Machine Learning
Project Report

Contents

List of Figures

ML PROJECT

Problem 1:

You are hired by one of the leading news channels CNBE who wants to analyze recent elections. This survey was conducted on 1525 voters with 9 variables. You have to build a model, to predict which party a voter will vote for on the basis of the given information, to create an exit poll that will help in predicting overall win and seats covered by a particular party.

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

Data Types:

```
<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
```

```
int64     7
object    2
dtype: int64
```

Null Values

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

Shape:
We have 1524 rows and 9 columns in our Data set.

Observation:

- There are a total of 1524 rows and 9 columns in the dataset. Out of 9, 7 are integer and 2 are object type variable.
- There is no null values in any column.
- I have dropped the 'unamed' column from the dataset since it has no meaningful for our analysis.
- The data set had 8 duplicated values. So I dropped them.

Checking for missing values:

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

There are no missing values.

Checking for duplicated Values:

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

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

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

There are 8 duplicated values and I dropped them.

Summary:

	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

Skewness

```
vote                        0.857014
age                         0.139800
economic.cond.national     -0.238474
economic.cond.household    -0.144148
Blair                      -0.539514
Hague                       0.146191
Europe                     -0.141891
political.knowledge        -0.422928
gender                      0.130929
dtype: float64
```

Here, we can see that there isn't much skewness in the data.

- The value of 'Blair' is a little bit higher than -0.5.
- The value of 'Vote' is little bit higher than 0.5.
- The data overall, is symmetrical.

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

Null Value Check:

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

There are no null values present in the given data set.

Data Types:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1517 entries, 1 to 1525
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   vote                    1517 non-null   int64
 1   age                     1517 non-null   int64
 2   economic.cond.national  1517 non-null   int64
 3   economic.cond.household 1517 non-null   int64
 4   Blair                   1517 non-null   int64
 5   Hague                   1517 non-null   int64
 6   Europe                  1517 non-null   int64
 7   political.knowledge     1517 non-null   int64
 8   gender                  1517 non-null   int64
dtypes: int64(9)
memory usage: 118.5 KB
```

Shape:
We have 1517 rows and 9 columns in our Data set.

Univariate Analysis:

Description of Age:

```
count    1517.000000
mean       54.241266
std        15.701741
min        24.000000
25%        41.000000
50%        53.000000
75%        67.000000
max        93.000000
Name: age, dtype: float64
```

Histogram and Box Plot of Age

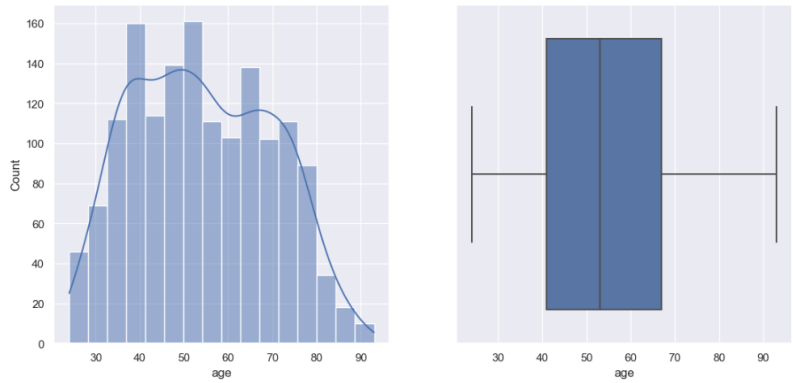


Figure 1: Histogram and Box Plot of Age

Observation:

- The data are normally distributed.
- Maximum number of people are aged between 40 and 70.
- There are no outliers present in this data
- The minimum value is 24 and maximum value is 93.
- The mean value is 54

Vote:

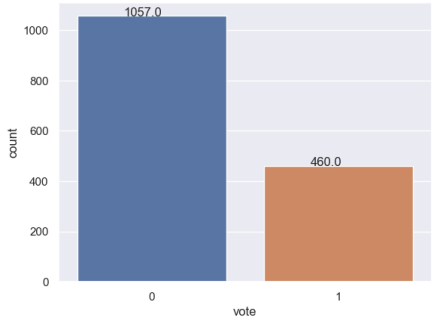


Figure 2: Count plot of Vote

```
0    1057
1     460
Name: vote, dtype: int64
```

Observation:

- Labor party has higher number of votes.
- Labor party has 1057 votes and Conservative party has 460 votes

economic.cond.national

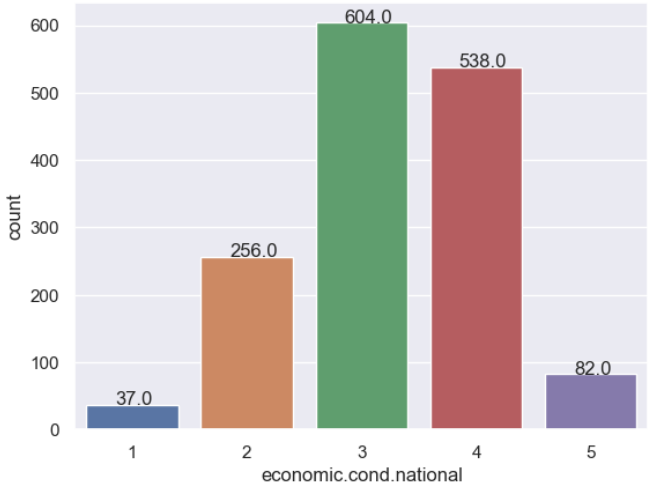


Figure 3: Count plot of the economic.cond.national

```
count    1517.000000
mean        3.245221
std         0.881792
min         1.000000
25%         3.000000
50%         3.000000
75%         4.000000
max         5.000000
Name: economic.cond.national, dtype: float64
```

Figure 4: Description of economic.cond.national

Observation:

- The top 2 variables are 3 and 4 having value of 604 and 538 respectively.
- 1 has the least value of 37.
- Mean of economic.cond.national is 3.245

economic.cond.household

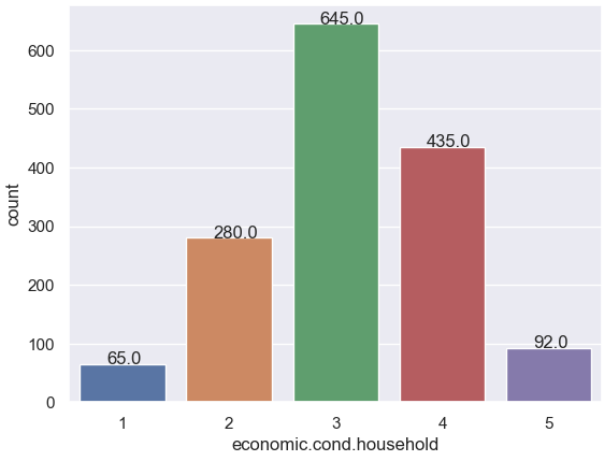


Figure 5: Count plot of the economic.cond.household

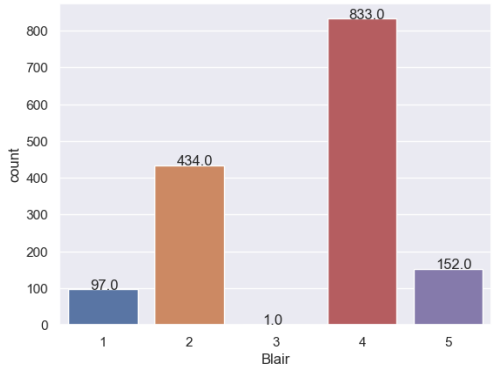
```
count    1517.000000
mean        3.137772
std         0.931069
min         1.000000
25%         3.000000
50%         3.000000
75%         4.000000
max         5.000000
Name: economic.cond.household, dtype: float64
```

Figure 6: Description of the economic.cond.household

Observation:

- The top 2 variables are 3 and 4 having value of 645 and 435 respectively.
- 1 has the least value which is 65
- The mean of economic.cond.household is 3.137

Blair

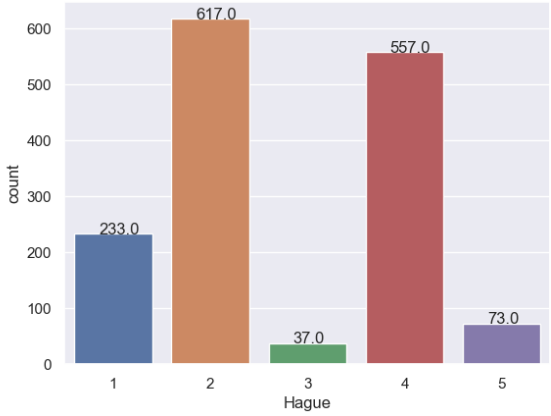



```
count    1517.000000
mean        3.335531
std         1.174772
min         1.000000
25%         2.000000
50%         4.000000
75%         4.000000
max         5.000000
Name: Blair, dtype: float64
```

Figure 7: Count plot and Description of Blair

Observation

- The top 2 variables are 4 and 2 having value of 833 and 434 respectively.
- 3 has the least value which is 1
- The mean of Blair is 3.335

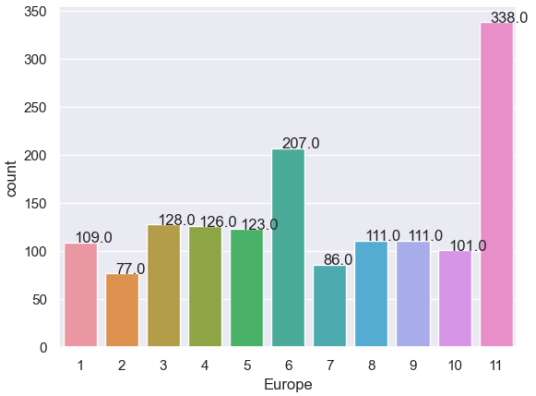



```
count    1517.000000
mean        2.749506
std         1.232479
min         1.000000
25%         2.000000
50%         2.000000
75%         4.000000
max         5.000000
Name: Hague, dtype: float64
```

Figure 8: Count plot and Description of Hague

Observation

- The top 2 variables are 2 and 4 having value of 617 and 557 respectively.
- 3 has the least value which is 37
- The mean of Hague is 2.749

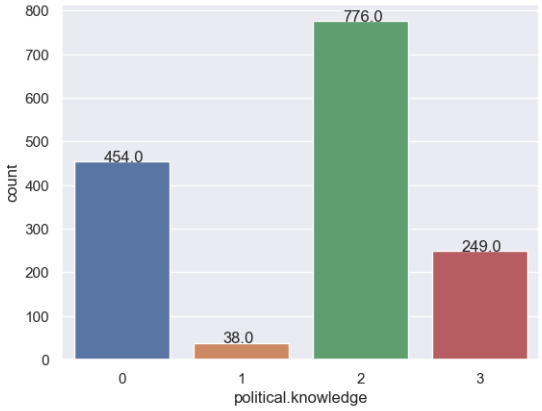



