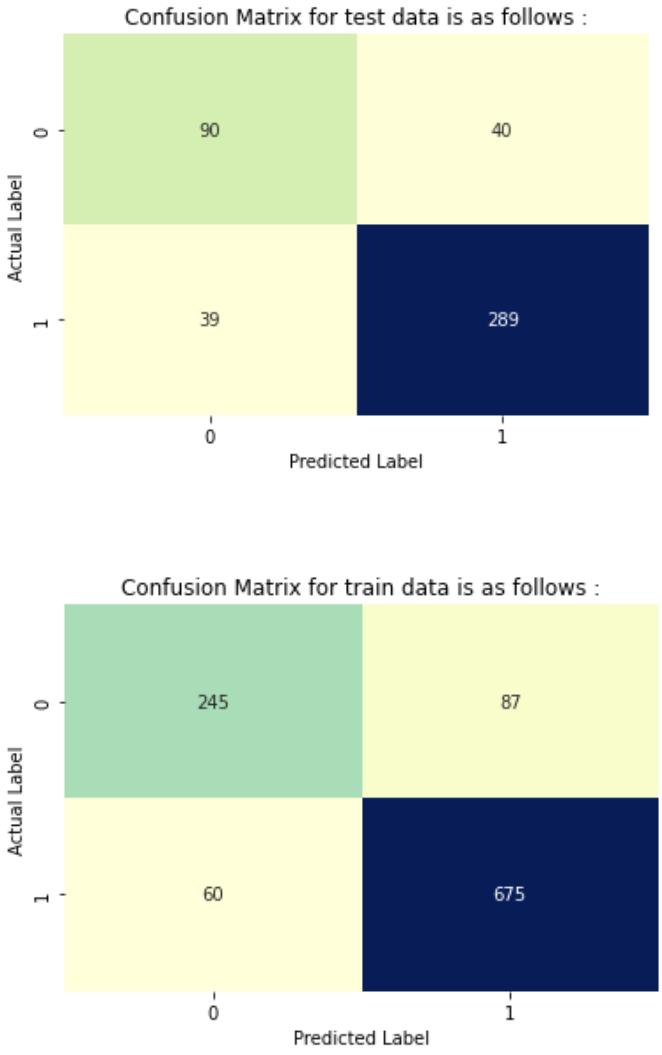


Fig 17: Confusion Matrix for Train and Test data for KNN, K=5 Model

Still we are seeing same Accuracy for train and test data set

Naïve Bayes Model

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship, given class variable y and dependent feature vector x1 through xn

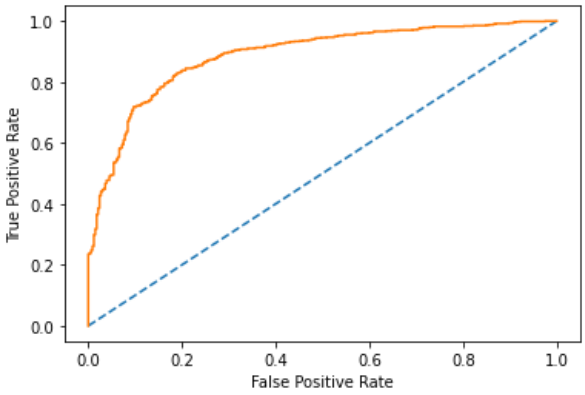
P(y|x1,…,xn)=P(y)P(x1,…xn|y)P(x1,…,xn) Advantages: In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters.

Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.

GaussianNB implements the Gaussian Naive Bayes algorithm for classification

source: scikit-learn

```
AUC for training data for Naive Model is : 0.886
```




```
AUC for testing data for Naive Model is : 0.885
```

