SMDM Project: MACHINE LEARNING

Classification models, Ensemble techniques, Text analysis

Student's Name: THILAK RAJ | Batch: 18 FEB 2022

Contents

Executive Summary (**Problem 1)**: 6

Data Dictionary for Market Segmentation: 6

1.1 Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it. 6

Head of the Table: 6

Shape of the Dataframe: 7

Null values in the data: 7

DATA INFO of the dataframe: 7

The Descriptive statistics of the dataset: 8

Datatypes of the dataframe: 8

Value counts of object variables: 8

Checking duplicate rows in dataframe: 9

Observation: 9

1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers 10

Univariate Analysis: 10

Observation: 10

Age: 11

Ecomnomic.comd.national: 11

Economic.cond.household: 13

Blair: 14

Hague: 16

Europe: 17

Political.knowledge: 18

Gender: 20

Vote: 20

Bivariate and Multivariate Analysis: 21

Vote and Age: 21

Vote and Gender: 22

Vote and economic.cond.national: 23

vote and economic.cond.household: 24

Vote and Blair: 25

vote and Hague: 26

Vote and Europe: 27

Vote and political.knowledge 29

Checking pair-wise distribution of the continuous variables: 30

Correlation matrix: 32
Checking outliers: 33
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). 34
Scaling: 34
Train-test-split: 35
1.4 Apply Logistic Regression and LDA (linear discriminant analysis). 36
Logistic Regression Model: 36
Linear Discriminant Analysis Model (LDA): 38
1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results. 39
K-Nearest Neighbour Model: 39
Naive Bayes: 42
1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting. 43
Model Tuning: 44
Logistic Regression: 44
Linear Discriminant Analysis Model Tuning(LDA): 45
K-Nearest Neighbour Model Tuning: 46
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. 47
Comparison of the different models with respect to Training data: 47
Comparison of the different models with respect to Testing data: 48
1.8 Based on these predictions, what are the insights? 48
Problem 2: 50
2.1 Find the number of characters, words, and sentences for the mentioned documents. 50
Character count of the speeches: 50
Words count if the speeches: 50
Sentence counts of the speeches: 50
2.2 Remove all the stopwords from all three speeches. 50
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) 51
2.4 Plot the word cloud of each of the speeches of the variable. (after removing the stopwords) 51

Figure 1.Boxplot, Distplot and Histogram of 'Age' attribute: 11
Figure 2: Countplot of economic.cond.national 12
Figure 3: Boxplot,Distplot and Histogram of economic.comd.national 12

Figure 4:Countplot of economic.cond.household 13
Figure 5: Countplot of Blair 15
Figure 6: Boxplot, Distplot and Histogram of Blair 15
Figure 7: Countplot of Hague 16
Figure 8:Countplot of Europe 17
Figure 9:Boxplot,Distplot and histogram of Europe 18
Figure 10:Countplot of political.knowledge 19
Figure 11:Boxplot, Distplot and Histogram of political.knowledge 19
Figure 12:Countplot of gender 20
Figure 13: Boxplot of vote and age 21
Figure 14: Stripplot of vote and age 22
Figure 15: Stripplot of Vote and economic.cond.national 23
Figure 16:Stripplot of vote and economic.cond.household 24
Figure 17:Stripplot of Vote and Blair 25
Figure 18:Stripplot of Vote and Hague 26
Figure 19:Stripplot of vote and Europe 27
Figure 20: Stripplot of vote and plotical.knowledge 29
Figure 21: Pairplot 30
Figure 22:Pairplot wrt vote 31

Executive Summary (**Problem 1**):

You are hired by one of the leading news channels CNBE who wants to analyse 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.

Data Dictionary for Market Segmentation:

1. vote: Party choice: Conservative or Labour
2. age: in years
3. economic.cond.national: Assessment of current national economic conditions, 1 to 5.
4. economic.cond.household: Assessment of current household economic conditions, 1 to 5.
5. Blair: Assessment of the Labour leader, 1 to 5.
6. Hague: Assessment of the Conservative leader, 1 to 5.\
7. Europe: an 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Eurosceptic' sentiment.
8. political.knowledge: Knowledge of parties' positions on European integration, 0 to 3.
9. gender: female or male.

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

Head of the Table:

Below is the head of the data table:

	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
5	6	Labour	47	3	4	4	4	4	2	male
6	7	Labour	57	2	2	4	4	11	2	male
7	8	Labour	77	3	4	4	1	1	0	male
8	9	Labour	39	3	3	4	4	11	0	female
9	10	Labour	70	3	2	5	1	11	2	male

We dropped the unnamed column as the column is not in use.

The head of the table after dropping the unnamed column.

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

Shape of the Dataframe:

The shape of the dataframe is:

```
(1525, 9)
```

Null values in the data:

```
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
```

We found no null values in the data.

DATA INFO of the dataframe:

The information of the dataframe

```
<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
```

The Descriptive statistics of the dataset:

The descriptive statistics of the dataset.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
vote	1525	2	Labour	1063	NaN	NaN	NaN	NaN	NaN	NaN	NaN
age	1525.0	NaN	NaN	NaN	54.182295	15.711209	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1525.0	NaN	NaN	NaN	3.245902	0.880969	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1525.0	NaN	NaN	NaN	3.140328	0.929951	1.0	3.0	3.0	4.0	5.0
Blair	1525.0	NaN	NaN	NaN	3.334426	1.174824	1.0	2.0	4.0	4.0	5.0
Hague	1525.0	NaN	NaN	NaN	2.746885	1.230703	1.0	2.0	2.0	4.0	5.0
Europe	1525.0	NaN	NaN	NaN	6.728525	3.297538	1.0	4.0	6.0	10.0	11.0
political.knowledge	1525.0	NaN	NaN	NaN	1.542295	1.083315	0.0	0.0	2.0	2.0	3.0
gender	1525	2	female	812	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Datatypes of the dataframe:

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

We found two object datatype columns in the dataset. Remaining all the columns are of int datatype.

Value counts of object variables:

Value count of gender:

```
female    812
male      713
Name: gender, dtype: int64
```

Value count of vote:

```
Labour          1063
Conservative     462
Name: vote, dtype: int64
```

Checking duplicate rows in dataframe:

The list of duplicate rows in the datframe:

```
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

We are dropping the duplicate data in the dataframe.

Below is the shape of the dataframe before and after dropping the duplicate entries.

```
Before (1525, 9)
After (1517, 9)
```

Observation:

- We have dropped the 'unnamed' column from the datasetas it is not useful for our study.
- The dataset had 8 duplicated values. So, we are dropped them.
- The data set had 1525 rows and 9 columns. After dropping the duplicate values, there are 1517 rows and 9columns.
- It has 7 numerical data types and 2 categorical datatypes.
- There is no null value in any column.

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

Univariate Analysis:

Descriptive data analyses of the data

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
vote	1517	2	Labour	1057	NaN	NaN	NaN	NaN	NaN	NaN	NaN
age	1517.0	NaN	NaN	NaN	54.241266	15.701741	24.0	41.0	53.0	67.0	93.0
economic.cond.national	1517.0	NaN	NaN	NaN	3.245221	0.881792	1.0	3.0	3.0	4.0	5.0
economic.cond.household	1517.0	NaN	NaN	NaN	3.137772	0.931069	1.0	3.0	3.0	4.0	5.0
Blair	1517.0	NaN	NaN	NaN	3.335531	1.174772	1.0	2.0	4.0	4.0	5.0
Hague	1517.0	NaN	NaN	NaN	2.749506	1.232479	1.0	2.0	2.0	4.0	5.0
Europe	1517.0	NaN	NaN	NaN	6.740277	3.299043	1.0	4.0	6.0	10.0	11.0
political.knowledge	1517.0	NaN	NaN	NaN	1.540541	1.084417	0.0	0.0	2.0	2.0	3.0
gender	1517	2	female	808	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Descriptive analysis of 'age' column:

```
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
```

Observation:

- From the above data we noticed that there are two object columns and remaining are int64.
- The data range of age column is quite greater than other attributes so we might need standardization process to bring the range to ideal level.

Age:

Boxplot, Distplot and Histogram of age variable:

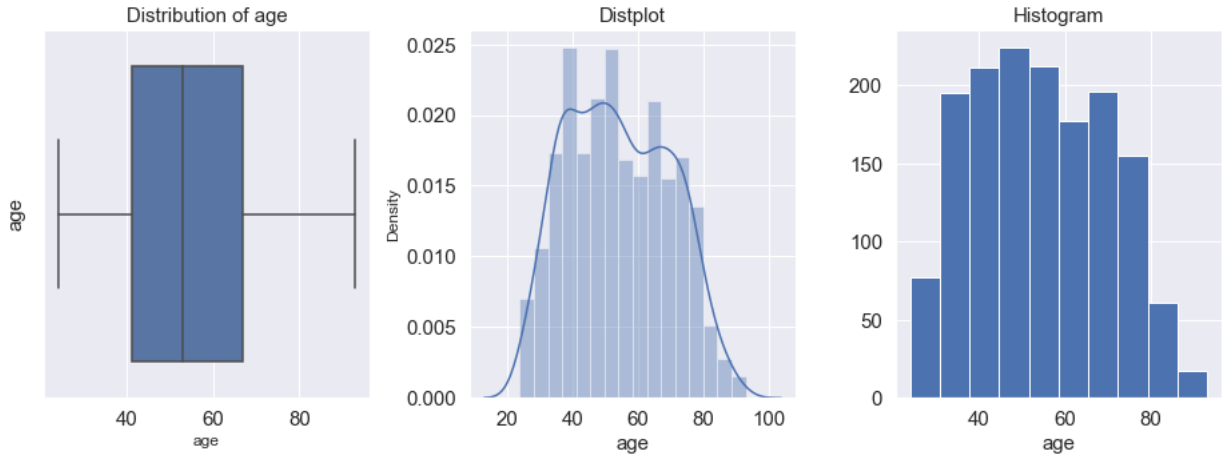


Figure 1.Boxplot, Distplot and Histogram of 'Age' attribute:

Observation:

- The data is normally distributed.
- Maximum number of people are aged between 40 and 70.
- Outliers are not present.
- The minimum value is 24 and the maximum value is 93.
- The mean value is 54.241266
-

Ecomnomic.comd.national:

Data description:

```
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
```

Value counts:

```
3    604
4    538
2    256
5     82
1     37
Name: economic.cond.national, dtype: int64
```

Count plot:

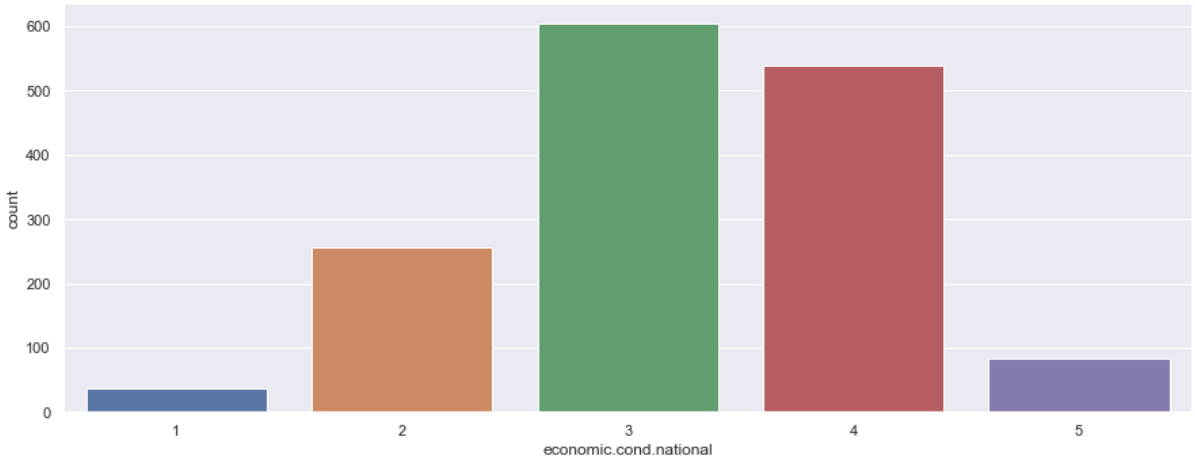


Figure 2: Countplot of economic.cond.national

Boxplot, Distplot and Histogram:

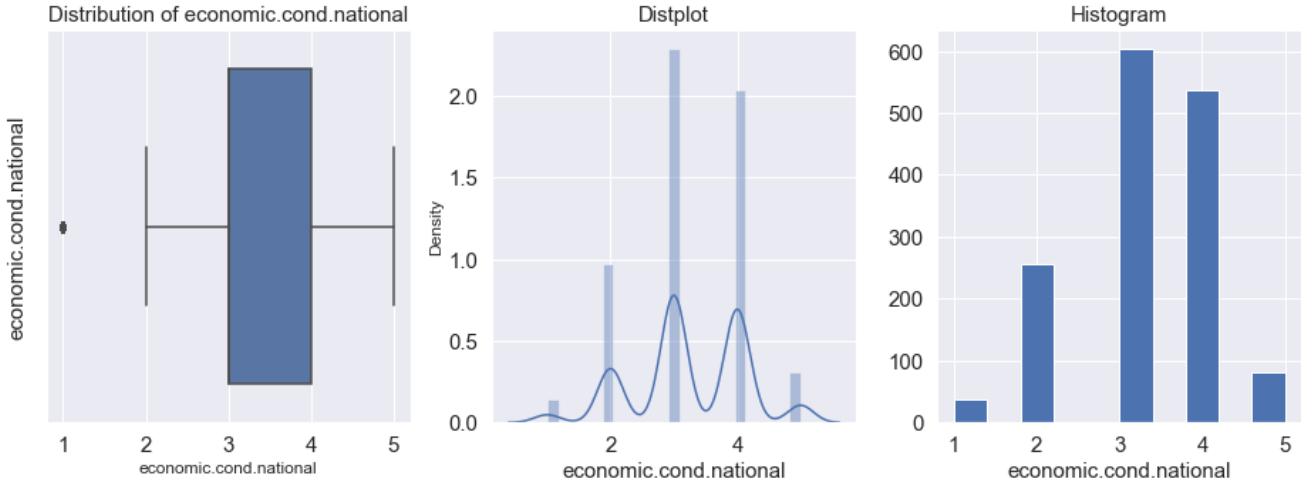


Figure 3: Boxplot,Distplot and Histogram of economic.comd.national

Observation:

- The top 2 variables are 3 and 4.
- 1 has the least value which is 37.
- has the highest value which is 604.
- is slightly higher than the 2nd highest variable 4 whosevalue is 538.
- The average score of 'economic.cond.national' is 3.245221
- There are outliers present in the data.

Economic.cond.household:

Data description:

```
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
```

Value counts:

```
3    645
4    435
2    280
5     92
1     65
Name: economic.cond.household, dtype: int64
```

Countplot:

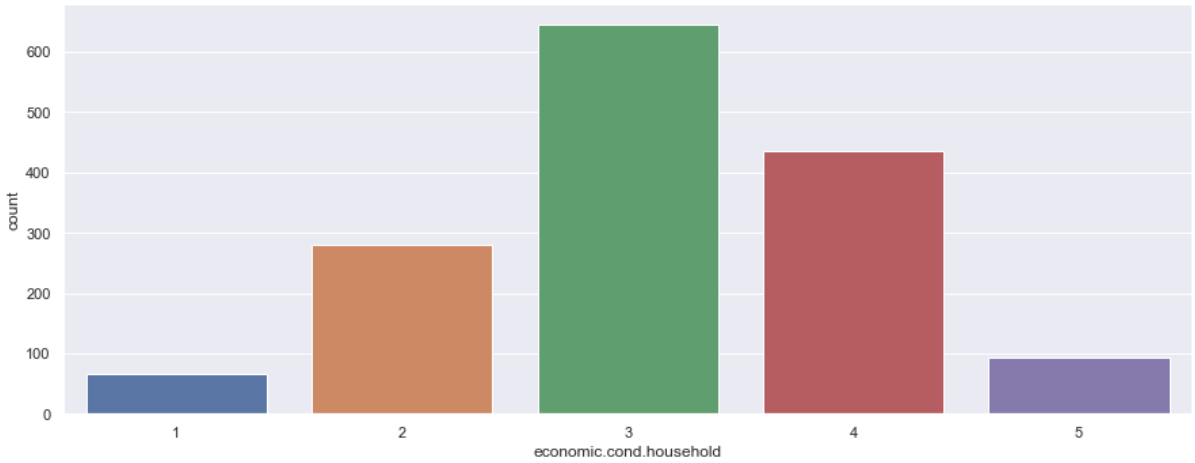
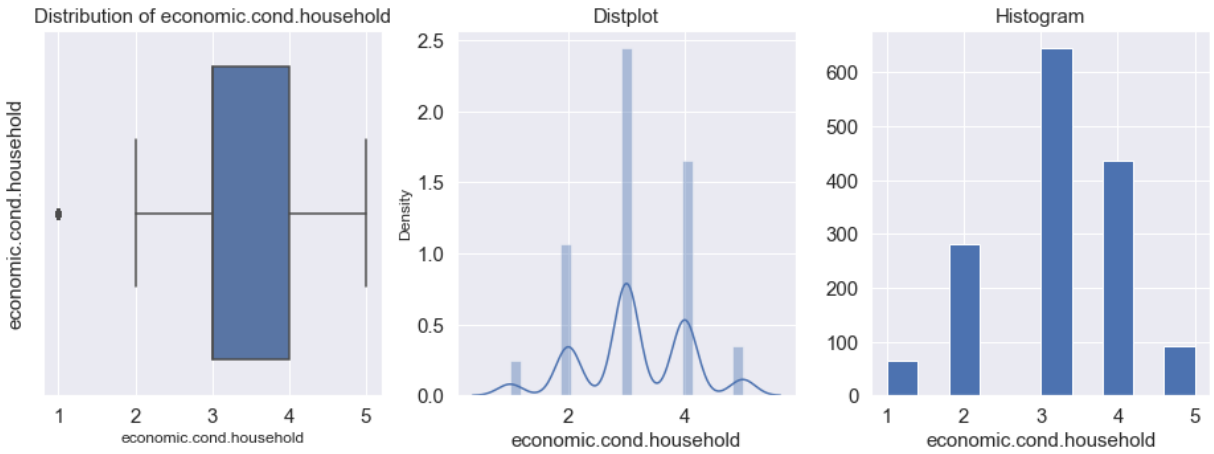


Figure 4:Countplot of economic.cond.household

Boxplot,Distplot and Histogram :



Observations:

- The top 2 variables are 3 and 4.
- 1 has the least value which is 65.
- has the highest value which is 645.
- is moderately higher than the 2nd highest variable 4 whose value is 435.
- The average score of 'economic.cond.household' is 3.137772

Blair:

Descriptive data:

```
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
```

Value counts:

```
4    833
2    434
5    152
1     97
3      1
Name: Blair, dtype: int64
```

Countplot:

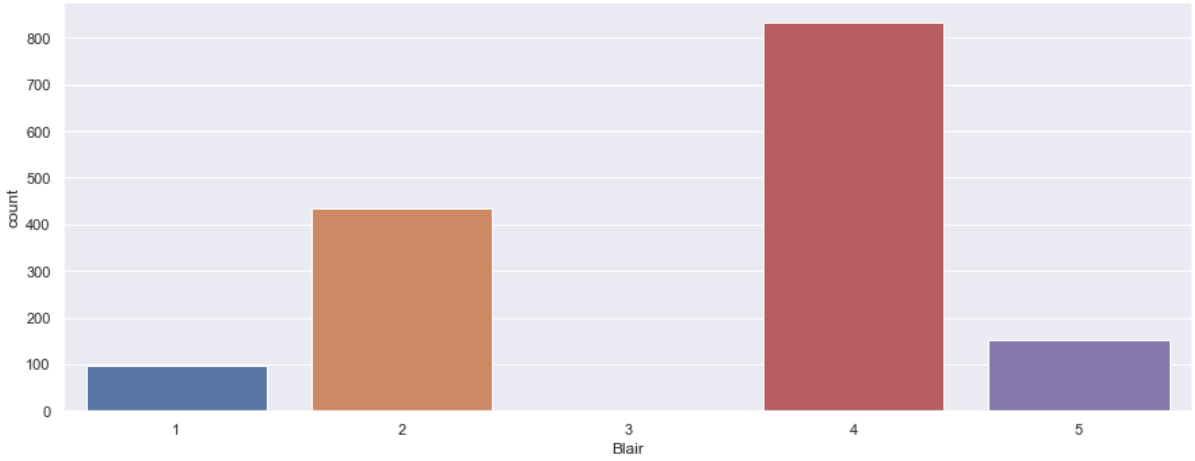


Figure 5: Countplot of Blair

Boxplot, Distplot and Histogram:

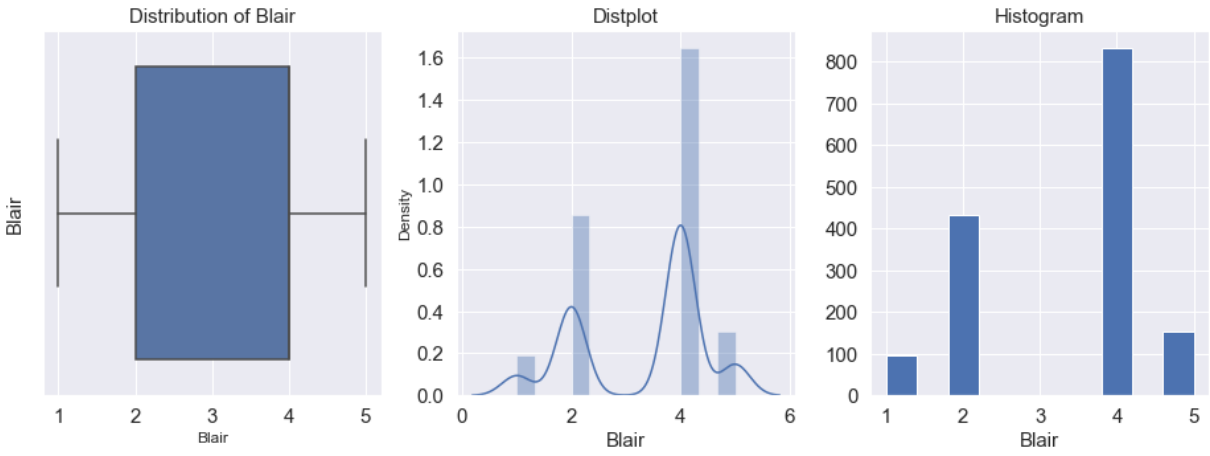


Figure 6: Boxplot, Distplot and Histogram of Blair

Observation:

- The top 2 variables are 2 and 4.
- has the least value which is 1.
- has the highest value which is 833.
- is much higher than the 2nd highest variable 2 whosevalue is 434.
- The average score of 'Blair' is 3.335531
- Found No outlier.

Hague:

Descriptive data:

```
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
```

Value counts:

```
2    617
4    557
1    233
5     73
3     37
Name: Hague, dtype: int64
```

Countplot:

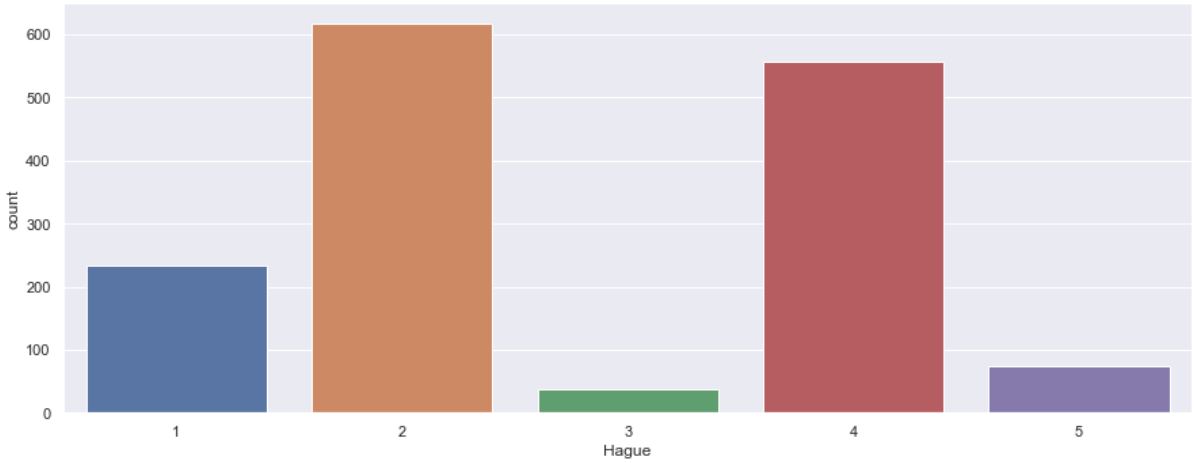
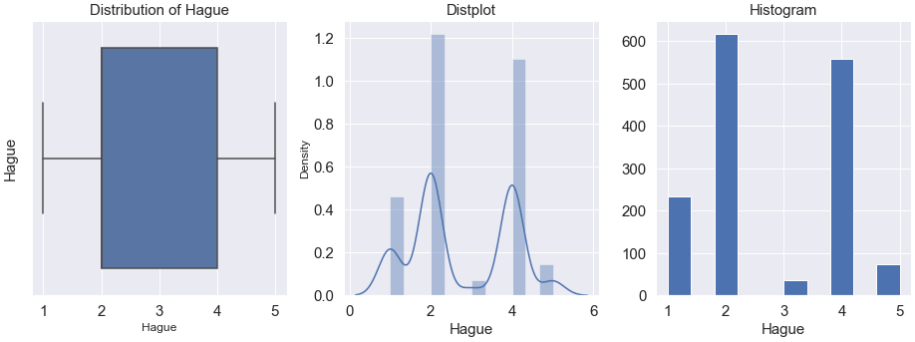


Figure 7: Countplot of Hague

Boxplot, Distplot and Histogram of Hague:



Observations:

- The top 2 variables are 2 and 4.
- has the least value which is 37.
- has the highest value which is 617.
- is slightly higher than the 2nd highest variable 4 whosevalue is 557.
- The average score of 'Blair' is 2.749506
- No outliers found.

Europe:

Descriptive data:

```
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
```

Value counts:

```
11    338
6     207
3     128
4     126
5     123
9     111
8     111
1     109
10    101
7      86
2      77
Name: Europe, dtype: int64
```

Countplot:

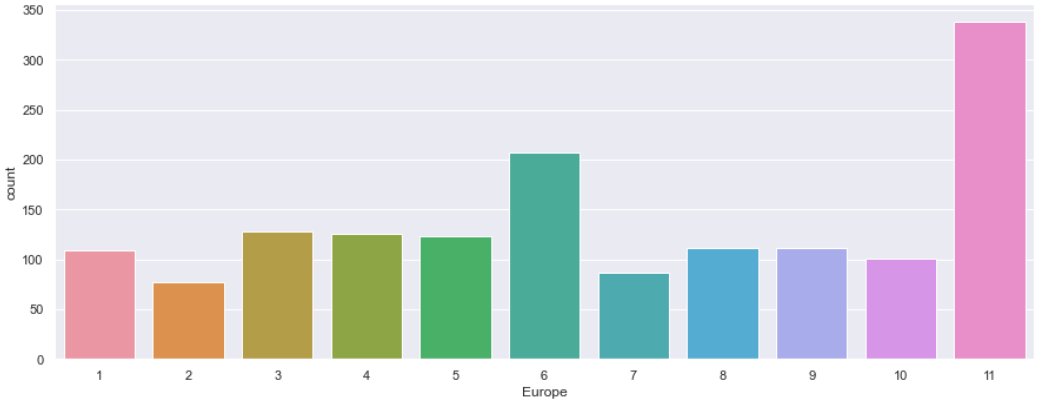


Figure 8:Countplot of Europe

Boxplot, Distplot and Histogram od Europe:

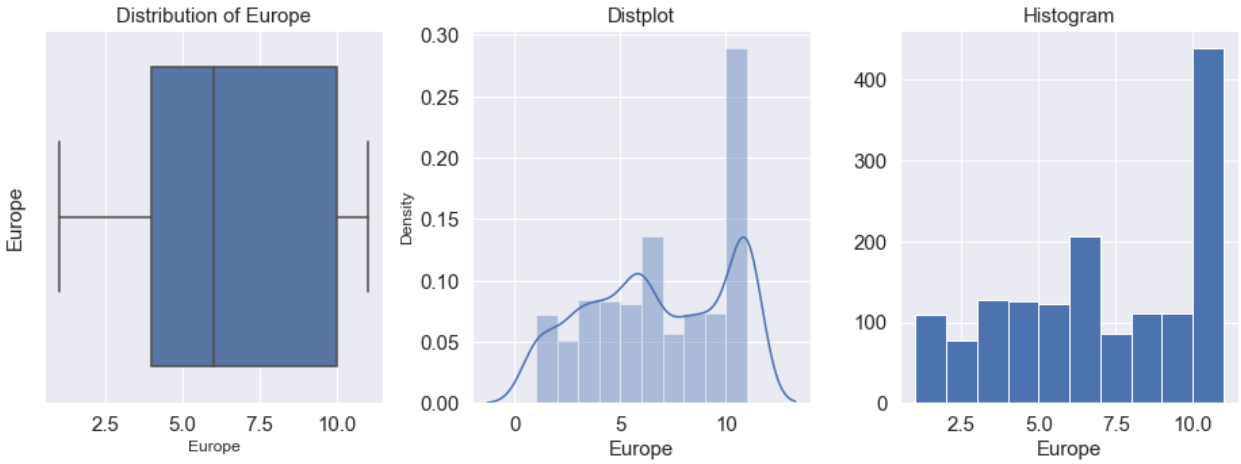


Figure 9:Boxplot,Distplot and histogram of Europe

Observation:

- The top 2 variables are 11 and 6.
- has the least value which is 77.
- 11 has the highest value which is 338.
- 11 is moderately higher than the 2nd highest variable 6 whose value is 207.
- The average score of 'Europe' is 6.740277
- No outliers are found.

Political.knowledge:

Descriptive data

```
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
```

Value Counts:

```
2    776
0    454
3    249
1     38
Name: political.knowledge, dtype: int64
```

Countplot:

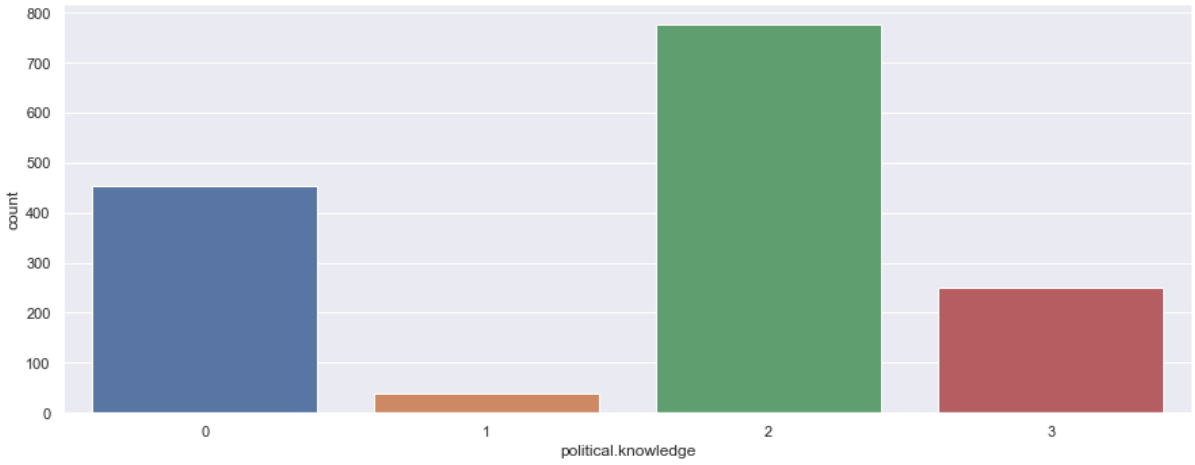


Figure 10:Countplot of political.knowledge

Boxplot, Distplot and Histogram of political.knowledge:

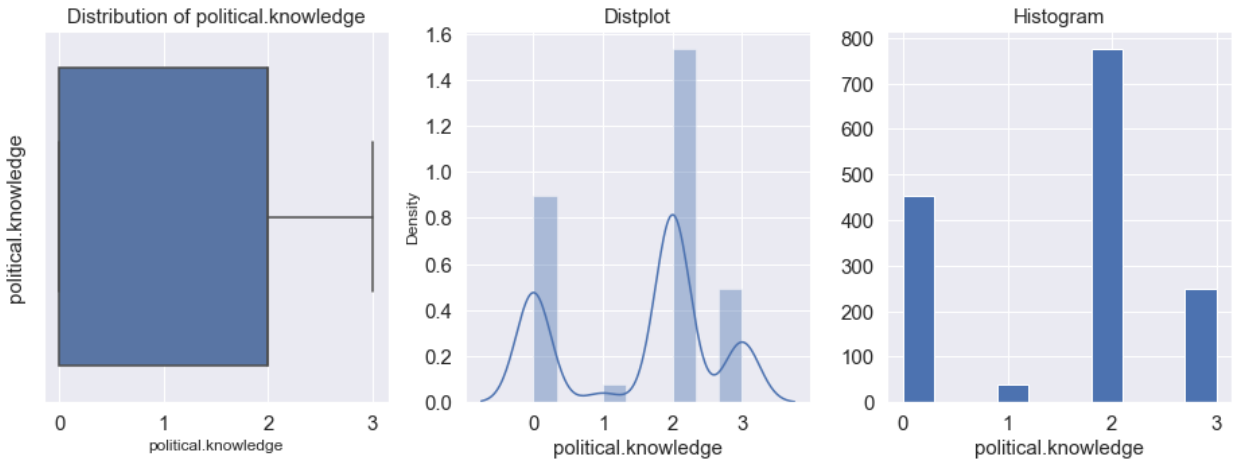


Figure 11:Boxplot, Distplot and Histogram of political.knowledge

Observation:

- The top 2 variables are 2 and 0.
- 1 has the least value which is 38.
- has the highest value which is 776.
- is much higher than the 2nd highest variable 0 whose value is 454.
- We can see that, 454 out of 1517 people do not have any knowledge of parties' positions on European integration which is 29.93% of the total population.
- The average score of 'Europe' is 6.740277
- No outliers are found.

Gender:

Value counts:

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

Countplot:

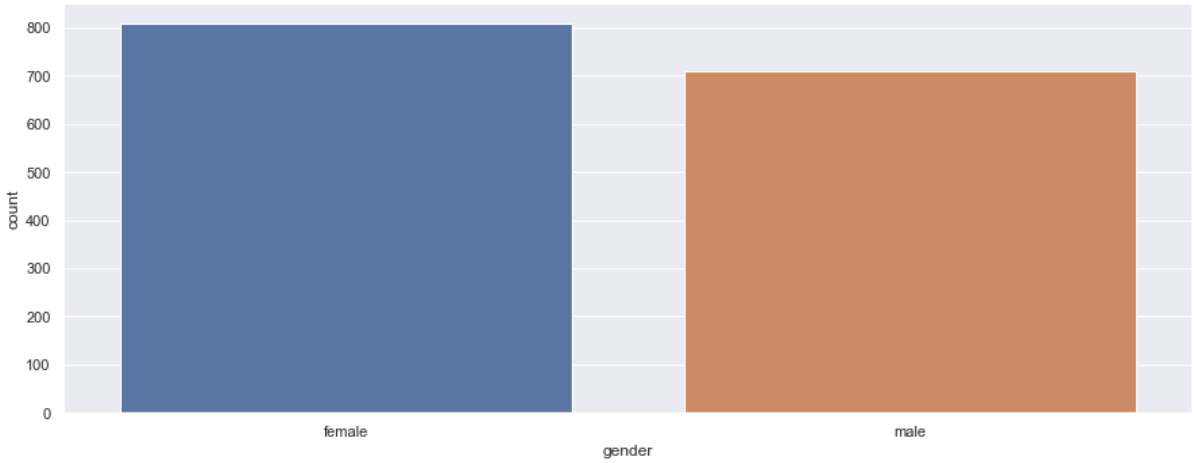


Figure 12:Countplot of gender

Observation:

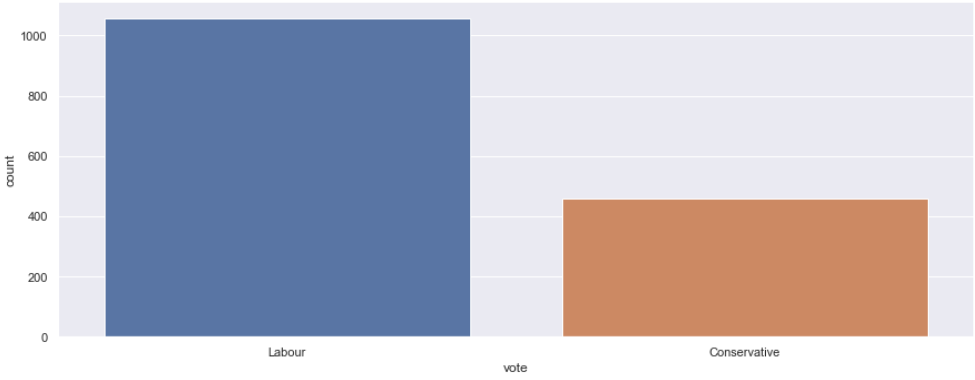
- Found women's count is more which is 808 and men's with 709 counts which is almost equal.

Vote:

Value counts:

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

Countplot:



Observation:

- Found most of the votes are from 'labours' with1057 counts followed by 'conservatives' with 460 counts.

Checking skewness:

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

Insights:

- If the skewness is between -0.5 and 0.5, the data are fairly symmetrical.
- If the skewness is between -1 and – 0.5 or between 0.5and 1, the data are moderately skewed.
- If the skewness is less than -1 or greater than 1, the data are highly skewed.
- Here, we can see that there isn't much skewness in the data.
- All the values seem to be between -0.5 and 0.5.
- The value of 'Blair' is a little bit higher than -0.5.
- The data overall, is fairly symmetrical.

Bivariate and Multivariate Analysis:

Vote and Age:

Boxplot of vote and age:

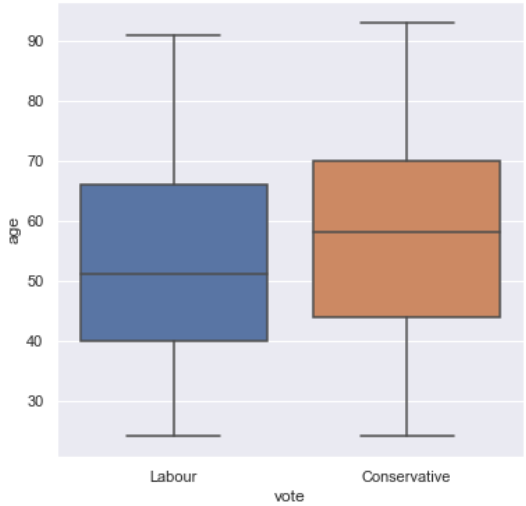


Figure 13: Boxplot of vote and age

Stripplot:

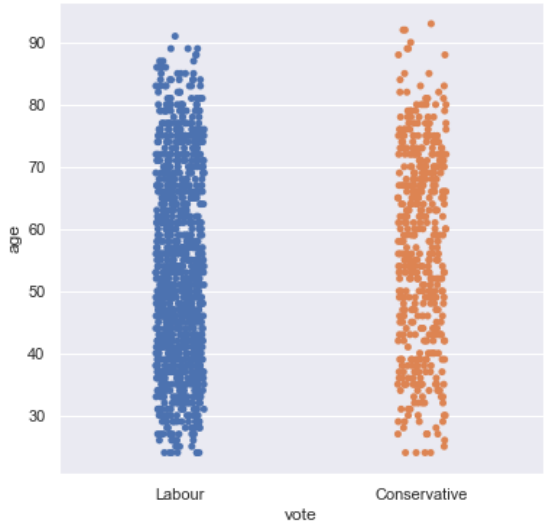


Figure 14: Stripplot of vote and age

Vote and Gender:

Value counts:

```
vote          gender
Labour        female    551
              male      506
Conservative  female    257
              male      203
dtype: int64
```

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 conservative 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 party.

Vote and economic.cond.national:

Stripplot:

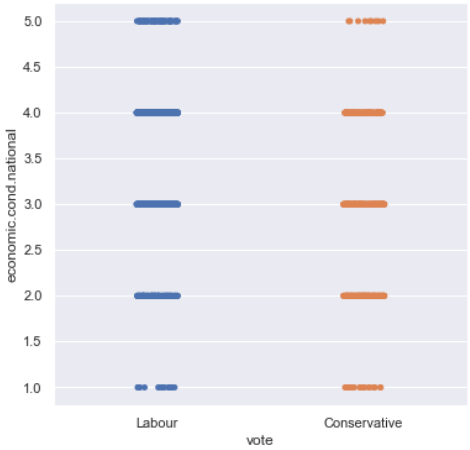


Figure 15: Stripplot of Vote and economic.cond.national

Value counts:

```
economic.cond.national  vote
1                       Conservative     21
                        Labour           16
2                       Conservative    140
                        Labour          116
3                       Labour          405
                        Conservative    199
4                       Labour          447
                        Conservative     91
5                       Labour           73
                        Conservative      9
dtype: int64
```

Observations:

- 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 and economic.cond.household:

Stripplot:

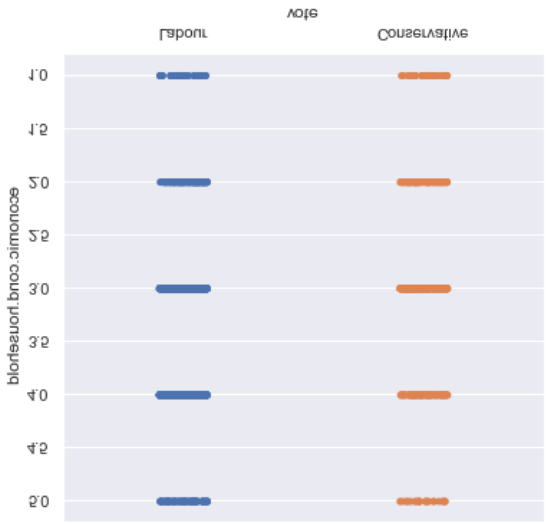


Figure 16:Stripplot of vote and economic.cond.household

Value counts:

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

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:

Stripplot:

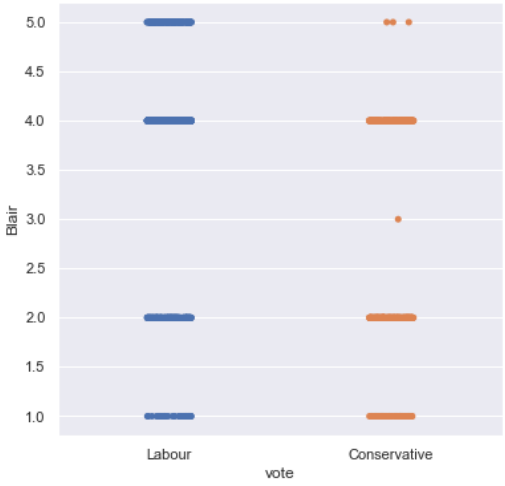


Figure 17:Stripplot of Vote and Blair

Value counts:

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

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:

Stripplot:

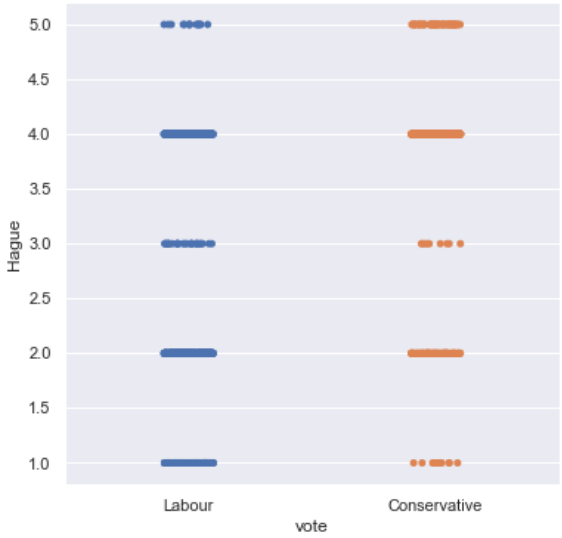


Figure 18:Stripplot of Vote and Hague

Value counts:

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

Observation:

- 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 14people, 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 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 and Europe:

Strippplot:

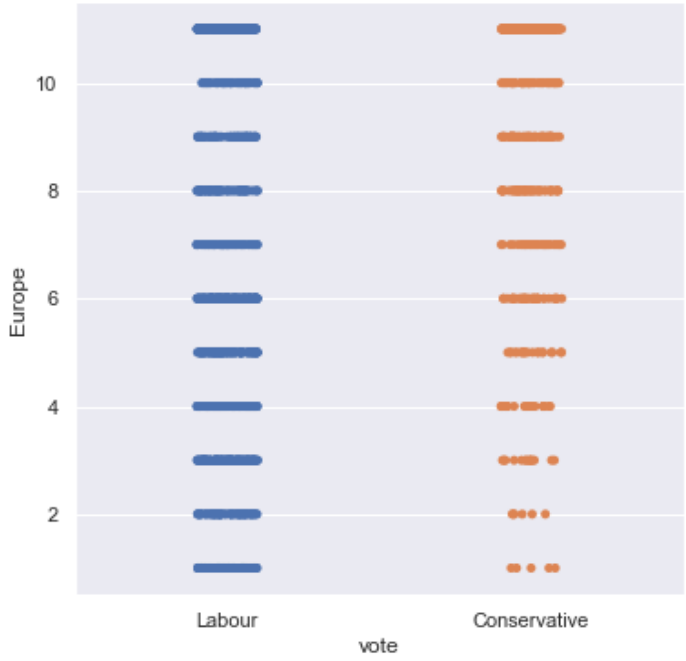


Figure 19:Stripplot of vote and Europe

Countplot:

```
Europe  vote
1       Labour          104
        Conservative      5
2       Labour           71
        Conservative      6
3       Labour          114
        Conservative     14
4       Labour          108
        Conservative     18
5       Labour          103
        Conservative     20
6       Labour          172
        Conservative     35
7       Labour           54
        Conservative     32
8       Labour           63
        Conservative     48
9       Conservative     56
        Labour           55
10      Conservative     54
        Labour           47
11      Conservative    172
        Labour          166
dtype: int64
```

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 almost equally. Conservative party seem to be 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 predominantly 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

Stripplot:

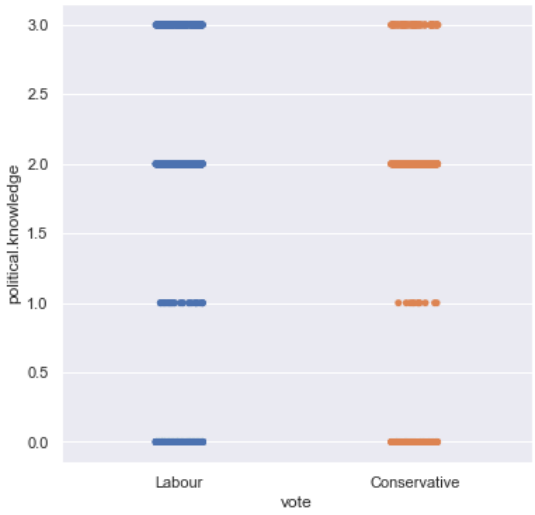


Figure 20: Stripplot of vote and plotical.knowledge

Value counts:

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

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.

Checking pair-wise distribution of the continuous variables:

Pairplot:

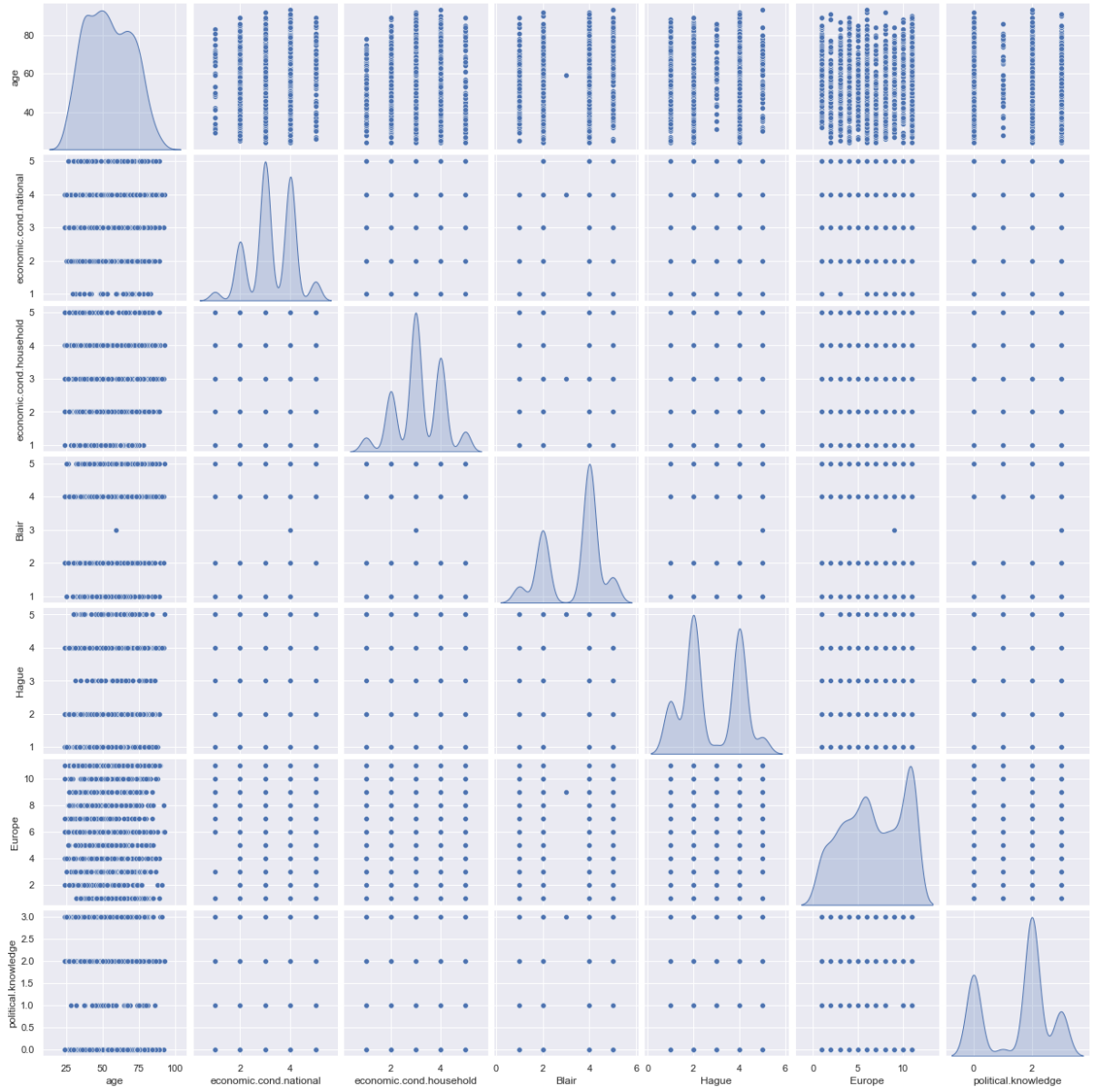


Figure 21: Pairplot

Pairplot wrt vote:

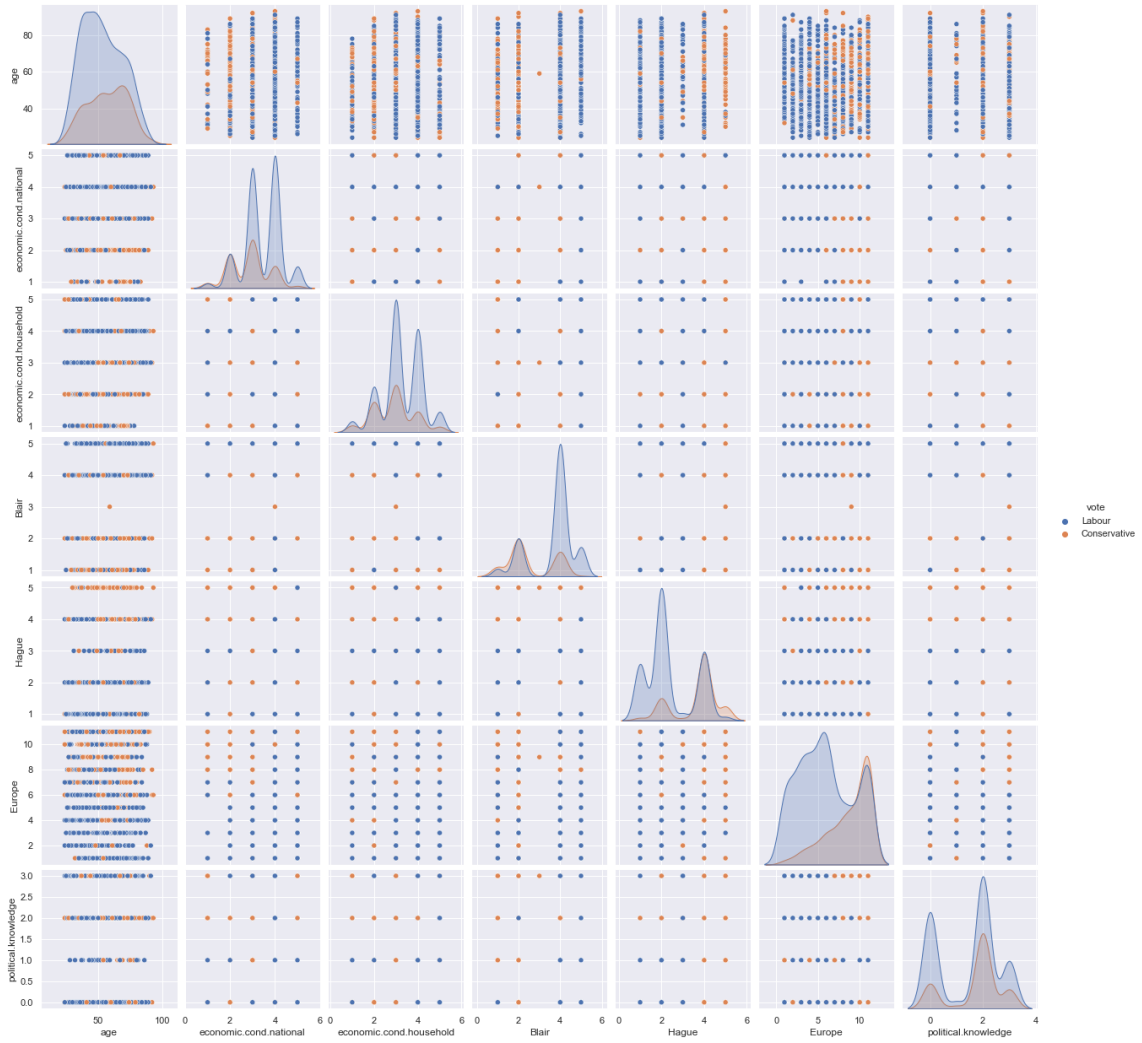


Figure 22:Pairplot wrt vote

Skewness:

vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
Conservative	-0.116904	0.063419	0.140893	0.353688	-0.892401	-0.996315	-0.813972
Labour	0.253535	-0.355276	-0.255752	-1.035030	0.574230	0.208960	-0.267753

Table: Skewness wrt Votes

Observation:

- 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.

- The 'conservative' votes in 'Hague','Europe' and 'political.knowledg' are slightly left skewed. 'age','economic.cond.national', 'economic.cond.household' are almost symmetrical. 'Blair' is right skewed.
- The 'Labour' votes in 'economic.cond.national', 'economic.cond.household','Blair' and 'political.knowledge' are left skewed. 'age', 'Hague' and 'Europe' are right skewed.
- All other variables seem to be normally distributed in overall votes.
- From the scatter plots, we can see that, there is mostly no correlation between the variables.

Correlation matrix:

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
age	1.000000	0.018687	-0.038868	0.032084	0.031144	0.064562	-0.046598
economic.cond.national	0.018687	1.000000	0.347687	0.326141	-0.200790	-0.209150	-0.023510
economic.cond.household	-0.038868	0.347687	1.000000	0.215822	-0.100392	-0.112897	-0.038528
Blair	0.032084	0.326141	0.215822	1.000000	-0.243508	-0.295944	-0.021299
Hague	0.031144	-0.200790	-0.100392	-0.243508	1.000000	0.285738	-0.029906
Europe	0.064562	-0.209150	-0.112897	-0.295944	0.285738	1.000000	-0.151197
political.knowledge	-0.046598	-0.023510	-0.038528	-0.021299	-0.029906	-0.151197	1.000000

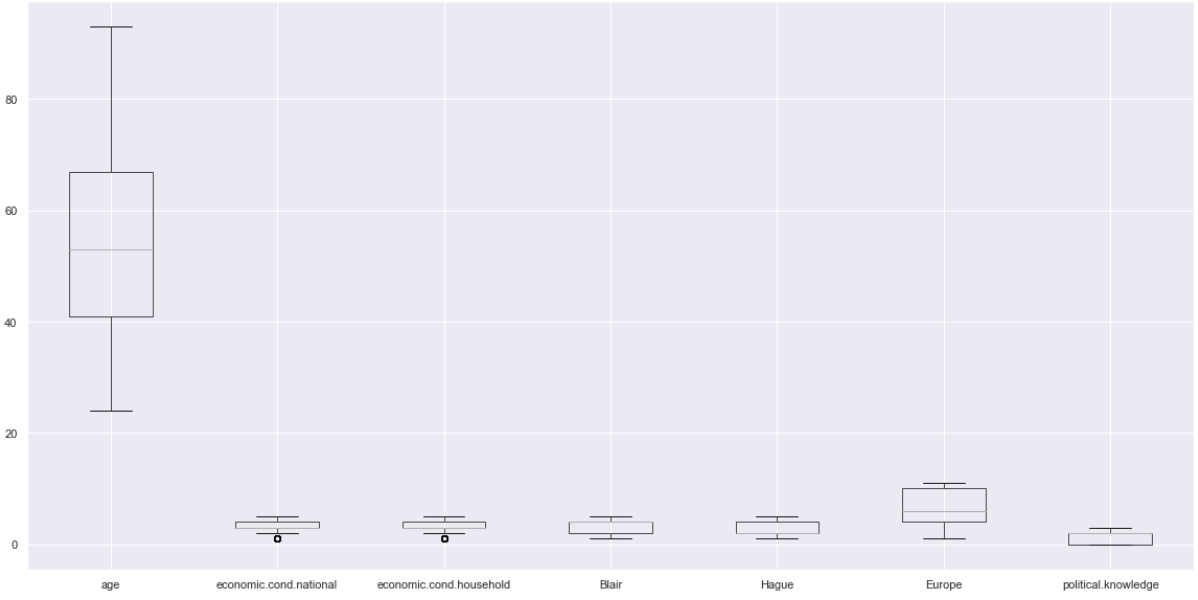
Table 1:Correlation matrix

Heatmap of correlation:

Observation:

- 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.

Checking outliers:



Observation:

- We found outliers in 'economic.cond.national' and 'economic.cond.household' variables. and we are not going to treat outliers in this approach as it is not required.

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).

As we found two non-integer columns vote and gender, we are encoding the values.

Below is the sample dataframe after encoding:

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	vote_Labour	gender_male
645	41	4	4	4	1	4	0	1	0
1377	66	4	3	4	2	6	2	1	1
883	81	5	3	5	2	8	2	1	1
866	49	2	3	4	1	6	2	1	0
266	71	5	2	4	3	3	3	1	1
831	66	4	3	4	2	5	3	1	0
1406	59	3	2	1	2	1	2	1	1
595	38	4	3	5	5	3	2	1	1
140	35	4	3	4	2	4	3	1	1
259	45	3	3	2	2	1	0	1	0

Table 2:Data after encoding

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1517 entries, 0 to 1524
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   age                     1517 non-null   int64
 1   economic.cond.national  1517 non-null   int64
 2   economic.cond.household 1517 non-null   int64
 3   Blair                   1517 non-null   int64
 4   Hague                   1517 non-null   int64
 5   Europe                  1517 non-null   int64
 6   political.knowledge     1517 non-null   int64
 7   vote_Labour             1517 non-null   uint8
 8   gender_male             1517 non-null   uint8
dtypes: int64(7), uint8(2)
memory usage: 130.1 KB
```