```
count    1517.000000
mean        6.740277
std         3.299043
min         1.000000
25%         4.000000
50%         6.000000
75%        10.000000
max        11.000000
Name: Europe, dtype: float64
```

Figure 9: Count plot and Description of Europe

Observation

- The top 2 variables are 11 and 6 having value of 388 and 207 respectively.
- 2 has the least value which is 77
- The mean of Europe is 6.740




```
count    1517.000000
mean        1.540541
std         1.084417
min         0.000000
25%         0.000000
50%         2.000000
75%         2.000000
max         3.000000
Name: political.knowledge, dtype: float64
```

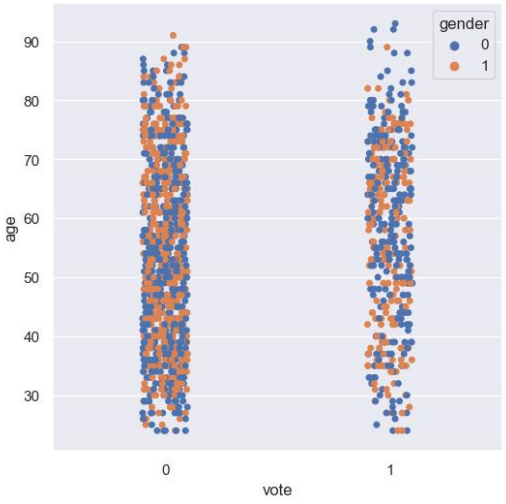
Figure 10: Count plot and Description of political.knowledge

Observation

- The top 2 variables are 2 and 1 having value of 776 and 454 respectively.
- 1 has the least value which is 38
- The mean of political.knowledge is 1.54

Bivariate Analysis

Vote & Age



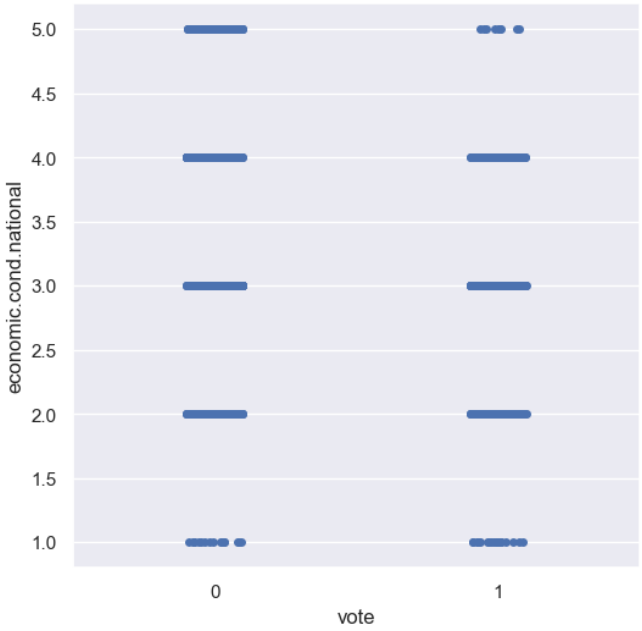

```
vote  gender
0     0         551
      1         506
1     0         257
      1         203
```

Figure 11: stripplot between Vote and Age

Observation

- We can clearly see that, the labour party has got more votes than the conservative party.
- In every age group, the labour party has got more votes than the conservation party.
- Female votes are considerably higher than the male votes in both parties.
- In both genders, the labour party has got more votes than the conservative.

Vote & economic.cond.national



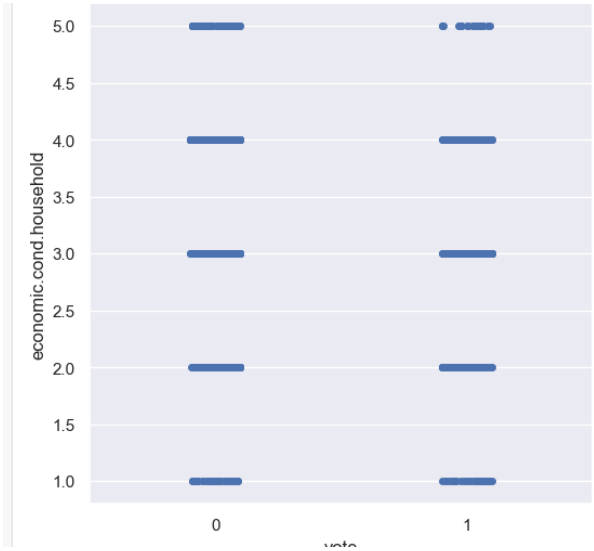
vote	economic.cond.national	
0	1	16
	2	116
	3	405
	4	447
	5	73
1	1	21
	2	140
	3	199
	4	91
	5	9

Figure 12: stripplot between Vote and economic.cond.national

Observation

- Labour party has higher votes overall.
- Out of 82 people who gave a score of 5, 73 people have voted for the labour party.
- Out of 538 people who gave a score of 4, 447 people have voted for the labour party. This is the highest set of people in the labour party.
- Out of 604 people who gave a score of 3, 405 people have voted for the labour party. This is the 2nd highest set of people in the labour party. The remaining 199 people who have voted for the conservative party is the highest set of people in that party.
- Out of 256 people who gave a score of 2, 116 people have voted for the labour party. 140 people have voted for the conservative party. This is the instance where the conservative party has got more votes than the labour party.
- Out of 37 people who gave a score of 1, 16 people have voted for the labour party. 21 people have voted for the conservative party.
- The score of 3, 4 and 5 have more votes in the labour party.
- The score of 1 and 2 have more votes in the conservative party.

Vote & economic.cond.household



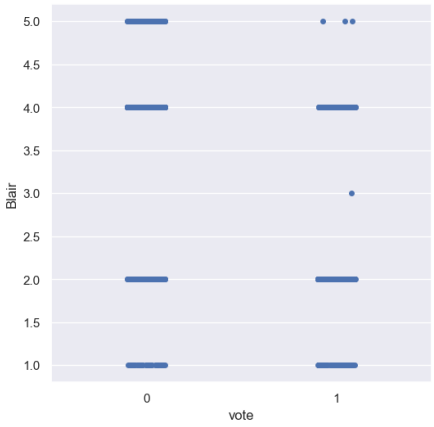

```
vote  economic.cond.household
0     1                          37
      2                         154
      3                         448
      4                         349
      5                          69
1     1                          28
      2                         126
      3                         197
      4                          86
      5                          23
dtype: int64
```

Figure 13: stripplot between Vote and economic.cond.household

Observation

- Labour party has higher votes overall.
- Out of 92 people who gave a score of 5, 69 people have voted for the labour party.
- Out of 435 people who gave a score of 4, 349 people have voted for the labour party. This is the 2nd highest set of people in the labour party.
- Out of 645 people who gave a score of 3, 448 people have voted for the labour party. This is the highest set of people in the labour party. The remaining 197 people who have voted for the conservative party is the highest set of people in that party.
- Out of 280 people who gave a score of 2, 154 people have voted for the labour party. 126 people have voted for the conservative party.
- Out of 65 people who gave a score of 1, 37 people have voted for the labour party. 28 people have voted for the conservative party.
- In all the instances, the labour party have more votes than the conservative party.

Vote and Blair



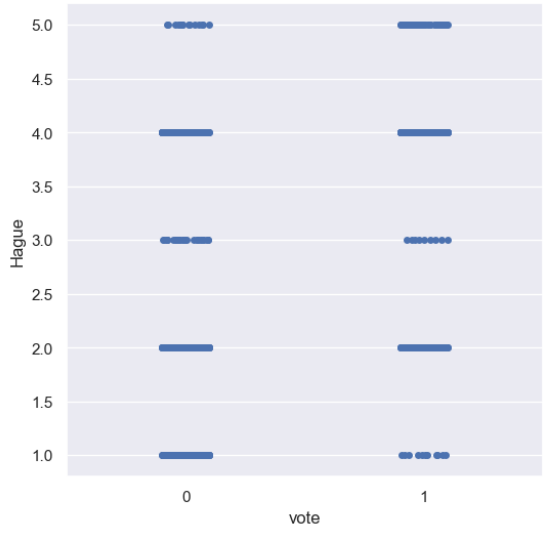

```
vote  Blair
0     1        38
      2       194
      4       676
      5       149
1     1        59
      2       240
      3         1
      4       157
      5         3
dtype: int64
```

Figure 14: stripplot between Vote and Blair

Observation

- Labour party has higher votes overall.
- Out of 152 people who gave a score of 5, 149 people have voted for the labour party. The remaining 3 people, despite giving a score of 5 to the labour leader, have chosen to vote for the conservative party.
- Out of 833 people who gave a score of 4, 676 people have voted for the labour party. The remaining 157 people, despite giving a score of 4 to the labour leader, have chosen to vote for the conservative party.
- Only 1 person has given a score of 3 and that person has voted for the conservative party.
- Out of 434 people who gave a score of 2, 240 people have voted for the conservative party. The remaining 194 people, despite giving an unsatisfactory score of 2 to the labour leader, have chosen to vote for the labour party.
- Out of 97 people who gave a score of 1, 59 people have voted for the conservative party. The remaining 38 people, despite giving the lowest score of 1 to the labour leader, have chosen to vote for the labour party.
- The score of 4 and 5 have more votes in the labour party.
- The score of 1, 2 and 3 have more votes in the conservative party.

Vote and Hague