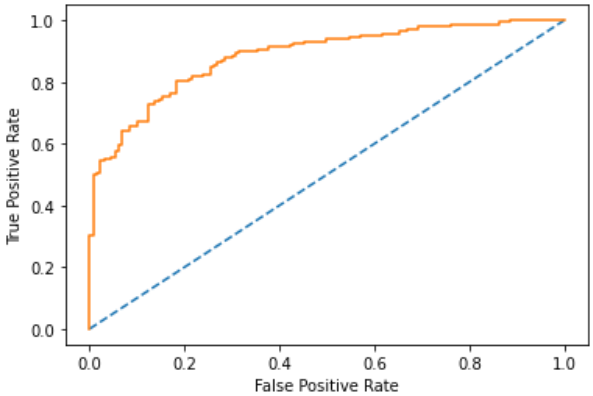


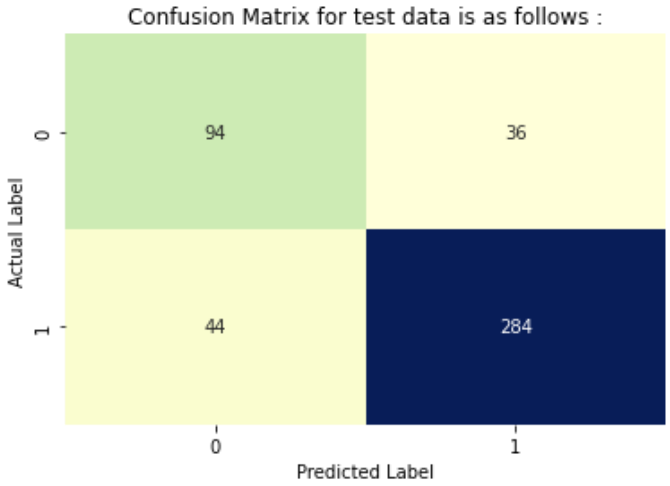
Fig 18: AUC Curve for Train and Test data for Naïve Model

Classification report for Train data is as follows:

	precision	recall	f1-score	support
0	0.74	0.72	0.73	332
1	0.88	0.88	0.88	735
accuracy			0.83	1067
macro avg	0.81	0.80	0.80	1067
weighted avg	0.83	0.83	0.83	1067

Classification report for Test data is as follows:

	precision	recall	f1-score	support
0	0.68	0.72	0.70	130
1	0.89	0.87	0.88	328
accuracy			0.83	458
macro avg	0.78	0.79	0.79	458
weighted avg	0.83	0.83	0.83	458



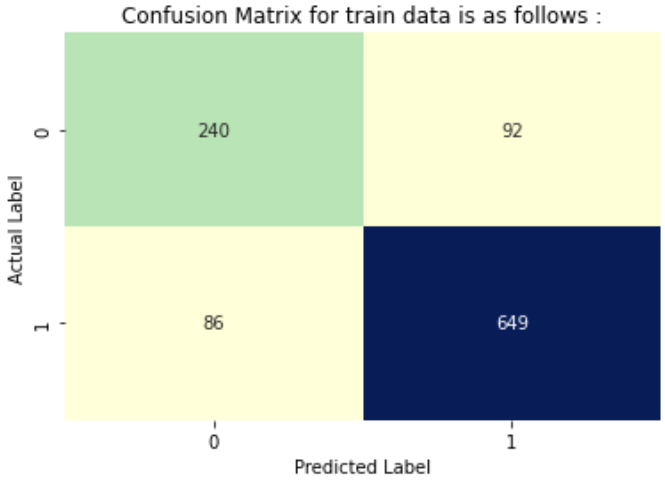


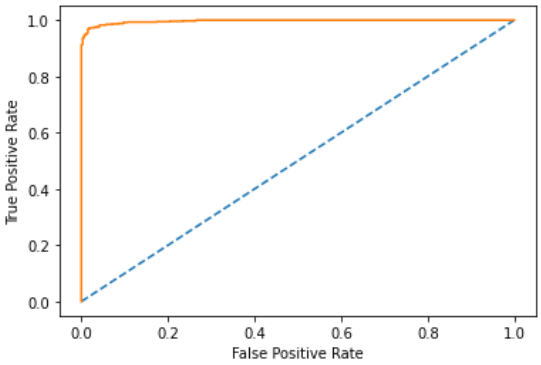
Fig 19: Confusion Matrix for Train and Test data for Naïve Model

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

Bagging Classifier (Random Forest should be applied for Bagging)

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Also, as per requirements, we have used random forest classifier for the calculation.

```
AUC for training data for bgcl Model is : 0.997
```




```
AUC for testing data for bgcl Model is : 0.897
```

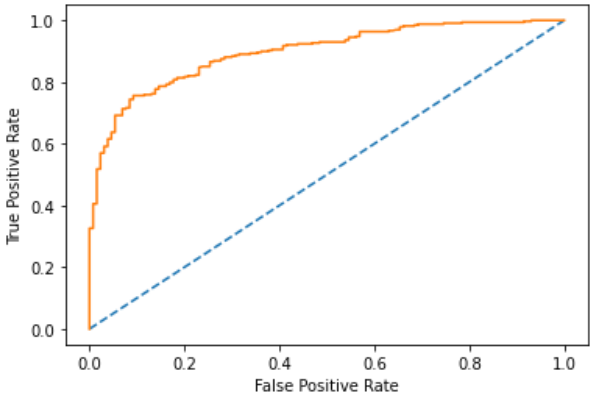
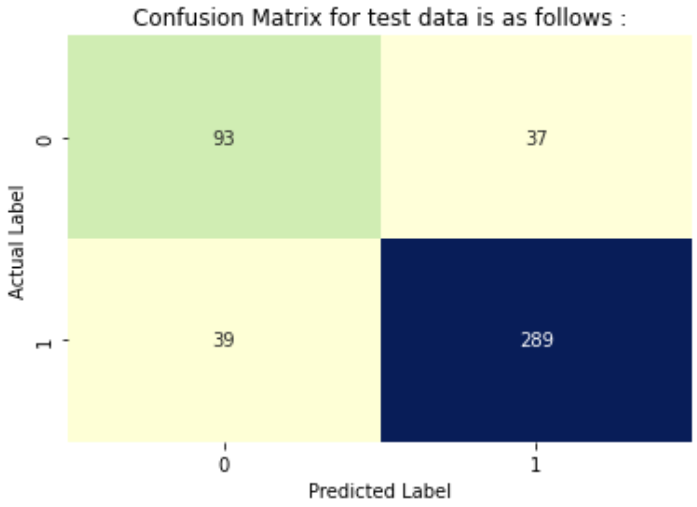