The above data looks fine for our model creations.

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 greatly 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.

Scaled dataframe header:

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	vote_Labour	gender_male
0	0.275362	0.50	0.50	0.75	0.00	0.1	0.666667	1.0	0.0
1	0.173913	0.75	0.75	0.75	0.75	0.4	0.666667	1.0	1.0
2	0.159420	0.75	0.75	1.00	0.25	0.2	0.666667	1.0	1.0
3	0.000000	0.75	0.25	0.25	0.00	0.3	0.000000	1.0	0.0
4	0.246377	0.25	0.25	0.00	0.00	0.5	0.666667	1.0	1.0

Table 3:Scaled dataframe

Train-test-split:

Our model will use all the variables and 'vote_Labour' is the target variable. The train-test split is a technique for evaluating the performance of a machine learning algorithm. The procedure involves taking a dataset and dividing it into two subsets.

- Train Dataset: Used to fit the machine learning model.
- Test Dataset: Used to evaluate the fit machine learning model.

Independent variables dataframe sample:

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
0	0.275362	0.50	0.50	0.75	0.00	0.1	0.666667	0.0
1	0.173913	0.75	0.75	0.75	0.75	0.4	0.666667	1.0
2	0.159420	0.75	0.75	1.00	0.25	0.2	0.666667	1.0
3	0.000000	0.75	0.25	0.25	0.00	0.3	0.000000	0.0
4	0.246377	0.25	0.25	0.00	0.00	0.5	0.666667	1.0

Table 4:Independent variables data sample:

Target variable data sample:

	vote_Labour
74	1.0
134	0.0
1348	1.0
595	0.0
1049	0.0
959	0.0
1203	1.0
928	1.0
191	1.0
803	0.0

After splitting the data into train and test data sets, below are the shape of the dataframes:

```
The shape of x_train is  (1061, 8)
The shape of y_train is  (1061, 1)
The shape of x_test  (456, 8)
The shape of y_test is  (456, 1)
```

Percentage of data in training and test dataset:

```
69.94% data is in training set
30.06% data is in test set
```

True-False value distribution in datframes:

```
Original Vote_Labour True Values    : 1057 (69.68%)
Original Vote_Labour False Values   : 460 (30.32%)

Training Vote_Labour True Values    : 754 (71.07%)
Training Vote_Labour False Values   : 307 (28.93%)

Test Vote_Labour True Values        : 303 (66.45%)
Test Vote_Labour False Values       : 153 (33.55%)
```

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

Logistic Regression Model:

Parameters considered for the building of Logistic Regression model:

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

The score of the Logistic Regression model with Training data:

```
0.8312912346842601
```

The classification Report of the model with training data:

	precision	recall	f1-score	support
0.0	0.74	0.64	0.69	307
1.0	0.86	0.91	0.88	754
accuracy			0.83	1061
macro avg	0.80	0.77	0.79	1061
weighted avg	0.83	0.83	0.83	1061

The score of the Logistic Regression model with Testing data:

```
0.8355263157894737
```

The classification Report of the model with Testing data:

	precision	recall	f1-score	support
0.0	0.76	0.74	0.75	153
1.0	0.87	0.88	0.88	303
accuracy			0.84	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.84	0.83	456

Mean-Square Error of LR model:

```
Mean square error-Train data: 0.16870876531573986
Mean square error-Test data: 0.16447368421052633
```

Logistic Regression Model - Observation:

Train data:

- Accuracy: 83.12%
- Precision: 86%
- Recall: 91%
- F1-Score: 88%
- Mean square error (MSE): 0.16870876531573986

Test data:

- Accuracy: 83.55%
- Precision: 87%
- Recall: 88%
- F1-Score: 88%
- Mean square error (MSE): 0.16447368421052633

Insights:

- The model is not over-fitted or under-fitted.
- 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.

Linear Discriminant Analysis Model (LDA):

We built the LDA model with default parameters.

The score of the LR model with Training data:

```
0.8341187558906692
```

The classification report of the LDA model with training data:

	precision	recall	f1-score	support
0.0	0.65	0.74	0.69	269
1.0	0.91	0.86	0.89	792
accuracy			0.83	1061
macro avg	0.78	0.80	0.79	1061
weighted avg	0.84	0.83	0.84	1061

The score of the LR model with Testing data:

```
0.8333333333333334
```

The classification report of the LDA model with Testing data:

	precision	recall	f1-score	support
0.0	0.73	0.77	0.74	145
1.0	0.89	0.86	0.88	311
accuracy			0.83	456
macro avg	0.81	0.82	0.81	456
weighted avg	0.84	0.83	0.83	456

Mean square error of the LDA model:

```
Mean square error-Train data: 0.16588124410933083
Mean square error-Test data: 0.16666666666666666
```

Linear Discriminant Analysis(LDA) - Observation:

Train data:

- Accuracy: 83.41%
- Precision: 91%
- Recall: 86%
- F1-Score: 89%
- Mean square error(MSE): 0.166

Test data:

- Accuracy: 83.33%
- Precision: 89%
- Recall: 86%
- F1-Score: 88%
- Mean square error (MSE): 0.166

Insights:

- The model is not over-fitted or under-fitted.
- 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.