```
vote  Hague
0     1        222
      2        522
      3         28
      4        271
      5         14
1     1         11
      2         95
      3          9
      4        286
      5         59
dtype: int64
```

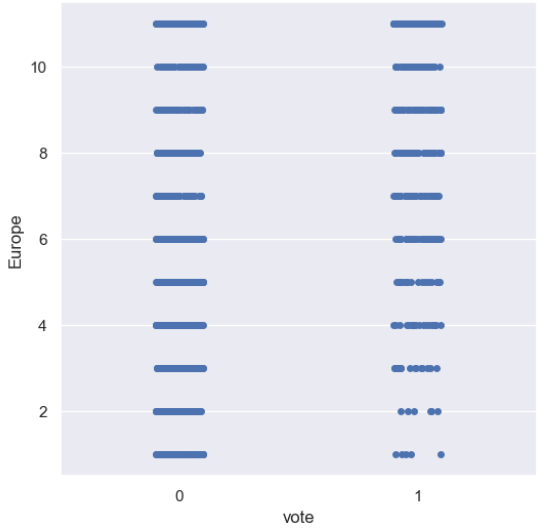
Figure 15: stripplot between Vote and Hague

Observations:

- Labour party has higher votes overall.
- Out of 73 people who gave a score of 5, 59 people have voted for the conservative party. The remaining 14 people, despite giving a score of 5 to the conservative leader, have chosen to vote for the labour party.
- Out of 557 people who gave a score of 4, 286 people have voted for the conservative party. The remaining 271 Page | 31 people, despite giving a score of 4 to the conservative leader, have chosen to vote for the labour party.

- Out of 37 people who gave a score of 3, 28 have voted for the labour party. The remaining 9, despite giving an average score of 3 to the conservative party, have chosen to vote for the conservative party.
- Out of 617 people who gave a score of 2, 522 people have voted for the labour party. The remaining 95 people, despite giving an unsatisfactory score of 2 to the conservative leader, have chosen to vote for the conservative party.
- Out of 233 people who gave a score of 1, 222 people have voted for the labour party. The remaining 11 people, despite giving the lowest score of 1 to the conservative leader, have chosen to vote for the conservative party.
- The score of 4 and 5 have more votes in the conservative party, although in 4, the votes are almost equal in both the parties. Conservative party gets slightly higher.
- The score of 1, 2 and 3 have more votes in the labour party. Still, a significant percentage of people who gave a bad score to the conservative leader still chose to vote for 'Hague'.

Vote & Europe



vote	Europe	
0	1	104
	2	71
	3	114
	4	108
	5	103
	6	172
	7	54
	8	63
	9	55
	10	47
	11	166
1	1	5
	2	6
	3	14
	4	18
	5	20
	6	35
	7	32
	8	48
	9	56
	10	54
	11	172

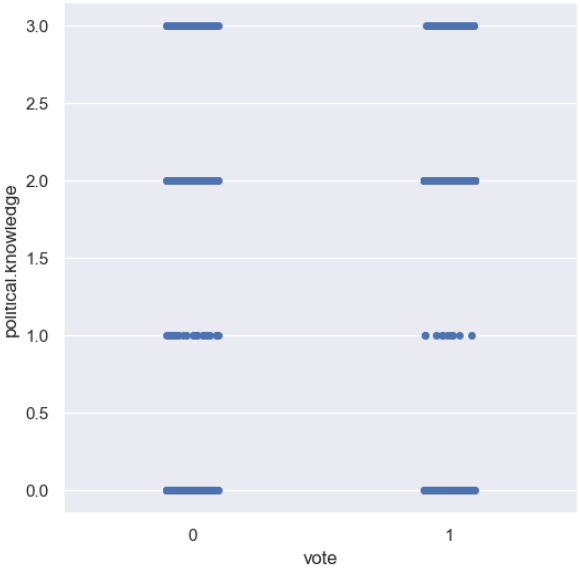
dtype: int64

Figure 16: stripplot between Vote and Europe

Observation:

- Out of 338 people who gave a score of 11, 166 people have voted for the labour party and 172 people have voted for the conservative party.
- People who gave score of 7 to 10 have voted for labour and conservative equally. Conservative party is slightly higher in these instances.
- Out of 207 people who gave a score of 6, 172 people have voted for the labour party and 35 people have voted for the conservative party.
- People who gave a score of 1 to 6 have voted for the labour party. As we can see, there are a total of 770 people who have given scores from 1 to 6. Out of 770 people, 672 people have voted for the labour party. So, 87.28% of the people have chosen labour party.
- So, we can infer that lower the 'Eurosceptic' sentiment, higher the votes for labour party.

Vote and Political.knowledge




```
vote  political.knowledge
0     0                    360
      1                     27
      2                    493
      3                    177
1     0                     94
      1                     11
      2                    283
      3                     72
dtype: int64
```

Figure 17: stripplot between Vote and political.knowledge

Observation:

- Out of 249 people who gave a score of 3, 177 people have voted for the labour party and 72 people have voted for the conservative party.
- Out of 776 people who gave a score of 2, 493 people have voted for the labour party and 283 people have voted for the conservative party.
- Out of 38 people who gave a score of 1, 27 people have voted for the labour party and 11 people have voted for the conservative party.
- Out of 454 people who gave a score of 0, 360 people have voted for the labour party and 94 people have voted for the conservative party.
- We can see that, in all instances, labour party gets the higher number of votes.
- Out of 1517 people, 454 people gave a score of 0. So, this means that, 29.93% of the people are casting their votes without any political knowledge.

Pair Plot

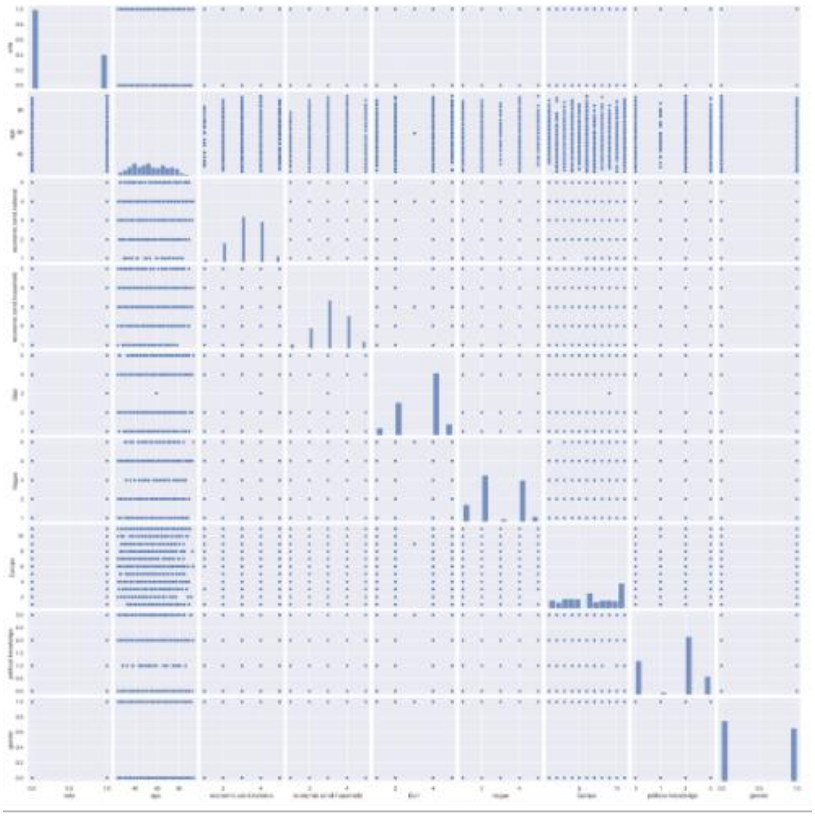


Figure 18: Pair plot of the numerical variables

Observations:

- Pair plot is a combination of histograms and scatter plots.
- From the histogram, we can see that, the 'Blair', 'Europe' and 'political.knowledge' variables are slightly left skewed.
- All other variables are normally distributed.
- From the scatter plots, we can see that, there is mostly no correlation between the variables.

Heat Map

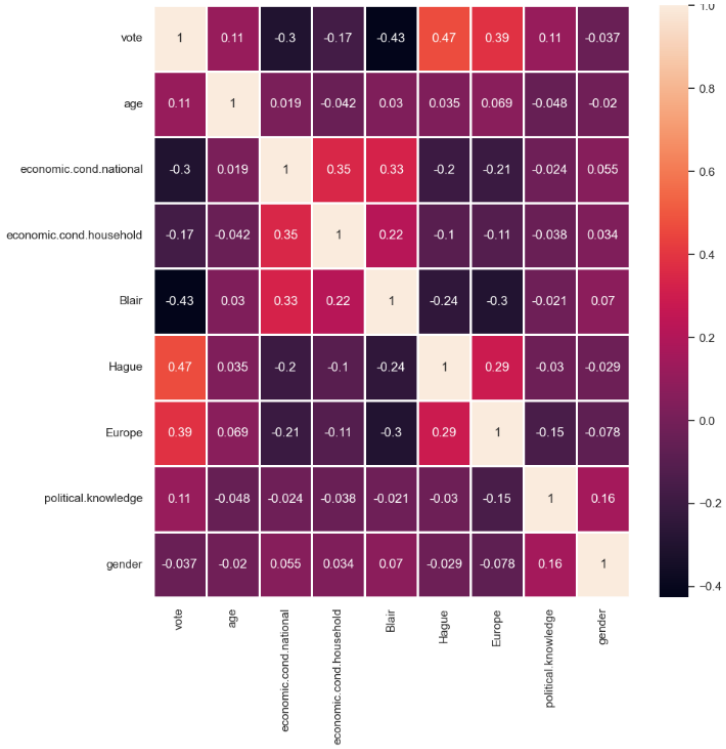


Figure 19: Heat Map of the numerical variables

Observations:

- We can see that, mostly there is no correlation in the dataset through this matrix. There are some variables that are moderately positively correlated and some that are slightly negatively correlated.
- ‘economic.cond.national’ with ‘economic.cond.household’ have moderate positive correlation.
- ‘Blair' with 'economic.cond.national' and ‘economic.cond.household’ have moderate positive correlation.
- ‘Europe’ with ‘Hague’ have moderate positive correlation.
- 'Hague' with 'economic.cond.national' and 'Blair' have moderate negative correlation.
- 'Europe' with 'economic.cond.national' and 'Blair' have moderate negative correlation

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 after Encoding:

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

Encoded Data Info:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1517 entries, 1 to 1525
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   vote                    1517 non-null   int64
 1   age                     1517 non-null   int64
 2   economic.cond.national  1517 non-null   int64
 3   economic.cond.household 1517 non-null   int64
 4   Blair                   1517 non-null   int64
 5   Hague                   1517 non-null   int64
 6   Europe                  1517 non-null   int64
 7   political.knowledge     1517 non-null   int64
 8   gender                  1517 non-null   int64
dtypes: int64(9)
```

Train -Test Split

- We will use vote as target variable.
- Training Set: 70percent of data.
- Testing Set: 30 percent of the data

Train – Test split shape

- x_train has 1061 rows and 8 columns in our Data set.
- y_train has 1061 rows and 1 column in our Data set.
- x_test has 456 rows and 8 columns in our Data set.
- y_test has 456 rows and 1 column in our Data set.

Scaling

- The dataset contains features highly varying in magnitudes, units and range between the 'age' column and other columns.
- But since, most of the machine learning algorithms use Eucledian distance between two data points in their computations, this is a problem.
- If left alone, these algorithms only take in the magnitude of features neglecting the units.

- The results would vary between different units, 1km and 1000 metres.
- The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes.
- To supress this effect, we need to bring all features to the same level of magnitudes. This can be achieved by scaling.
- In this case, we have a lot of encoded, ordinal, categorical and continuous variables. So, we use the minmaxscaler technique to scale the data.