Fig 20: AUC Curve for Train and Test data for Bagging Model

```
Classification report for Train data is as follows:
              precision    recall  f1-score   support

           0       0.97      0.91      0.94       332
           1       0.96      0.99      0.97       735

    accuracy                           0.96      1067
   macro avg       0.97      0.95      0.96      1067
weighted avg       0.96      0.96      0.96      1067

Classification report for Test data is as follows:
              precision    recall  f1-score   support

           0       0.70      0.72      0.71       130
           1       0.89      0.88      0.88       328

    accuracy                           0.83       458
   macro avg       0.80      0.80      0.80       458
weighted avg       0.83      0.83      0.83       458
```



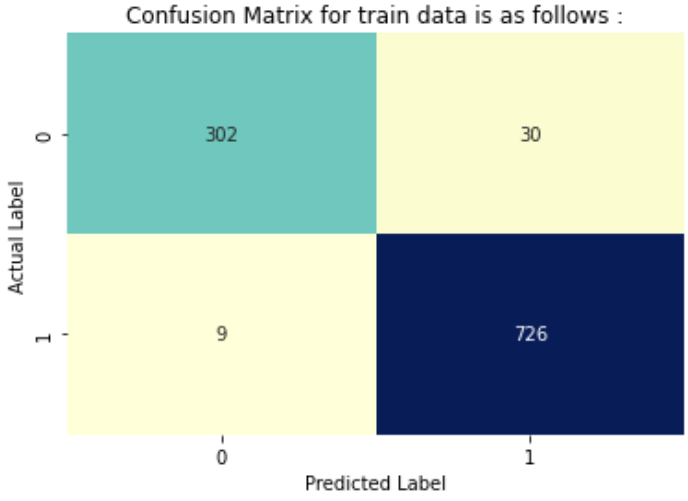


Fig 21: Confusion Matrix for Train and Test data for Bagging Model

Ada Boosting

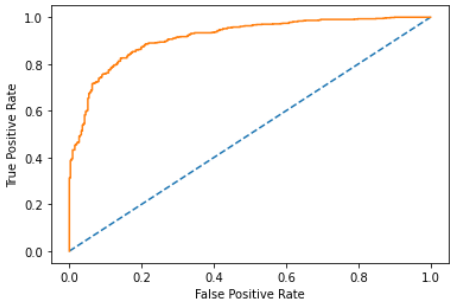
Ada Boosting is called Adaptive Boosting. An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. In AdaBoost, the successive learners are created with a focus on the ill fitted data of the previous learner. Each successive learner focuses more and more on the harder to fit data i.e. their residuals in the previous tree

The above shown parameters are used for the model building. Here,

• n_estimators=100: The maximum number of estimators at which boosting is terminated. In case of perfect fit, the learning procedure is stopped early. The default value of n_estimators is 50, here we have given for more learning of the model.

• random_state =1: random_state is used to pass an int for reproducible output across multiple function calls. We are free to give any value for the random_state but ensure to give the same value everywhere

```
AUC for training data for ADB Model is : 0.913
```




```
AUC for testing data for ADB Model is : 0.879
```

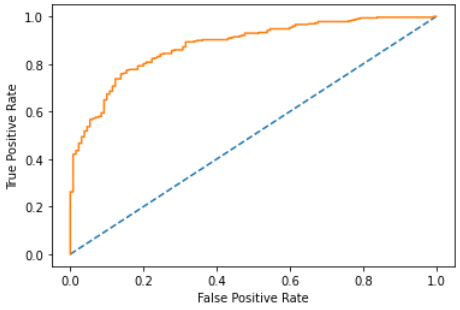


Fig 22 : AUC Curve for Train and Test data for Ada Boosting Model

Classification report for Train data is as follows:

	precision	recall	f1-score	support
0	0.78	0.72	0.74	332
1	0.88	0.91	0.89	735
accuracy			0.85	1067
macro avg	0.83	0.81	0.82	1067
weighted avg	0.84	0.85	0.85	1067

Classification report for Test data is as follows:

	precision	recall	f1-score	support
0	0.68	0.69	0.68	130
1	0.88	0.87	0.87	328
accuracy			0.82	458
macro avg	0.78	0.78	0.78	458
weighted avg	0.82	0.82	0.82	458