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

K-Nearest Neighbour Model:

We created the KNN model with default parameters:

The KNN model score with training data:

```
0.8567389255419415
```

Classification Report of the KNN model with Training data:

	precision	recall	f1-score	support
0.0	0.77	0.72	0.74	307
1.0	0.89	0.91	0.90	754
accuracy			0.86	1061
macro avg	0.83	0.82	0.82	1061
weighted avg	0.85	0.86	0.86	1061

The KNN model score with Testing data:

```
0.8201754385964912
```

Classification Report of the KNN model with Testing data:

	precision	recall	f1-score	support
0.0	0.75	0.70	0.72	153
1.0	0.85	0.88	0.87	303
accuracy			0.82	456
macro avg	0.80	0.79	0.79	456
weighted avg	0.82	0.82	0.82	456

Mean square error of the KNN model:

```
Mean square error-Train data: 0.14326107445805844
Mean square error-Test data: 0.17982456140350878
```

The best model score value for the k value between 1 to 30 with 2 values intervals:

```
[0.756578947368421,
 0.7982456140350878,
 0.8201754385964912,
 0.8399122807017544,
 0.8421052631578947,
 0.8333333333333334,
 0.8377192982456141,
 0.8421052631578947,
 0.8377192982456141,
 0.8355263157894737,
 0.8333333333333334,
 0.8333333333333334,
 0.8289473684210527,
 0.8289473684210527,
 0.8267543859649122]
```

The MCE value; The misclassification error value is given as below.

```
[0.24342105263157898,
 0.20175438596491224,
 0.17982456140350878,
 0.1600877192982456,
 0.1578947368421053,
 0.16666666666666663,
 0.16228070175438591,
 0.1578947368421053,
 0.16228070175438591,
 0.16447368421052633,
 0.16666666666666663,
 0.16666666666666663,
 0.17105263157894735,
 0.17105263157894735,
 0.17324561403508776]
```

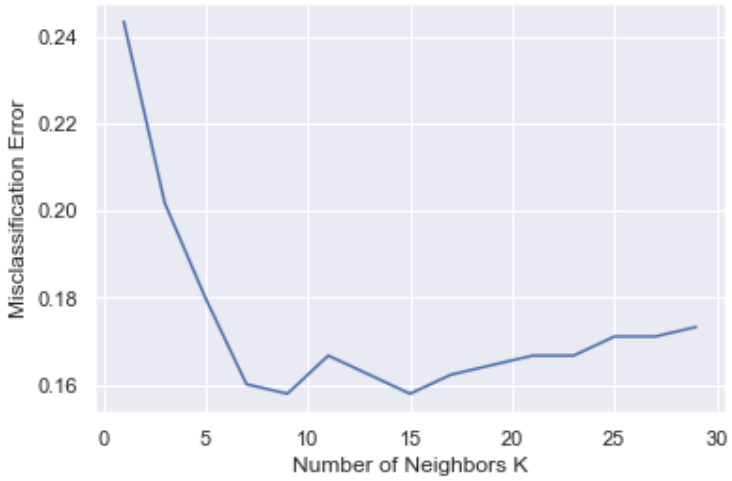


Table 5: Misclassification error against K values

Observation: From the above data we found that the model score is better when the value of K (n_neighbors) value is 9 and 15 which is 0.8421 The error when K=9 or 15 is 0.1578

Now we are building KNN model with K value=9.

The KNN model (k=9) score with training data:

```
0.8454288407163054
```

Classification Report of the KNN model(k=9) with Training data:

	precision	recall	f1-score	support
0.0	0.75	0.70	0.72	307
1.0	0.88	0.91	0.89	754
accuracy			0.85	1061
macro avg	0.82	0.80	0.81	1061
weighted avg	0.84	0.85	0.84	1061

The KNN model(k=9) score with Testing data:

```
0.8421052631578947
```

Classification Report of the KNN model(k=9) with Testing data:

	precision	recall	f1-score	support
0.0	0.78	0.74	0.76	153
1.0	0.87	0.89	0.88	303
accuracy			0.84	456
macro avg	0.83	0.82	0.82	456
weighted avg	0.84	0.84	0.84	456

Mean square error of the KNN model:

```
Mean square error-Train data: 0.15457115928369464
Mean square error-Test data: 0.15789473684210525
```

K Nearest Neighbour Model Observations(k=9):

Train data:

- Accuracy: 84.54%
- Precision: 87%
- Recall: 89%
- F1-Score: 88%
- Mean square error (MSE): 0.154

Test data:

- Accuracy: 84.21%
- Precision: 87%
- Recall: 89%
- F1-Score: 88%
- Mean square error (MSE): 0.157
-

Insights of the Model:

- The model is not over-fitted or under-fitted.
- The error in the test data is slightly lesser 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.

Naive Bayes:

We created the Naïve Bayes model with default parameters.

The NB model score with training data:

```
0.8350612629594723
```

Classification Report of the NB with Training data:

	precision	recall	f1-score	support
0.0	0.73	0.69	0.71	307
1.0	0.88	0.90	0.89	754
accuracy			0.84	1061
macro avg	0.80	0.79	0.80	1061
weighted avg	0.83	0.84	0.83	1061

The NB model score with Testing data:

```
0.8223684210526315
```

Classification Report of the KNN model(k=9) with Testing data:

	precision	recall	f1-score	support
0.0	0.74	0.73	0.73	153
1.0	0.87	0.87	0.87	303
accuracy			0.82	456
macro avg	0.80	0.80	0.80	456
weighted avg	0.82	0.82	0.82	456

Mean square error of the KNN model:

```
Mean square error-Train data: 0.1649387370405278
Mean square error-Test data: 0.17763157894736842
```

Naive Bayes Model Observations(k=9):

Train data:

- Accuracy: 83.5%
- Precision: 88%
- Recall: 90%
- F1-Score: 89%
- Mean square error (MSE): 0.165

Test data:

- Accuracy: 82.23%
- Precision: 87%
- Recall: 87%
- F1-Score: 87%
- Mean square error (MSE): 0.177

Insight of the Model

- The model is not over-fitted or under-fitted.
- 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.

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

Tuning is the process of maximizing a model's performance without overfitting or creating too high of a variance. In machine learning, this is accomplished by selecting appropriate "hyperparameters."

Hyperparameters can be thought of as the “dials” or “knobs” of a machine learning model. Choosing an appropriate set of hyperparameters is crucial for model accuracy, but can be computationally challenging. Hyperparameters differ from other model parameters in that they are not learned by the model automatically through training methods. Instead, these parameters must be set manually. Many methods exist for selecting appropriate hyperparameters. This post focuses on three:

- Grid Search
- Random Search
- Bayesian Optimization

Model Tuning:

Logistic Regression:

Logistic Regression tuned model is created with below parameters which creates the best model among different combination of the parameters.

```
LogisticRegression(C=0.7, max_iter=10000, penalty='l1', solver='libline
ar')
```

Tunning LR model score with the training data:

```
Model score of Train data is: 0.8360037700282752
```

Classification report of the tuned LR mode with training datal:

	precision	recall	f1-score	support
0.0	0.76	0.63	0.69	307
1.0	0.86	0.92	0.89	754
accuracy			0.84	1061
macro avg	0.81	0.77	0.79	1061
weighted avg	0.83	0.84	0.83	1061

Tunning LR model score with the Testing data:

```
Model score of Testing data is: 0.8267543859649122
```

Classification report of the tuned LR mode with Testing datal:

	precision	recall	f1-score	support
0.0	0.75	0.72	0.74	153
1.0	0.86	0.88	0.87	303
accuracy			0.83	456
macro avg	0.81	0.80	0.80	456
weighted avg	0.83	0.83	0.83	456

Observation:

- For training data, we could see the improvement in the score of Linear Regression model after tuning. Before Tuning the score was 0.83 and score after tuning is 0.84
- But for testing data, we could not see the any improvements. Model scoring before tuning was 0.84 and after tuning is 0.83
- Overall, there is no much difference noticed in the model. The values are high overall and there is no over-fitting or under-fitting. Therefore, both models are equally good models.

Linear Discriminant Analysis Model Tuning(LDA):

We built the model with best parameters using param grid technique.

```
grid_search_lda=GridSearchCV(ldat, param_grid = param_grid, cv=kfold,
                     scoring="accuracy", n_jobs= 4, verbose = 1)
```

```
Fitting 10 folds for each of 3 candidates, totalling 30 fits
```

Tunning LDA model score with the training data:

```
Model score of Train data is: 0.8341187558906692
```

Classification report of the tuned LDA mode with training datal:

	precision	recall	f1-score	support
0.0	0.74	0.65	0.69	307
1.0	0.86	0.91	0.89	754
accuracy			0.83	1061
macro avg	0.80	0.78	0.79	1061
weighted avg	0.83	0.83	0.83	1061

Tunning LDA model score with the Testing data:

```
Model score of Test data is: 0.8333333333333334
```

Classification report of the tuned LDA mode with Testing datal:

	precision	recall	f1-score	support
0.0	0.77	0.73	0.74	153
1.0	0.86	0.89	0.88	303
accuracy			0.83	456
macro avg	0.82	0.81	0.81	456
weighted avg	0.83	0.83	0.83	456

Observation:

Train data:

- Accuracy: 83.41%
- Precision: 86%
- Recall: 91%
- F1-Score: 89%

Test data:

- Accuracy: 83.33%
- Precision: 86%
- Recall: 89%
- F1-Score: 88%

- Overall, there is no much difference noticed in the model except the values of Precision, Recall and F1-score. The values are high overall and there is no over-fitting or under-fitting. Therefore, both models are equally good models.

K-Nearest Neighbour Model Tuning:

KNN tuned model is created with below parameters obtained from param technique.

```
KNeighborsClassifier(leaf_size=15, n_neighbors=19)
```

Tunning KNN model score with the training data:

```
Model score of Train data is: 0.8350612629594723
```

Classification report of the tuned KNN mode with training datal:

	precision	recall	f1-score	support
0.0	0.76	0.64	0.69	307
1.0	0.86	0.92	0.89	754
accuracy			0.84	1061
macro avg	0.81	0.78	0.79	1061
weighted avg	0.83	0.84	0.83	1061

Tunning KNN model score with the Testing data:

```
Model score of Test data is: 0.8355263157894737
```

Classification report of the tuned KNN mode with Testing datal:

	precision	recall	f1-score	support
0.0	0.80	0.69	0.74	153
1.0	0.85	0.91	0.88	303
accuracy			0.84	456
macro avg	0.82	0.80	0.81	456
weighted avg	0.83	0.84	0.83	456

K Nearest Neighbour Model after tuning Observations:

Train data:

- Accuracy: 83.504%
- Precision: 86%
- Recall: 92%
- F1-Score: 89%

Test data:

- Accuracy: 83.55%
- Precision: 85%
- Recall: 91%
- F1-Score: 88%

Model Insights:

- For Testing data, we could see the improvement in the score of KNN model after tuning. Before Tuning the score was 0.8201 and score after tuning is 0.8355
- But for Training data, we could not see the any improvements. Model scoring before tuning was 0.8567 and after tuning is 0.8350
- Overall, there is no much difference noticed in the model. The values are high overall and there is no over-fitting or under-fitting. Therefore, both models are equally good models.