Data After scaling

	0	1	2	3	4	5	6	7	8
0	0.0	0.275362	0.50	0.50	0.75	0.00	0.1	0.666667	0.0
1	0.0	0.173913	0.75	0.75	0.75	0.75	0.4	0.666667	1.0
2	0.0	0.159420	0.75	0.75	1.00	0.25	0.2	0.666667	1.0
3	0.0	0.000000	0.75	0.25	0.25	0.00	0.3	0.000000	0.0
4	0.0	0.246377	0.25	0.25	0.00	0.00	0.5	0.666667	1.0

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

Logistic Regression:

There are no outliers present in the continuous variable 'age'. The remaining variables are categorical in nature. Our model will use all the variables and 'vote' is the target variable.

Training Data:

Accuracy:

```
0.8312912346842601
```

Classification report Train Data

```
              precision    recall  f1-score   support

           0       0.86      0.91      0.88       754
           1       0.74      0.64      0.69       307

    accuracy                           0.83      1061
   macro avg       0.80      0.77      0.79      1061
weighted avg       0.83      0.83      0.83      1061
```

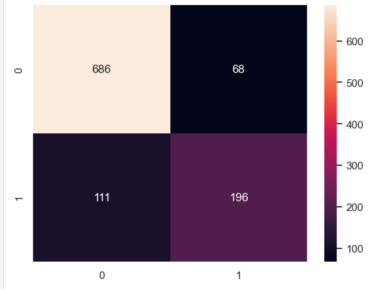


Figure 20: Confusion Matrix of Logistic Regression on Training Data

Mean Squared Error on Training

```
0.16870876531573986
```

Mean Squared Error on Testing

```
0.16447368421052633
```

Testing Data:

```
Accuracy on Testing 0.8355263157894737
```

Classification Report on Test Data:

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

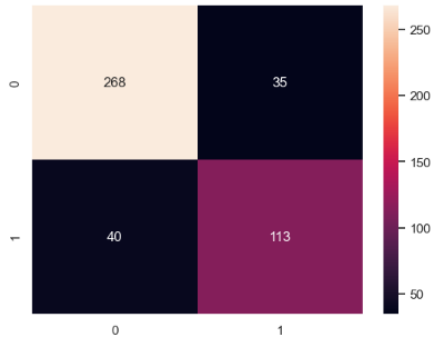


Figure 21: Confusion Matrix of Logistic Regression on Testing Data

Validate:

The error in the test data is slightly higher than the train data, which is fine because the error margin is low and the error in both train and test data is not too high. Thus, the model is not over-fitted or under-fitted.

Linear Discriminant Analysis Model:

Training Data:

Accuracy:

```
Accuracy on Training 0.8341187558906692
```

Classification Report

	precision	recall	f1-score	support
0	0.91	0.86	0.89	792
1	0.65	0.74	0.69	269
accuracy			0.83	1061
macro avg	0.78	0.80	0.79	1061
weighted avg	0.84	0.83	0.84	1061

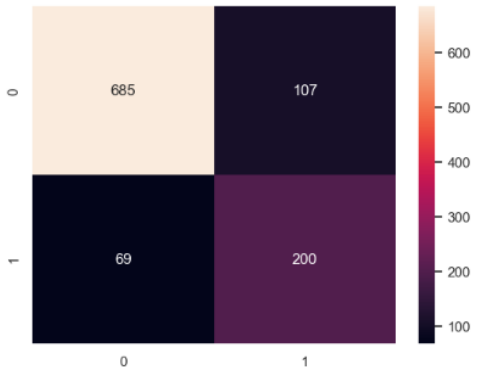


Figure 22: Confusion Matrix of Linear Discriminant Regression on Training Data

Testing Data:

Accuracy:

```
Accuracy on Testing 0.8333333333333334
```

Classification Report

	precision	recall	f1-score	support
0	0.89	0.86	0.88	311
1	0.73	0.77	0.74	145
accuracy			0.83	456
macro avg	0.81	0.82	0.81	456
weighted avg	0.84	0.83	0.83	456

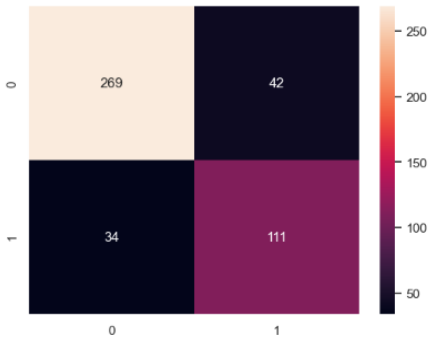


Figure 23: Confusion Matrix of Linear Discriminant Regression on Testing Data

Validate:

The error in the test data is slightly higher than the train data, which is absolutely fine because the error margin is low and the error in both train and test data is not too high. Thus, the model is not over-fitted or under-fitted.

Observation between Logistic Regression and Linear Regression:

- There is not much difference between the performance of regular LDA model and tuned LDA model.
- The values are high overall and there is no over-fitting or under-fitting. Therefore, both models are equally good models.

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

KNN Model:

Training Data

Confusion Matrix

```
[[699  98]
 [ 55 209]]
```

Accuracy:

```
Accuracy on Training 0.8557964184731386
```

Classification Report

	precision	recall	f1-score	support
0	0.93	0.88	0.90	797
1	0.68	0.79	0.73	264
accuracy			0.86	1061
macro avg	0.80	0.83	0.82	1061
weighted avg	0.87	0.86	0.86	1061

Testing Data

Confusion Matrix:

```
[[275  52]
 [ 28 101]]
```

Accuracy:

```
Accuracy on Testing 0.8245614035087719
```

Classification Report

	precision	recall	f1-score	support
0	0.91	0.84	0.87	327
1	0.66	0.78	0.72	129
accuracy			0.82	456
macro avg	0.78	0.81	0.79	456
weighted avg	0.84	0.82	0.83	456

Naïve Bayes

Training Data:

Confusion Matrix:

```
[[675  96]
 [ 79 211]]
```

Accuracy:

```
Accuracy on Training 0.8350612629594723
```

Classification Report:

	precision	recall	f1-score	support
0	0.90	0.88	0.89	771
1	0.69	0.73	0.71	290
accuracy			0.84	1061
macro avg	0.79	0.80	0.80	1061
weighted avg	0.84	0.84	0.84	1061

Testing Data:

Confusion Matrix:

```
[[263  41]
 [ 40 112]]
```

Accuracy:

```
Accuracy on Testing 0.8223684210526315
```

Classification Report

	precision	recall	f1-score	support
0	0.87	0.87	0.87	304
1	0.73	0.74	0.73	152
accuracy			0.82	456
macro avg	0.80	0.80	0.80	456
weighted avg	0.82	0.82	0.82	456

Observation between KNN and Naïve Bayes

- KNN having higher accuracy than Naïve bayes.
- KNN model is better when compared to Naïve Bayes.

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

Bagging (Random Forest After Tuning):
Training
Confusion Matrix:

```
[[708  46]
 [105 202]]
```

Accuracy:

```
0.8576814326107446
```

Classification Matrix:

	precision	recall	f1-score	support
0	0.87	0.94	0.90	754
1	0.81	0.66	0.73	307
accuracy			0.86	1061
macro avg	0.84	0.80	0.82	1061
weighted avg	0.85	0.86	0.85	1061

Testing
Confusion Matrix:

```
[[280  23]
 [ 60  93]]
```

Accuracy:

```
0.8179824561403509
```

Classification Matrix:

	precision	recall	f1-score	support
0	0.82	0.92	0.87	303
1	0.80	0.61	0.69	153
accuracy			0.82	456
macro avg	0.81	0.77	0.78	456
weighted avg	0.82	0.82	0.81	456

Bagging (Decision Tree):
Training
Confusion Matrix

```
[[754   0]
 [  0 307]]
```

Accuracy:

```
1.0
```

Classification Matrix:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	754
1	1.00	1.00	1.00	307
accuracy			1.00	1061
macro avg	1.00	1.00	1.00	1061
weighted avg	1.00	1.00	1.00	1061

Testing:
Confusion Matrix:

```
[[266  37]
 [ 45 108]]
```

Accuracy

```
0.8201754385964912
```

Classification Matrix:

	precision	recall	f1-score	support
0	0.86	0.88	0.87	303
1	0.74	0.71	0.72	153
accuracy			0.82	456
macro avg	0.80	0.79	0.80	456
weighted avg	0.82	0.82	0.82	456

Boosting (Ada Boost after Tuning)

Training:
Confusion Matrix:

```
[[702  52]
 [121 186]]
```

Accuracy

```
Accuracy on Training 0.8369462770970783
```

Classification Report

	precision	recall	f1-score	support
0	0.85	0.93	0.89	754
1	0.78	0.61	0.68	307
accuracy			0.84	1061
macro avg	0.82	0.77	0.79	1061
weighted avg	0.83	0.84	0.83	1061

Testing
Confusion Matrix:

```
[[271  32]
 [ 55  98]]
```

Accuracy:

```
0.8092105263157895
```

Classification Report:

	precision	recall	f1-score	support
0	0.83	0.89	0.86	303
1	0.75	0.64	0.69	153
accuracy			0.81	456
macro avg	0.79	0.77	0.78	456
weighted avg	0.81	0.81	0.80	456

Boosting (Gradient Boost After tuning)
Training:
Confusion Matrix:

```
[[708  46]
 [ 68 239]]
```

Accuracy

```
0.8925541941564562
```

Classification Report:

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

Testing
Confusion Matrix:

```
[[276  27]
 [ 48 105]]
```

Accuracy:

```
0.8355263157894737
```

Classification Report:

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

Observation:

- The gradient boost classifier after tuning, has improved the model significantly.
- The difference between the train and test accuracies has also been reduced.

- Overall, the tuned Gradient Boost classifier is a better 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
Logistic Regression:

Logistic Regression Model:

There are no outliers present in the continuous variable 'age'. The remaining variables are categorical in nature. Our model will use all the variables and 'vote' is the target variable.