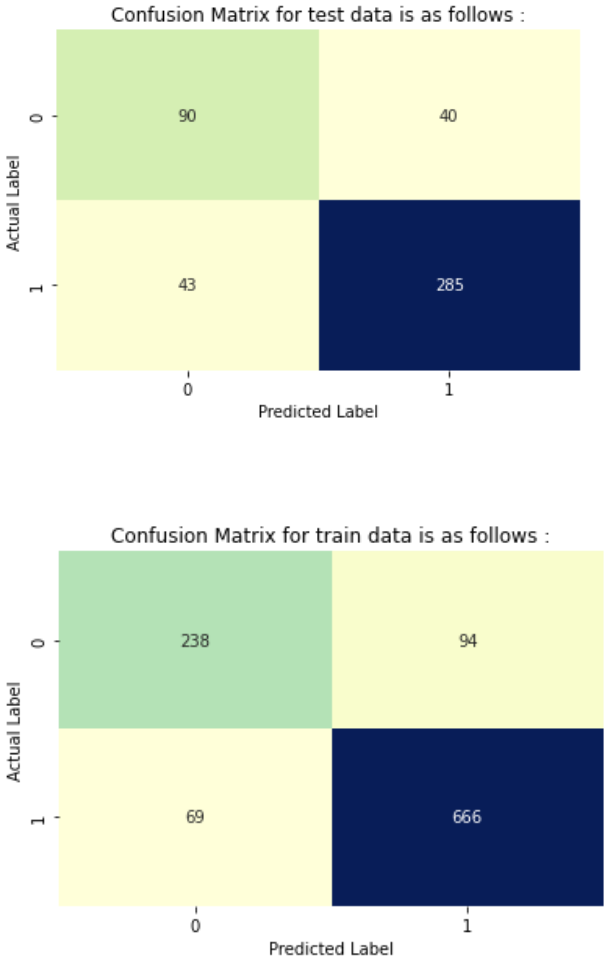
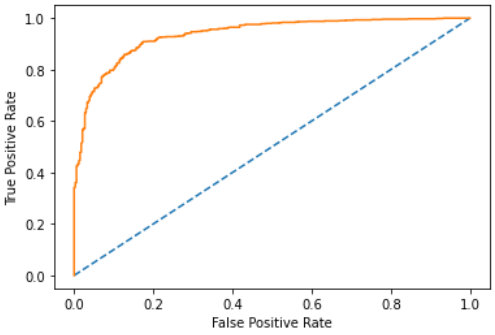


Fig 23: Confusion Matrix for Train and Test data for Ada boosting Model

Gradient Boosting

Gradient Boosting is another type of Boosting technique where each learner is fit on a modified version of original data (original data is replaced with the x values and residuals from previous learner). By fitting new models to the residuals, the overall learner gradually improves in areas where residuals are initially high.

```
AUC for training data for GDB Model is : 0.936
```




```
AUC for testing data for GDB Model is : 0.907
```

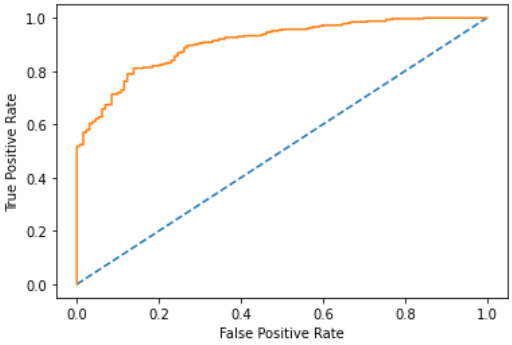


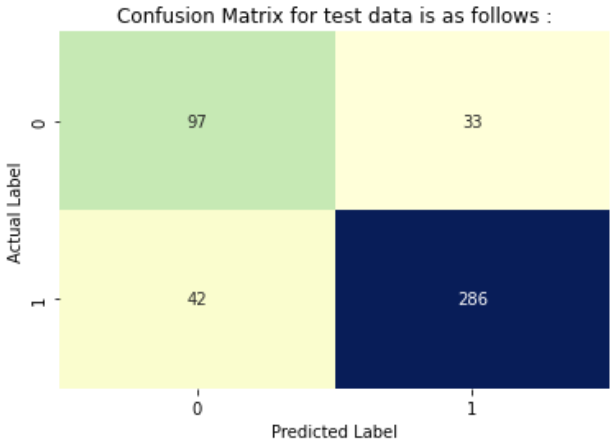
Fig 24: AUC Curve for Train and Test data for Gradient Boosting Model

Classification report for Train data is as follows:

	precision	recall	f1-score	support
0	0.83	0.75	0.79	332
1	0.89	0.93	0.91	735
accuracy			0.87	1067
macro avg	0.86	0.84	0.85	1067
weighted avg	0.87	0.87	0.87	1067

Classification report for Test data is as follows:

	precision	recall	f1-score	support
0	0.70	0.75	0.72	130
1	0.90	0.87	0.88	328
accuracy			0.84	458
macro avg	0.80	0.81	0.80	458
weighted avg	0.84	0.84	0.84	458