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.

(All the Confusion matrix, AUS and ROC curves for all the models are shown in the code file)

Comparison of the different models with respect to Training data:

MODEL	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)	AUC

LR - Regular	83.1	86	91	88	0.89
LR - Tuned	83.6	86	92	89	0.89
LDA - Regular	83.41	91	86	89	0.889
LDA - Tuned	83.41	86	91	89	0.889
KNN - Regular	84.54	87	89	88	0.913
KNN - Tuned	83.5	86	92	89	0.901
Naïve Bayes	83.5	88	90	89	0.888
Random Forest- Regular, Tuning	85.76	87	94	90	0.918
Bagging RF	96.6	96	99	98	0.997
Ada Boosting	85.01	88	91	90	0.915
Gradient Boosting	89.25	91	94	93	0.951

Comparison of the different models with respect to Testing data:

MODEL	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)	AUC
LR - Regular	83.5	87	88	88	0.883
LR - Tuned	82.67	86	88	87	0.88
LDA - Regular	83.33	89	86	88	0.888
LDA - Tuned	83.33	86	89	88	0.888
KNN - Regular	84.31	87	89	88	0.892
KNN - Tuned	83.55	85	91	88	0.894
Naïve Bayes - Regular	82.23	87	87	87	0.876
Random Forest- Regular tuning	81.79	82	92	87	0.891
Bagging RF	82.89	85	90	88	0.896
Ada Boosting	81.35	84	88	86	0.877
Gradient Boosting	83.33	85	91	88	0.899

Conclusion:

- There is no under-fitting or over-fitting in any of the tuned models other than KNN model.
- All the tuned models have high values and every model is good.
- We can consider Naïve Bayes model or LR tuned/Regular model as the best models compare to others with accuracy, Precision, AUC, Recall and F1 scores.

1.8 Based on these predictions, what are the insights?

Insights:

- Labour party has more than double the votes of conservative party.
- 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 datset. The tuned models have performed better than the regular models in some cases.
- There is no over-fitting in any model except Random Forest, Bagging and Gradient boosting models.

Business recommendations:

- Hyper-parameters tuning is an import aspect of modelbuilding. There are limitations to this as to process these combinations, huge amount of processing power is required. But if tuning can be done with many sets of parameters, we might get even better results.
- Gathering more data will also help in training the models and thus improving the predictive powers.
- We can also create a function in which all the models predict the outcome in sequence. This will help in better understanding and the probability of what the outcome will be.
- Using Logical Regression with tuning, tuned KNN and NB for predicting the outcome as these has the best optimized performance.

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

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

Character count of the speeches:

- `Character count of President Franklin D. Roosevelt sleech: 7571`
- `Character count of President John F. Kennedy sleech: 7618`
- `Character count of President Richard Nixon sleech: 9991`

Words count if the speeches:

- `Words count of President Franklin D. Roosevelt sleech: 1536`
- `Words count of President John F. Kennedy sleech: 1546`
- `Words count of President Richard Nixon sleech: 2028`

Sentence counts of the speeches:

- `Sentence count of President Franklin D. Roosevelt sleech: 68`
- `Sentence count of President John F. Kennedy sleech: 52`
- `Sentence count of President Richard Nixon sleech: 69`

2.2 Remove all the stopwords from all three speeches.

We removed all the stopwords which are part of English language and punctuations.

Below are the words counts after removing the stopwords from the string.

- `Words count of President Franklin D. Roosevelt sleech after removing stopwords: 632`
- `Words count of President John F. Kennedy sleech after removing stopwords: 697`
- `Words count of President Richard Nixon sleech after removing stopwords: 836`

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)

Below are the top 3 words in each speeches:

- `Top 3 words for Roosevelt: [('nation', 12), ('know', 10), ('spirit', 9)]`
- `Top 3 words for Kennedy: [('let', 16), ('us', 12), ('world', 8)]`
- `Top 3 words for Nixon: [('us', 26), ('let', 22), ('america', 21)]`

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

Word cloud of Roosevelt:

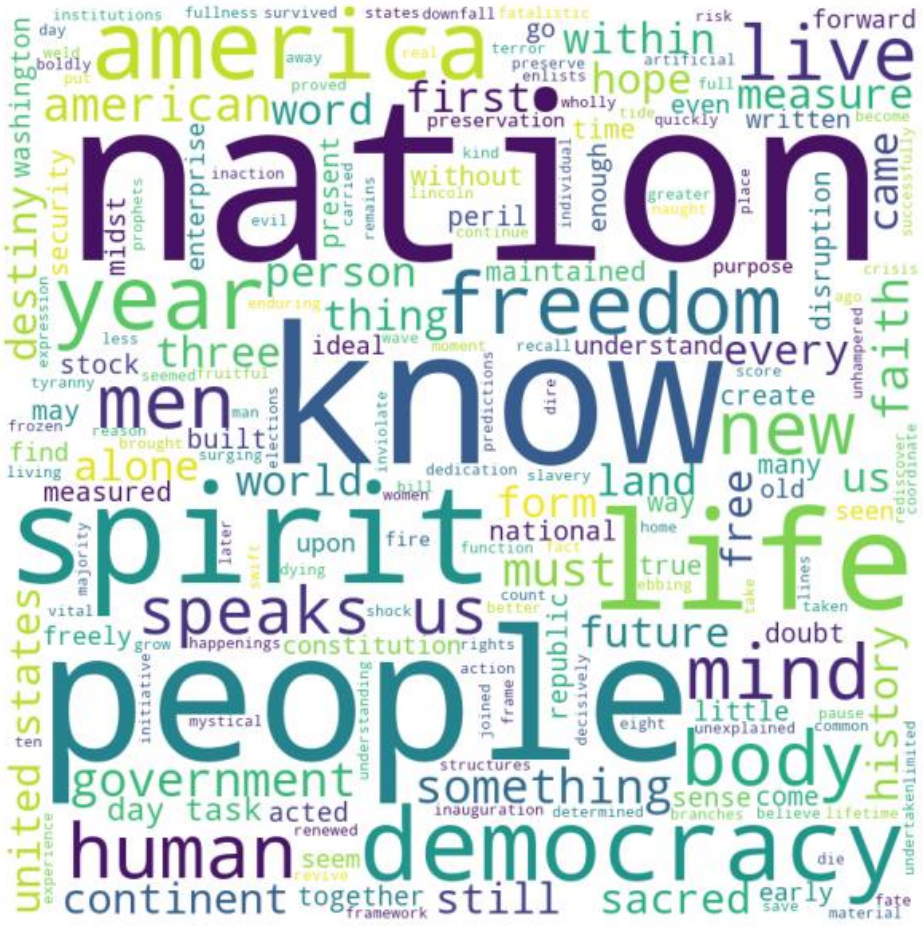


Figure: Wordcloud for Roosevelt speech

Word cloud of Kennedy:

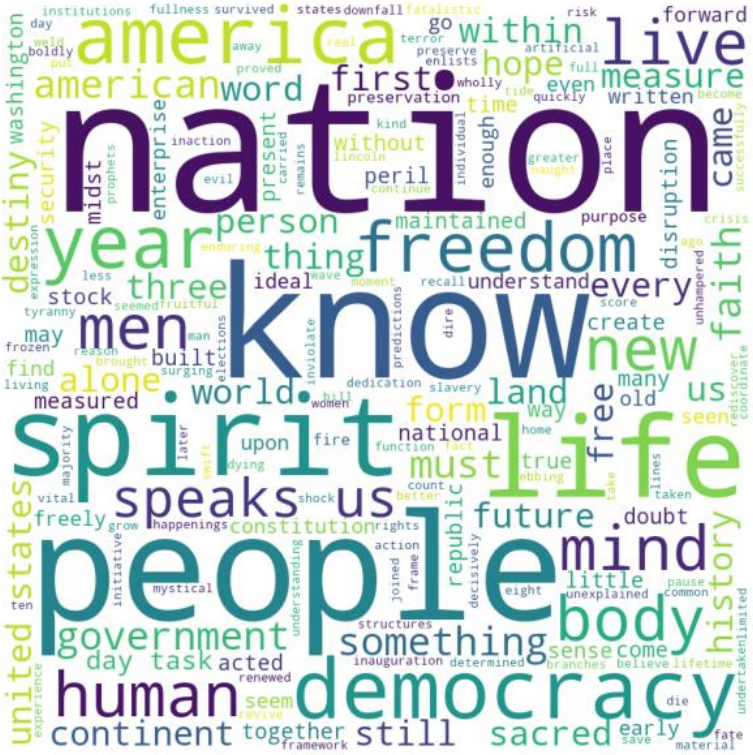


Figure: Word cloud of Kennedy

Word cloud for Nixon:

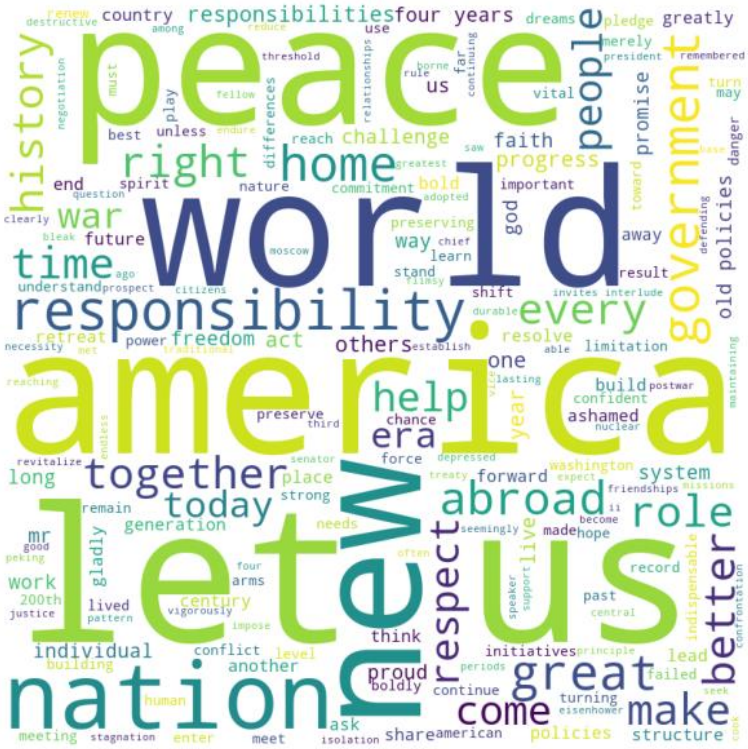


Figure: Wordcloud for Nixon

----------The End----------