Training Data:

Accuracy:

```
0.8312912346842601
```

Classification report Train Data

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

Confusion Matrix

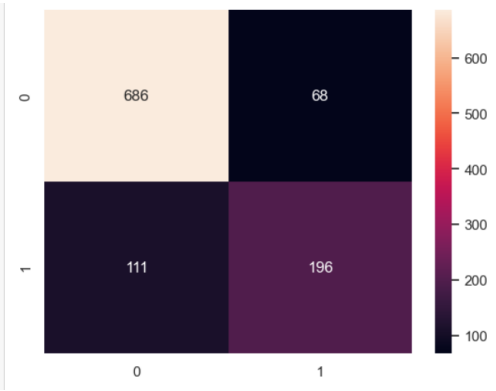


Figure 24: Confusion Matrix of Logistic Regression on Training Data

ROC Curve

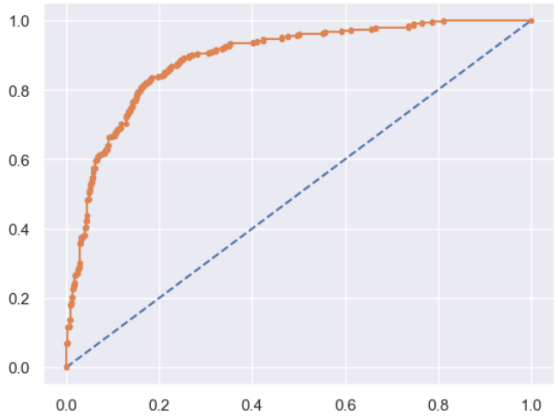


Figure 25: ROC Curve of Logistic Regression on Training Data

AUC Score

```
AUC: 0.890
```

Testing Data:

```
Accuracy on Testing 0.8355263157894737
```

Classification Report on Test Data:

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

Confusion Matrix

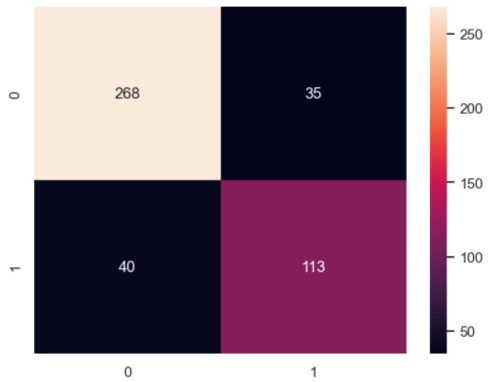


Figure 26: Confusion Matrix of Logistic Regression on Testing Data

ROC Curve

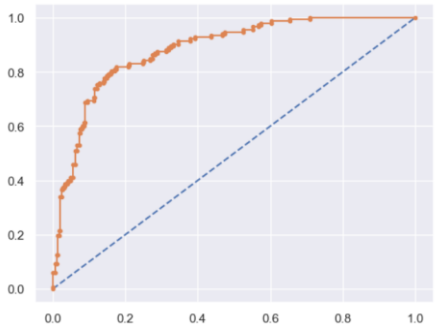


Figure 27: ROC Curve of Logistic Regression on Testing Data

AUC Score:

```
AUC: 0.883
```

Linear Discriminant Analysis Model:

Training Data:

Accuracy:

```
Accuracy on Training 0.8341187558906692
```

Classification Report

	precision	recall	f1-score	support
0	0.91	0.86	0.89	792
1	0.65	0.74	0.69	269
accuracy			0.83	1061
macro avg	0.78	0.80	0.79	1061
weighted avg	0.84	0.83	0.84	1061

Confusion Matrix

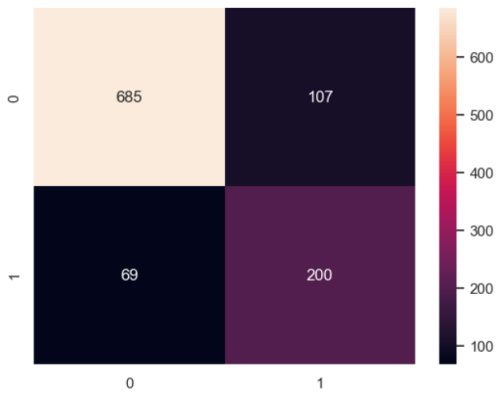


Figure 28: Confusion Matrix of Linear Discriminant Regression on Training Data

ROC Curve:

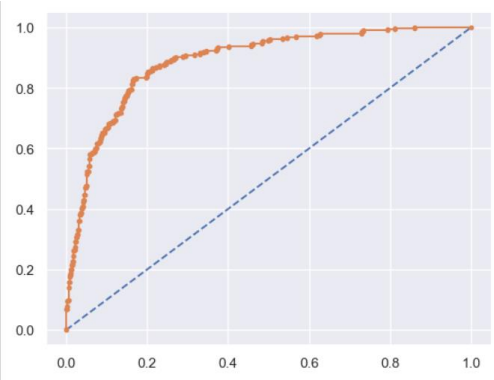


Figure 29: ROC Curve of Linear Discriminant Regression on Training Data

AUC :

```
AUC: 0.889
```

Testing Data:

Accuracy:

```
Accuracy on Testing 0.8333333333333334
```

Classification Report

	precision	recall	f1-score	support
0	0.89	0.86	0.88	311
1	0.73	0.77	0.74	145
accuracy			0.83	456
macro avg	0.81	0.82	0.81	456
weighted avg	0.84	0.83	0.83	456

Confusion Matrix:

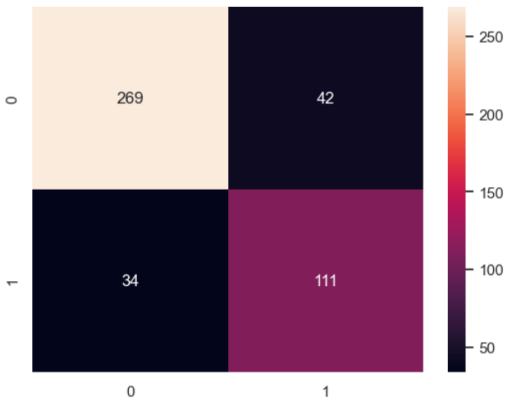


Figure 30: Confusion Matrix of Linear Discriminant Regression on Testing Data

AUC

```
AUC: 0.888
```

ROC Curve

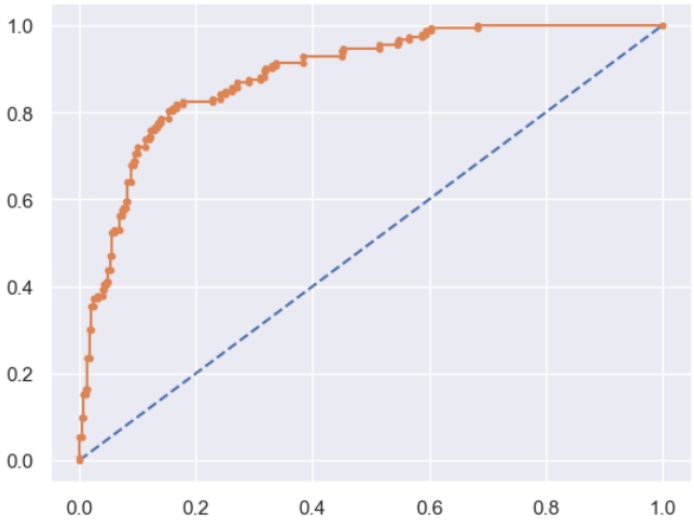


Figure 31: ROC Curve of Linear Discriminant Regression on Training Data

KNN Model:

Training Data

Confusion Matrix

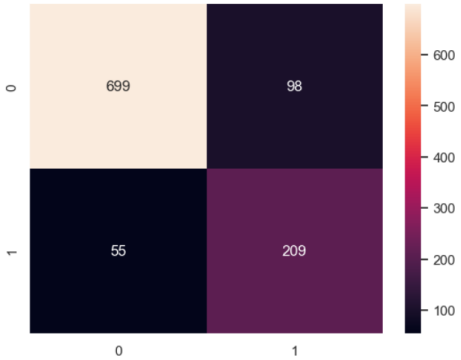


Figure 32: Confusion Matrix of KNN on Training Data

Accuracy:

```
Accuracy on Training 0.8557964184731386
```

Classification Report

	precision	recall	f1-score	support
0	0.93	0.88	0.90	797
1	0.68	0.79	0.73	264
accuracy			0.86	1061
macro avg	0.80	0.83	0.82	1061
weighted avg	0.87	0.86	0.86	1061

ROC Curve:

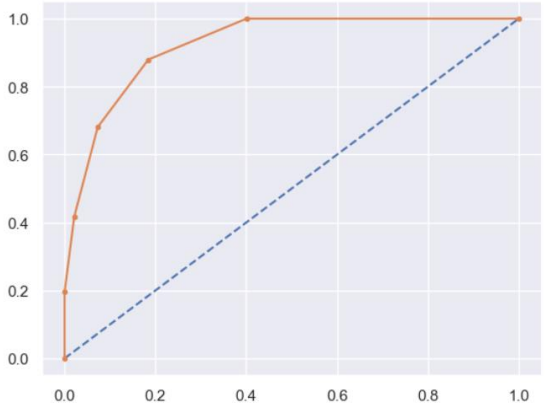


Figure 33: ROC Curve of KNN Regression on Training Data

AUC Score

AUC: 0.924

Testing Data

Confusion Matrix:

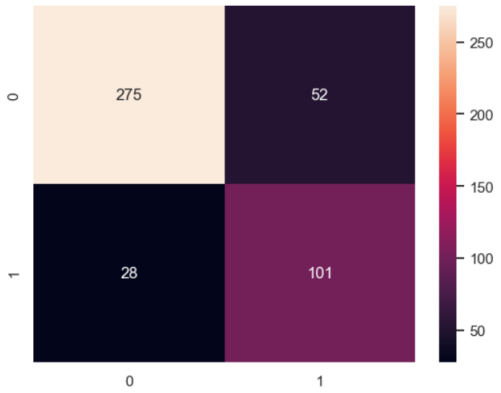


Figure 34: Confusion Matrix of KNN on Testing Data

Accuracy:

```
Accuracy on Testing 0.8245614035087719
```

Classification Report

	precision	recall	f1-score	support
0	0.91	0.84	0.87	327
1	0.66	0.78	0.72	129
accuracy			0.82	456
macro avg	0.78	0.81	0.79	456
weighted avg	0.84	0.82	0.83	456

ROC Curve

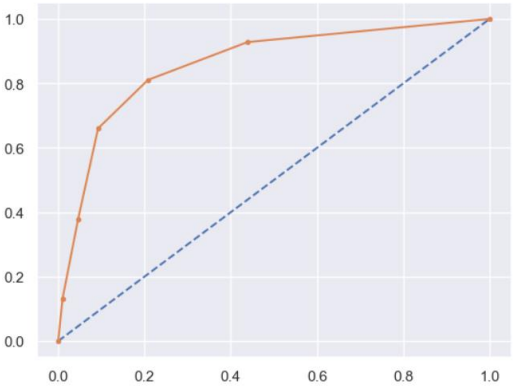


Figure 35: ROC Curve of KNN Regression on Testing Data

AUC Score

```
AUC: 0.861
```

Naïve Bayes

Training Data:

Confusion Matrix:

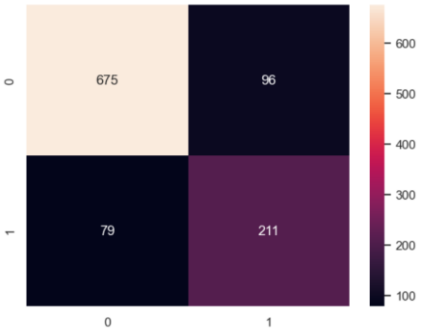


Figure 36: Confusion Matrix of Naïve Bayes on Training Data

Accuracy:

```
Accuracy on Training 0.8350612629594723
```

Classification Report:

	precision	recall	f1-score	support
0	0.90	0.88	0.89	771
1	0.69	0.73	0.71	290
accuracy			0.84	1061
macro avg	0.79	0.80	0.80	1061
weighted avg	0.84	0.84	0.84	1061

ROC Curve

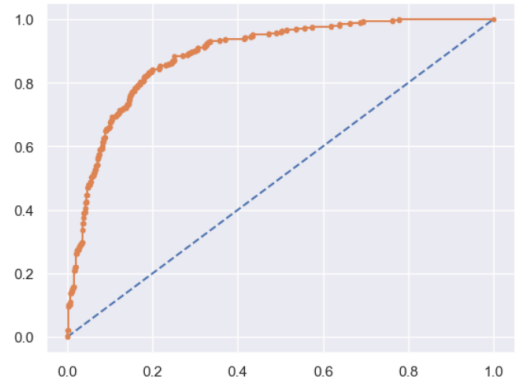


Figure 37: ROC Curve of Naïve Bayes on Training Data

AUC Score:

```
AUC: 0.888
```

Testing Data:

Confusion Matrix:

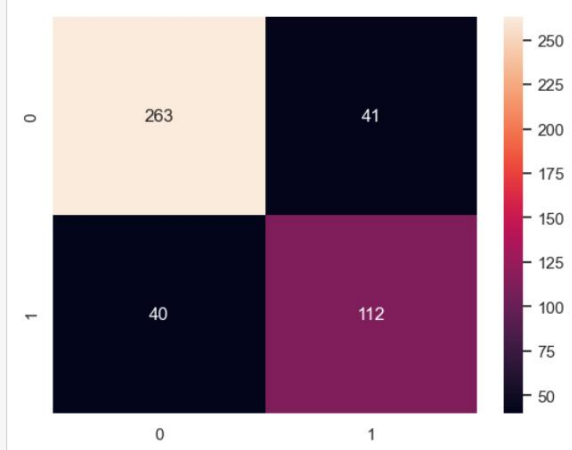


Figure 38: Confusion Matrix of Naïve Bayes on Testing Data

Accuracy:

```
Accuracy on Testing 0.8223684210526315
```

Classification Report

	precision	recall	f1-score	support
0	0.87	0.87	0.87	304
1	0.73	0.74	0.73	152
accuracy			0.82	456
macro avg	0.80	0.80	0.80	456
weighted avg	0.82	0.82	0.82	456

ROC Curve:

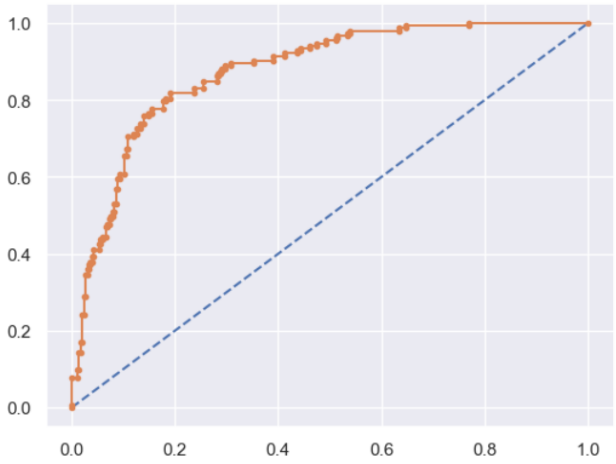


Figure 39: ROC Curve of Naïve Bayes on Testing Data

AUC Score

```
AUC: 0.876
```

Bagging (Random Forest After Tuning):

Training

Confusion Matrix:

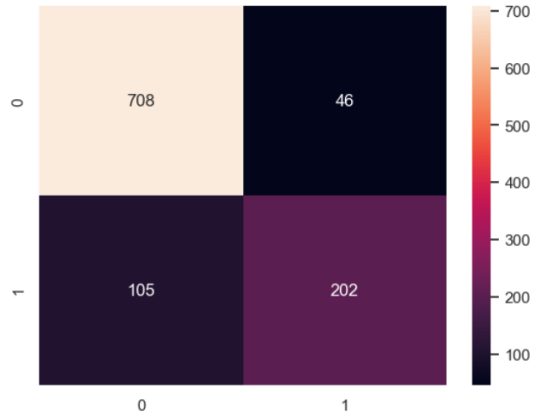


Figure 40: Confusion Matrix of Random Forest on Training Data

Accuracy:

```
0.8576814326107446
```

Classification Matrix:

	precision	recall	f1-score	support
0	0.87	0.94	0.90	754
1	0.81	0.66	0.73	307
accuracy			0.86	1061
macro avg	0.84	0.80	0.82	1061
weighted avg	0.85	0.86	0.85	1061

ROC Curve:

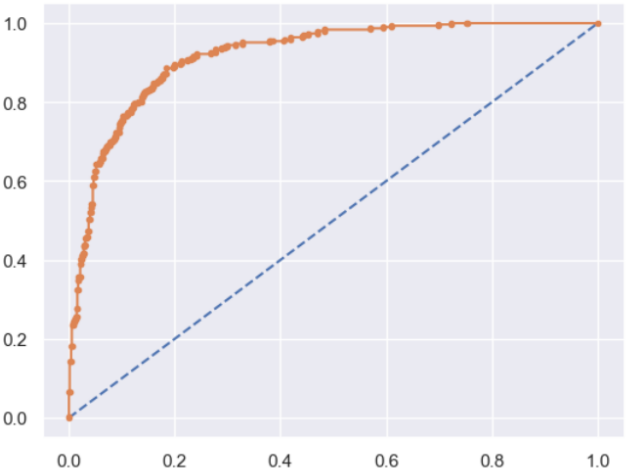


Figure 41: ROC Curve of Random Forest on Training Data

AUC Score

```
AUC: 0.918
```

Testing

Confusion Matrix:

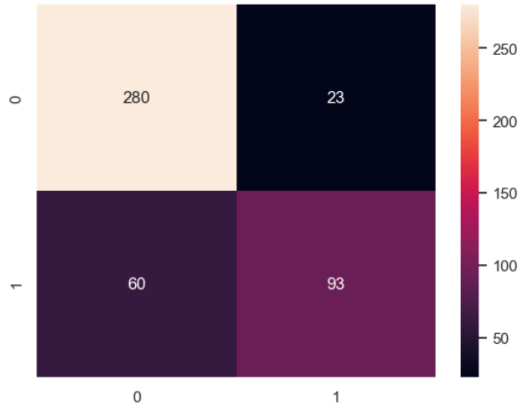


Figure 42: Confusion Matrix of Random Forest on Testing Data

Accuracy:

```
0.8179824561403509
```

Classification Matrix:

	precision	recall	f1-score	support
0	0.82	0.92	0.87	303
1	0.80	0.61	0.69	153
accuracy			0.82	456
macro avg	0.81	0.77	0.78	456
weighted avg	0.82	0.82	0.81	456

ROC Curve:

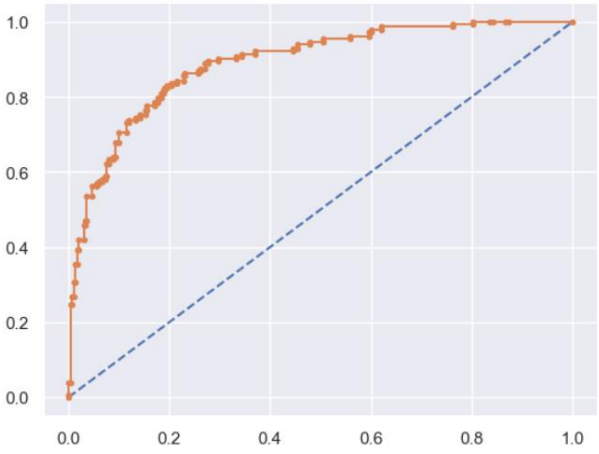


Figure 43: ROC Curve of Random Forest on Testing Data

AUC Score

```
AUC: 0.891
```

Bagging (Decision Tree):
Training
Confusion Matrix

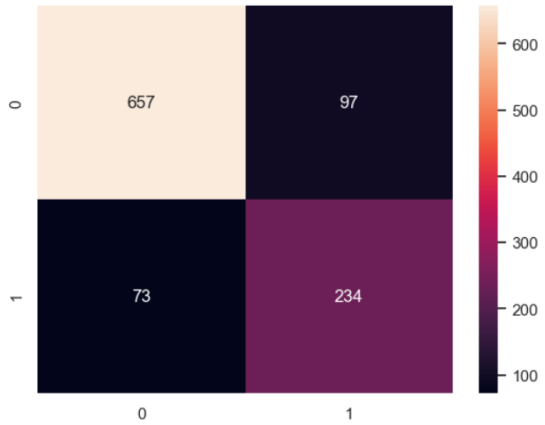


Figure 44: Confusion Matrix of Random Forest on Training Data

Accuracy:

```
0.8397737983034873
```

Classification Matrix:

	precision	recall	f1-score	support
0	0.90	0.87	0.89	754
1	0.71	0.76	0.73	307
accuracy			0.84	1061
macro avg	0.80	0.82	0.81	1061
weighted avg	0.84	0.84	0.84	1061

ROC Curve

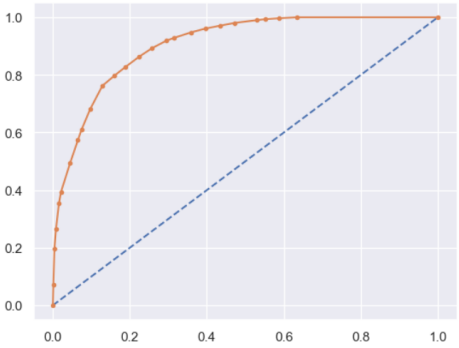


Figure 45: ROC Curve of Decision Tree on Training Data

AUC Score

```
AUC: 0.907
```

Testing:

Confusion Matrix:

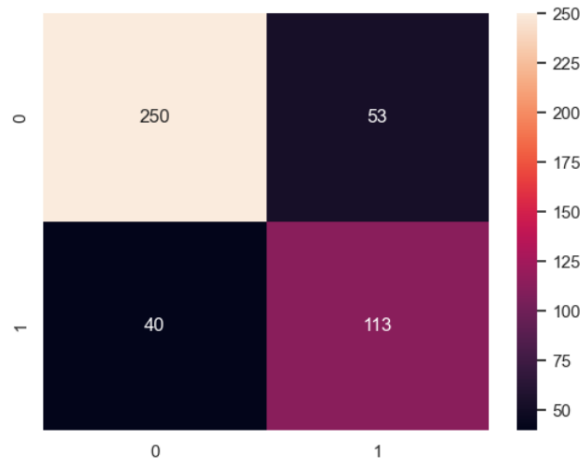


Figure 46: Confusion Matrix of Decision Tree on Testing Data

Accuracy

```
0.8201754385964912
```

Classification Matrix:

	precision	recall	f1-score	support
0	0.86	0.83	0.84	303
1	0.68	0.74	0.71	153
accuracy			0.80	456
macro avg	0.77	0.78	0.78	456
weighted avg	0.80	0.80	0.80	456