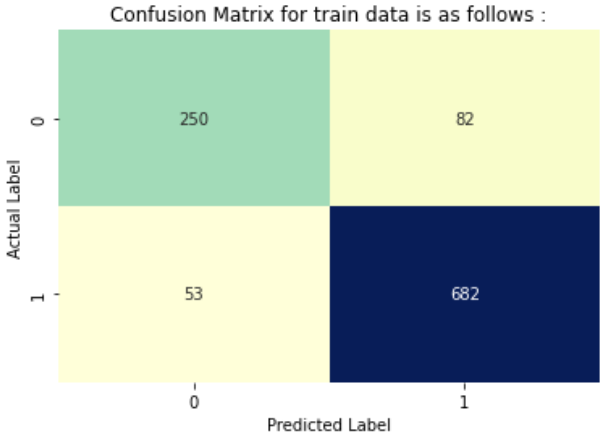


Fig 25: Confusion Matrix for Train and Test data for Gradient Boosting Model

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.

We have performed many Models of our training and testing data and . and also calculated Confusion matrices, and performance matrices for all the models. Following are the results for all the models:

```
logit_train_precision  0.87
logit_train_recall  0.91
logit_train_f1  0.89
logit_accuracy_train 0.84
logit_auc_train 0.88
logit_test_precision  0.87
logit_test_recall  0.89
logit_test_f1  0.88
logit_accuracy_test 0.82
logit_auc_test 0.89

LDA_train_precision  0.87
LDA_train_recall  0.9
LDA_train_f1  0.88
LDA_accuracy_train 0.84
LDA_auc_train 0.88
LDA_test_precision  0.87
LDA_test_recall  0.88
LDA_test_f1  0.87
LDA_accuracy_test 0.82
LDA_auc_test 0.89

KNN_train_precision  0.89
KNN_train_recall  0.92
KNN_train_f1  0.9
KNN_accuracy_train 0.86
KNN_auc_train 0.87
KNN_test_precision  0.88
KNN_test_recall  0.88
KNN_test_f1  0.88
KNN_accuracy_test 0.83
KNN_auc_test 0.93

Naive_train_precision  0.88
Naive_train_recall  0.88
Naive_train_f1  0.88
Naive_accuracy_train 0.83
Naive_auc_train 0.88
Naive_test_precision  0.89
Naive_test_recall  0.87
Naive_test_f1  0.88
```

```
Naive_accuracy_test 0.83
Naive_auc_test 0.89

bgcl_train_precision  0.96
bgcl_train_recall  0.99
bgcl_train_f1  0.97
bgcl_accuracy_train 0.96
bgcl_auc_train 0.9
bgcl_test_precision  0.89
bgcl_test_recall  0.88
bgcl_test_f1  0.88
bgcl_accuracy_test 0.83
bgcl_auc_test 1.0

ADB_train_precision  0.88
ADB_train_recall  0.91
ADB_train_f1  0.89
ADB_accuracy_train 0.85
ADB_auc_train 0.88
ADB_test_precision  0.88
ADB_test_recall  0.87
ADB_test_f1  0.87
ADB_accuracy_test 0.82
ADB_auc_test 0.91

GDB_train_precision  0.89
GDB_train_recall  0.93
GDB_train_f1  0.91
GDB_accuracy_train 0.87
GDB_auc_train 0.91
GDB_test_precision  0.9
GDB_test_recall  0.87
GDB_test_f1  0.88
GDB_accuracy_test 0.84
GDB_auc_test 0.94
```

Lets frame them all in the data frame and see the data into tabular format :

	index	Accuracy	AUC	Recall	Precision	F1 Score
0	Logistic regression Train	0.87	0.91	0.89	0.84	0.88
1	LDA regression Train	0.87	0.90	0.88	0.84	0.88
2	KNN regression Train	0.89	0.92	0.90	0.86	0.87
3	Naive regression Train	0.88	0.88	0.88	0.83	0.88
4	Bagging regression Train	0.96	0.99	0.97	0.96	0.90
5	AdaBoosting regression Train	0.88	0.91	0.89	0.85	0.88
6	Gredient Boost regression Train	0.89	0.93	0.91	0.87	0.91
7	Logistic regression Test	0.87	0.89	0.88	0.82	0.89
8	LDA regression Test	0.87	0.88	0.87	0.82	0.89
9	KNN regression Test	0.88	0.88	0.88	0.83	0.93
10	Naive regression Test	0.89	0.87	0.88	0.83	0.89
11	Bagging regression Test	0.89	0.88	0.88	0.83	1.00
12	AdaBoosting regression Test	0.88	0.87	0.87	0.82	0.91
13	Gredient Boost regression Test	0.90	0.87	0.88	0.84	0.94

inference:

1. Accuracy wise all the models are performing really well, because all the models have accuracy of ore than 85% in Train and test data both
2. Best accuracy for Test data from "Gradient Boost regression" Model with 90% and train data it is 89%, so it is performing well on train and test data both.
3. Best accuracy for Train data from "Bagging regression" Model with 96%, which is based on Random forest classifier, but it has less accuracy with test data of 89%, so it seems slightly over fit the data
4. I would not say, that Logistic and LDA are the worst Model with test data, but as compare to other models, their accuracy level is slightly poor, they have 87% of accuracy with test data as well as 87% of accuracy with Train data. So both train and test data are performing equally good, so these models are also performing well.
5. Similarly, Ada boosting is also performing same to both train and test data with accuracy level of 88% in both data set.
6. KNN and Naive based models are also performing 88-89% in both Train and test data set, not bad at all.
7. Best Area Under the Curve for train data is Bagging Model, which has 99% AUC in train data, but it has 88% of AUC in Test data, so as Accuracy also indicate , this model is slightly Over fit.
8. Best Recall is same for Gradient boosting, KNN, Naive and Logistic Model with 88% of Recall, which is good
9. Gradient Boosting have best precision of 84% with test data

10. Bagging have Best F1 score as 1, which is 100%, but this is Slightly Over fit Model for accuracy. 2nd best F1 score is for Gradient boosting

Final Selection should be done with Cooperation of Train and test data both result set, and in my opinion , best fit model is "gradient boosting", which is giving proper results with both train and test data set

1.8 Based on these predictions, what are the insights?

Insights:

Based on best Model selection as "Gradient Boosting", And we have see the prediction classes for Test data set. Following are the insights can be given based on the predictions:

1. We can see that Majority of the Voters are Voting to Labourse with number of 319 out of total 458 records. SO "Labourse " parties really need to work hard on ground level and Start doing campaigns
2. At economic level 0 of national condition, Labourse are doing better than Conservatives, but at other levels, they are not doing good,
3. At economic condition on household level 3, majority of Voters choosing to Conservatives, So Labourse need to attract Middle class Voters, which can be a good winning factors and work on some attractive schemes, Like Employments, security , something like that.
4. Voters are NOT liking "Labourse " leaders for the Blair categories of Level 4 and 5 , Voters are choosing Conservative Leader in place of them, so "Labourse " need to work on that group of leaders.
5. Voters are liking a lot Conservative leaders for Hague level 2 are playing Majority role in winning ,Conservatives need to maintain this group.
6. Female voters are choosing Conservatives more , "Labourse " needs to work on Women empowerment, security, Employment, Education , something like this to attract more Female Voters.
7. Both the current national economic conditions and household economic conditions are at an average level right now, So there is a need for a lot of reforms to be developed to improve these conditions. Both the political parties should bring in new strategies to improve the conditions and propagate it to their voters efficiently.

2. Problem 2: In this particular project, we are going to work on the inaugural corpora from the nltk in Python. We will be looking at the following speeches of the Presidents of the United States of America:

President Franklin D. Roosevelt in 1941

President John F. Kennedy in 1961

President Richard Nixon in 1973

INTRODUCTION

The purpose of this whole exercise is to various techniques involved in text analytics to read the test data and do various operations on the text data such as counting the number of words, sentences, characters etc. We can even do sentimental analysis on the data after learning the nature of the text.

The purpose of this whole exercise is to various techniques involved in text analytics to read the test data and do various operations on the text data such as counting the number of words, sentences, characters etc

We have downloaded Speeches for all the data set and have about 59 Speeches.

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

For the asked speeches , Following are the results for number of words, characters and sentences :

```
Number of words in Text file 1941-Roosevelt.txt is  1526
Number of words in Text file 1941-Roosevelt.txt is  1543
Number of words in Text file 1941-Roosevelt.txt is  2006
```

```
Number of Sentences in Text file 1941-Roosevelt.txt is  68
Number of Sentences in Text file 1941-Roosevelt.txt is  52
Number of Sentences in Text file 1941-Roosevelt.txt is  68
```

```
Number of Characters in Text file 1941-Roosevelt.txt is  7571
Number of Characters in Text file 1941-Roosevelt.txt is  7618
Number of Characters in Text file 1941-Roosevelt.txt is  9991
```

This is just a preliminary examination of the data, without removing white spaces, punctuation marks and other bad hidden characters . Lets form all data into a Data Frame for better analysis

	serial_number	president	text
0	1	1941-Roosevelt	On each national day of inauguration since 178...
1	2	1961-Kennedy	Vice President Johnson, Mr. Speaker, Mr. Chief...
2	3	1973-Nixon	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...