ROC Curve

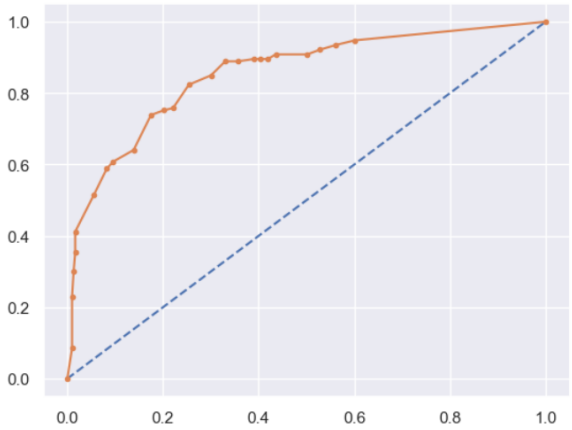


Figure 47: ROC Curve of Decision Tree on Testing Data

AUC Score

```
AUC: 0.856
```

Boosting (Ada Boost after Tuning)

Training:

Confusion Matrix:

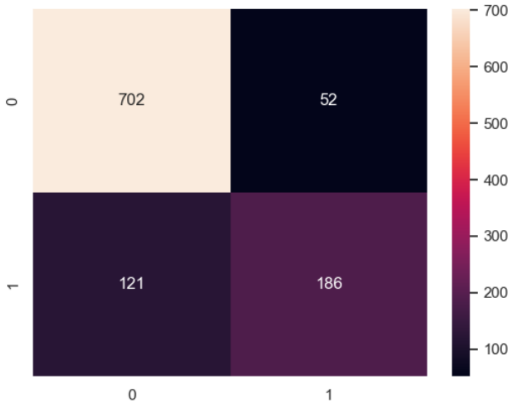


Figure 48: Confusion Matrix of Ada Boost on Training Data

Accuracy

```
Accuracy on Training 0.8369462770970783
```

Classification Report

	precision	recall	f1-score	support
0	0.85	0.93	0.89	754
1	0.78	0.61	0.68	307
accuracy			0.84	1061
macro avg	0.82	0.77	0.79	1061
weighted avg	0.83	0.84	0.83	1061

ROC Curve

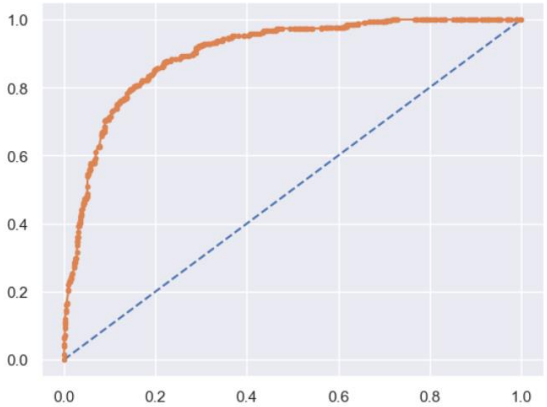


Figure 49: ROC Curve of Ada Boost on Training Data

AUC

```
AUC: 0.902
```

Testing

Confusion Matrix:

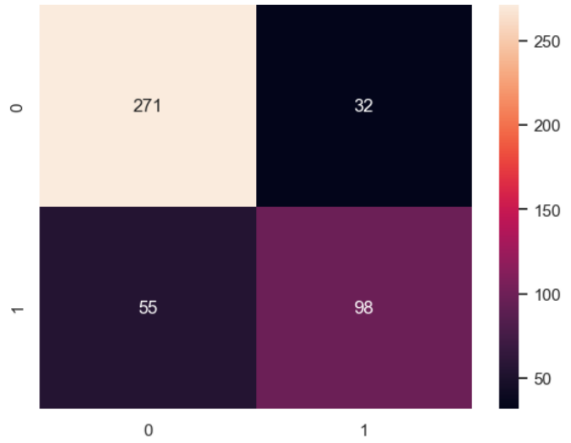


Figure 50: Confusion Matrix of Ada Boost on Testing Data

Accuracy:

```
0.8092105263157895
```

Classification Report:

	precision	recall	f1-score	support
0	0.83	0.89	0.86	303
1	0.75	0.64	0.69	153
accuracy			0.81	456
macro avg	0.79	0.77	0.78	456
weighted avg	0.81	0.81	0.80	456

ROC Curve

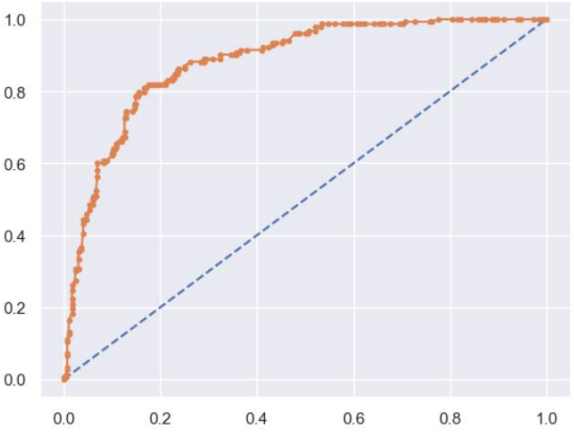


Figure 51: ROC Curve of Ada Boost on Testing Data

AUC

```
AUC: 0.884
```

Boosting (Gradient Boost After tuning)
Training:
Confusion Matrix:

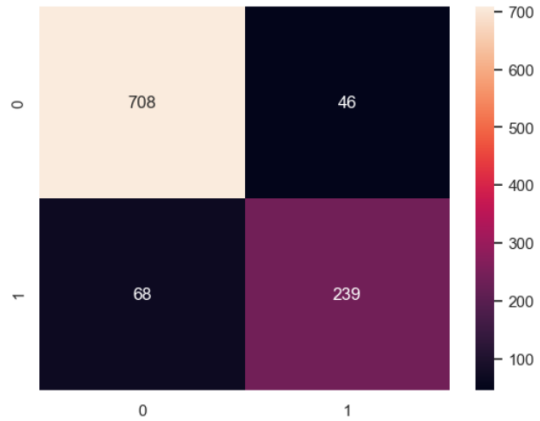


Figure 52: Confusion Matrix of Gradient Boost on Training Data

Accuracy

```
0.8925541941564562
```

Classification Report:

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

ROC Curve

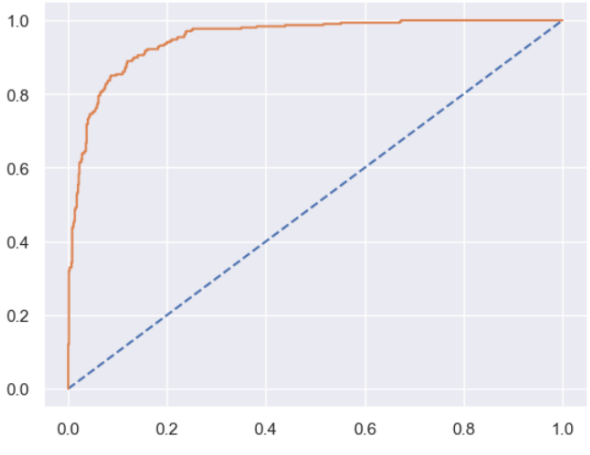


Figure 53: ROC Curve of Gradient Boost on Training Data

AUC

```
AUC: 0.951
```

Testing
Confusion Matrix:

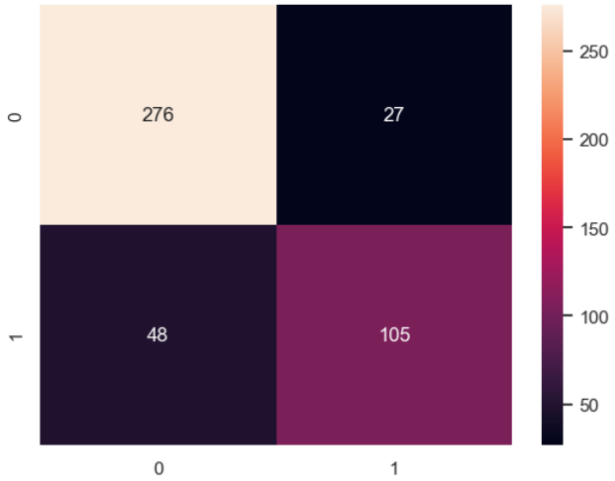


Figure 54: Confusion Matrix of Gradient Boost on Testing Data

Accuracy:

```
0.8355263157894737
```

Classification Report:

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

ROC Curve

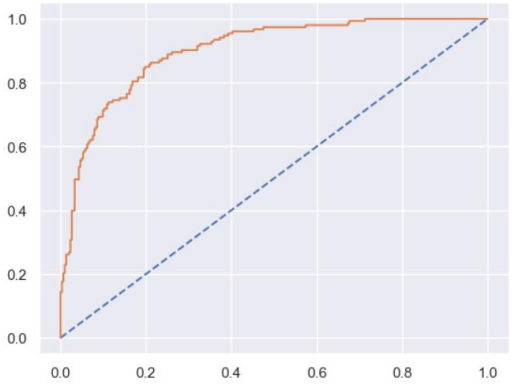


Figure 55: ROC Curve of Gradient Boost on Testing Data

AUC

```
AUC: 0.899
```

Conclusion

- There is no under-fitting or over-fitting in any of the tuned models.
- All the tuned models have high values, and every model is good. But as we can see, the most consistent tuned model in both train and test data is the Gradient Boost model.
- The tuned gradient boost model performs the best with 88.31% accuracy score in train and 87.28% accuracy score in test. Also, it has the best AUC score of 94% in both train and test data which is the highest of all the models.
- It also has a precision score of 88% and recall of 94% which is also the highest of all the models. So, we conclude that Gradient Boost Tuned model is the best/optimized model.