Add no of characters in the data frame:

	serial_number	president	text	character_counts
0	1	1941-Roosevelt	On each national day of inauguration since 178...	7571
1	2	1961-Kennedy	Vice President Johnson, Mr. Speaker, Mr. Chief...	7618
2	3	1973-Nixon	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...	9991

Add number of words in the data frame:

	serial_number	president	text	character_counts	word_counts
0	1	1941-Roosevelt	On each national day of inauguration since 178...	7571	1323
1	2	1961-Kennedy	Vice President Johnson, Mr. Speaker, Mr. Chief...	7618	1364
2	3	1973-Nixon	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...	9991	1769

Add sentences in the data frame:

	serial_number	president	text	character_counts	word_counts	sentences_counts
0	1	1941-Roosevelt	On each national day of inauguration since 178...	7571	1323	68
1	2	1961-Kennedy	Vice President Johnson, Mr. Speaker, Mr. Chief...	7618	1364	52
2	3	1973-Nixon	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...	9991	1769	68

2.2 Remove all the stop words from all three speeches

We have downloaded the Stop words , and punctuations, words from NLTK.Corpus library.

We have also extended the stop words, with following words:

"http","bit","bitly","bit ly", "dear", "im", "i'm", "please","the","of","and",'mr','on','it','in','let','to','us','shall'

Lets remove the Stop words, from the Text column of the data frame, which is capturing the speeches:

Following are the the number of words, characters, sentences , after removing the Stop words , punctuations :

	serial_number	president	text	character_counts	word_counts	sentences_counts
0	1	1941-Roosevelt	national day inauguration since 1789 people re...	4556	617	1
1	2	1961-Kennedy	vice president johnson speaker chief justice p...	4635	658	1
2	3	1973-Nixon	vice president speaker chief justice senator c...	5733	775	1

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)

Top three words in the 1941-Roosevelt speech are as follows:

```
nation    11
know      10
spirit     9
```

Top three words in the 1961-Kennedy speech are as follows:

```
sides    8
world    8
new      7
```

Top three words in the 1973-Nixon speech are as follows:

```
peace    19
world    16
new      15
```