1.8 Based on these predictions, what are the insights?

Insights:

- Labour party has more than double the votes of conservative party. Labour party is performing better than Conservative from huge margin.
- Most number of people have given a score of 3 and 4 for the national economic condition and the average score is 3.245221.
- Most number of people have given a score of 3 and 4 for the household economic condition and the average score is 3.137772.
- Blair has higher number of votes than Hague and the scores are much better for Blair than for Hague.
- The average score of Blair is 3.335531 and the average score of Hague is 2.749506. So, here we can see that, Blair has a better score.
- On a scale of 0 to 3, about 30% of the total population has zero knowledge about politics/parties.
- People who gave a low score of 1 to a certain party, still decided to vote for the same party instead of voting for the other party. This can be because of lack of political knowledge among the people.
- People who have higher Eurosceptic sentiment, has voted for the conservative party and lower the Eurosceptic sentiment, higher the votes for Labour party.
- Out of 454 people who gave a score of 0 for political knowledge, 360 people have voted for the labour party and 94 people have voted for the conservative party.
- All models performed well on training data set as well as test data set. The tuned models have performed better than the regular models.
- There is no over-fitting in any model except Random Forest and Bagging regular models.

- Gradient Boosting model tuned is the best/optimized model.

Problem 2:

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

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

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

Number of characters

- President Franklin D. Roosevelt's speech have 7571 characters (including spaces).
- President John F. Kennedy's speech have 7618 characters (including spaces).
- President Richard Nixon's speech has 9991 characters (including spaces).

Number of words

- There are 1360 words in President Franklin D. Roosevelt's speech.
- There are 1390 words in President John F. Kennedy's speech.
- There are 1819 words in President Richard Nixon's speech.

Number of sentences

- There are 67 sentences in President Franklin D. Roosevelt's speech.
- There are 52 sentences in President John F. Kennedy'sspeech.
- There are 68 sentences in President Richard Nixon's speech.

2.2 **Remove all the stopwords from all three speeches.**

Before Removal

- There are 1360 words in President Franklin D. Roosevelt's speech.
- There are 1390 words in President John F. Kennedy's speech.
- There are 1819 words in President Richard Nixon's speech.

After removing Stopwords

- There are 632 words in President Franklin D. Roosevelt's speech.
- There are 696 words in President John F. Kennedy's speech.
- There are 848 words in President Richard Nixon's speech.

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 for Roosevelt:

```
FreqDist({'nation': 12, 'know': 10, 'spirit': 9, 'life': 9, 'democracy': 9, 'us': 8, 'people': 7, 'america': 7, 'years': 6, 'fr
eedom': 6, ...})
```

  - nation
  - know
  - sprit

- Top Three words for Kennedy:

```
FreqDist({'let': 16, 'us': 12, 'world': 8, 'sides': 8, 'new': 7, 'pledge': 7, 'citizens': 5, 'power': 5, 'shall': 5, 'free': 5,
...})
```

  - let
  - us
  - world

- Top Three words for Nixon:

```
FreqDist({'us': 26, 'let': 22, 'america': 21, 'peace': 19, 'world': 18, 'new': 15, "'s": 14, 'nation': 11, 'responsibility': 1
1, 'government': 10, ...})
```

  - us
  - let
  - America

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

Word cloud for Roosevelt

Figure 56: Word Cloud for Roosevelt (after cleaning)

Word cloud for Kennedy

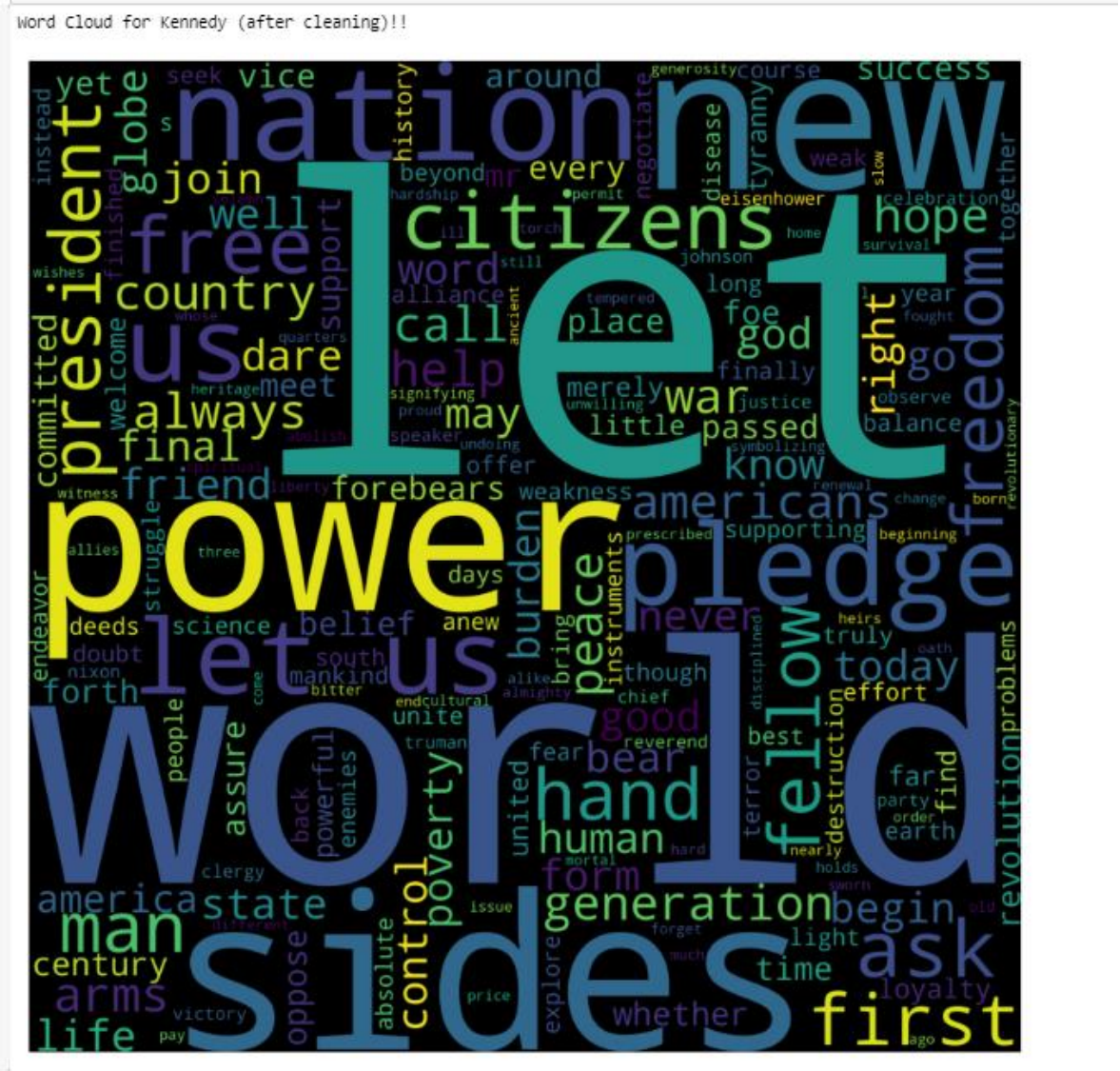


Figure 57: Word Cloud for Kennedy (after cleaning)

Word cloud for Nixon

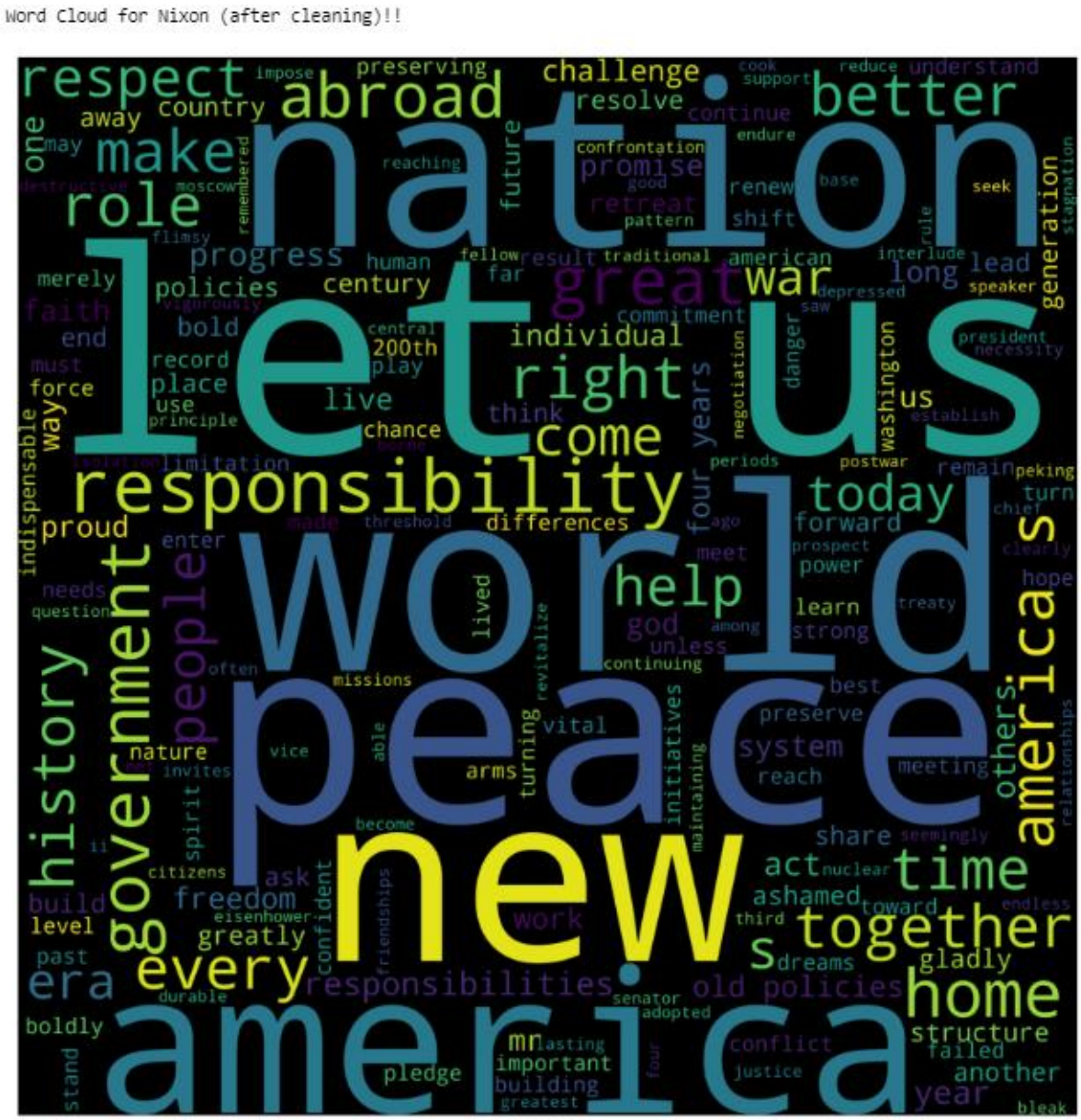


Figure 58: Word Cloud for Nixon (after cleaning)