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

Fig 26: World Cloud for 1941-Roosvelt's Speech

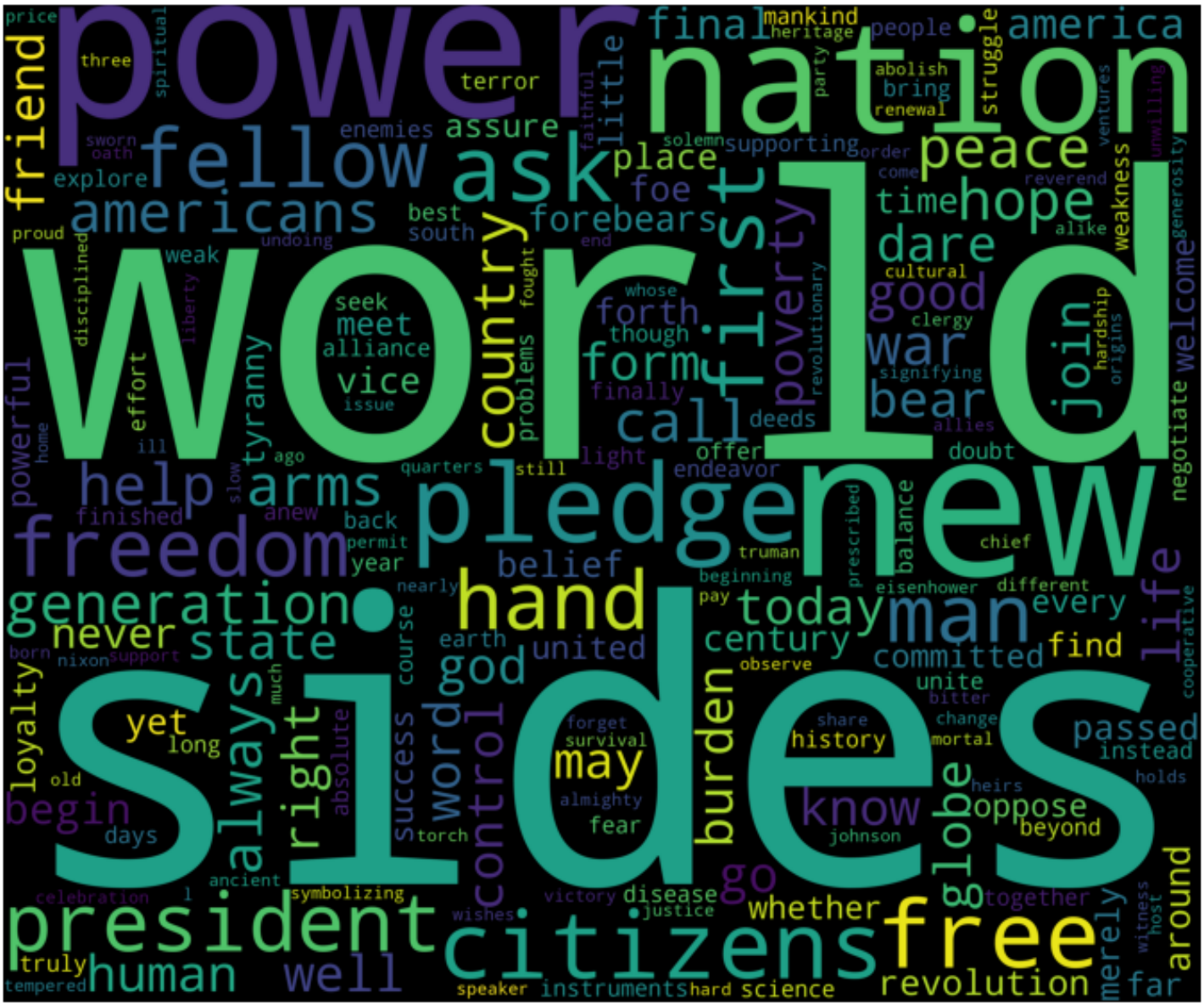


Fig 27: World Cloud for 1961-kennedy's Speech

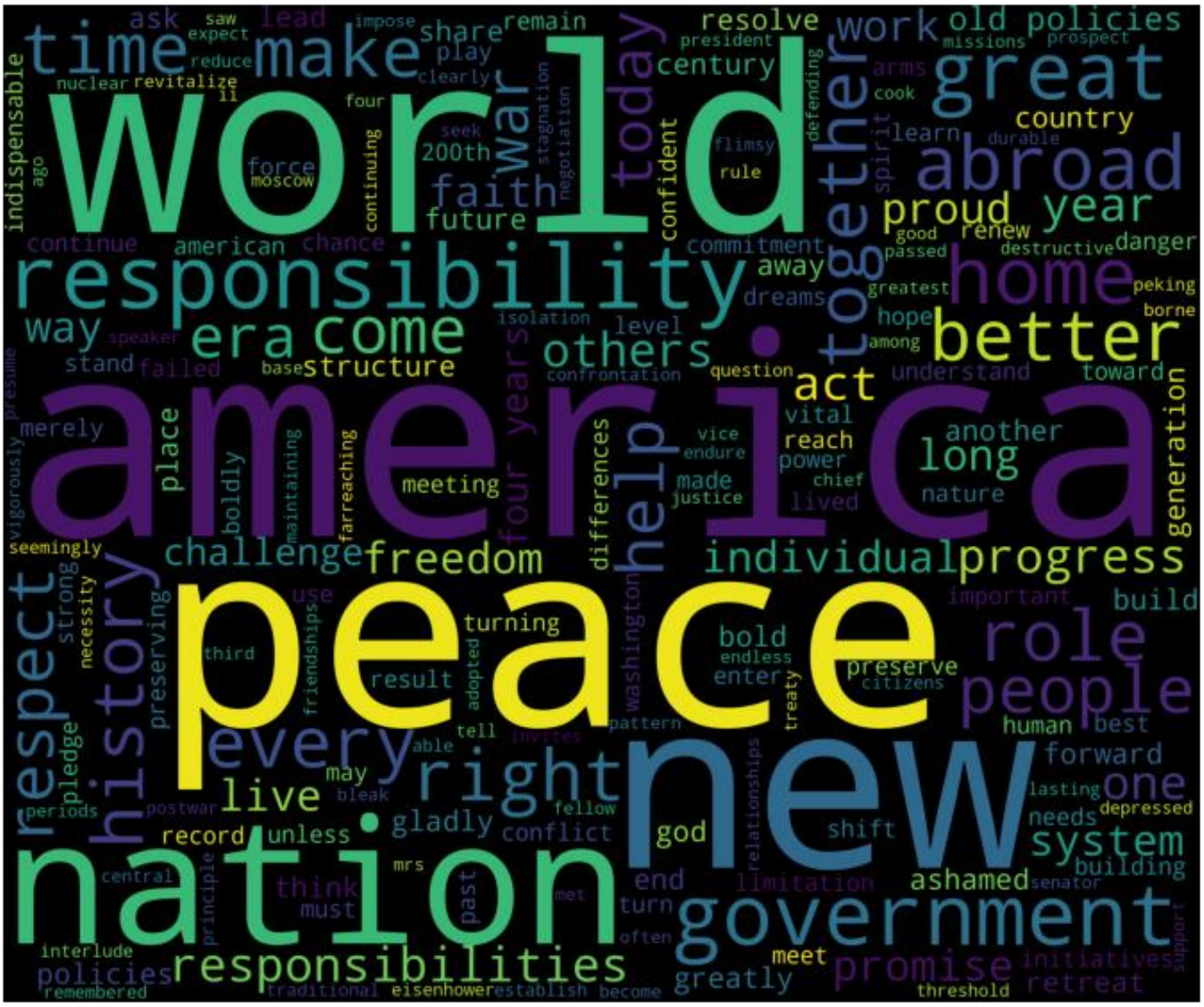


Fig 28: World Cloud for 1973-Nixon's Speech