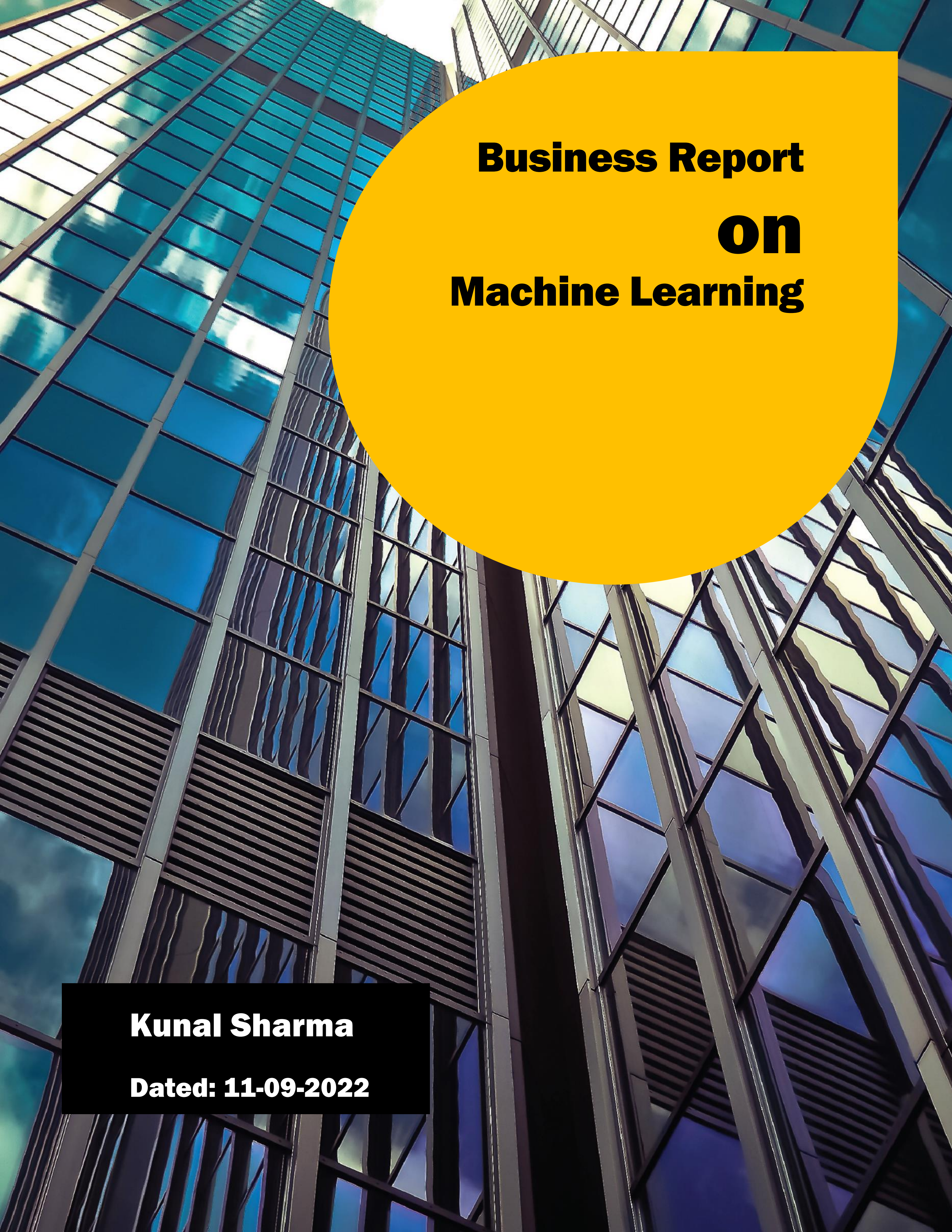


Business Report on Machine Learning

Kunal Sharma

Dated: 11-09-2022

Table of Contents

List of Figures & Tables

Problem 1: Voting Analysis

Executive Summary

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.

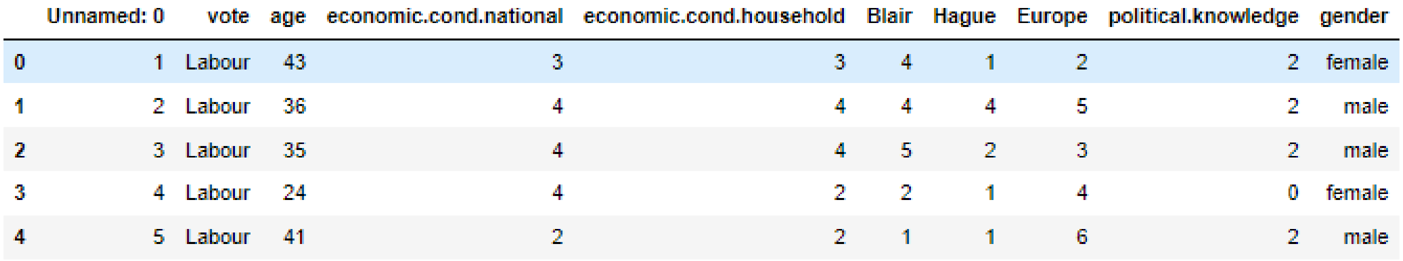
Data Dictionary

- Vote: Party choice: Conservative or Labour
- age: in years
- economic.cond.national: Assessment of current national economic conditions, 1 to 5.
- economic.cond.household: Assessment of current household economic conditions, 1 to 5.
- Blair: Assessment of the Labour leader, 1 to 5.
- Hague: Assessment of the Conservative leader, 1 to 5.
- Europe: an 11-point scale that measures respondents' attitudes toward European integration. High scores represent ‘Eurosceptic’ sentiment.
- political.knowledge: Knowledge of parties' positions on European integration, 0 to 3.
- 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.

Snapshot of DataFrame

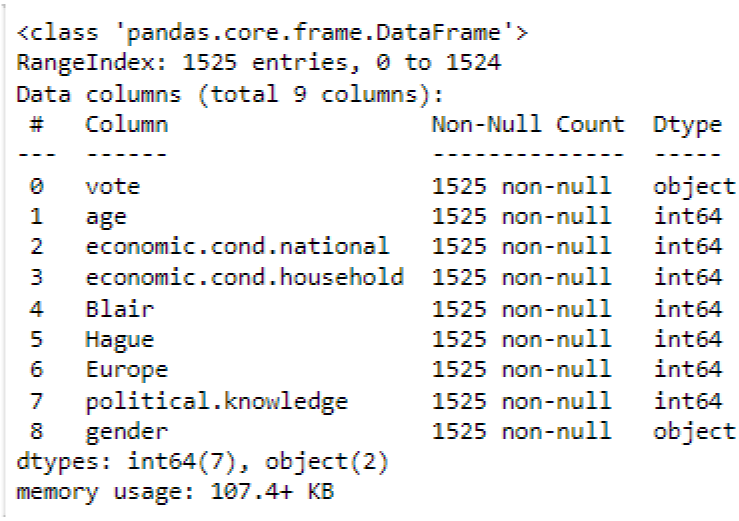
Table 1: Snapshot of Dataframe



	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

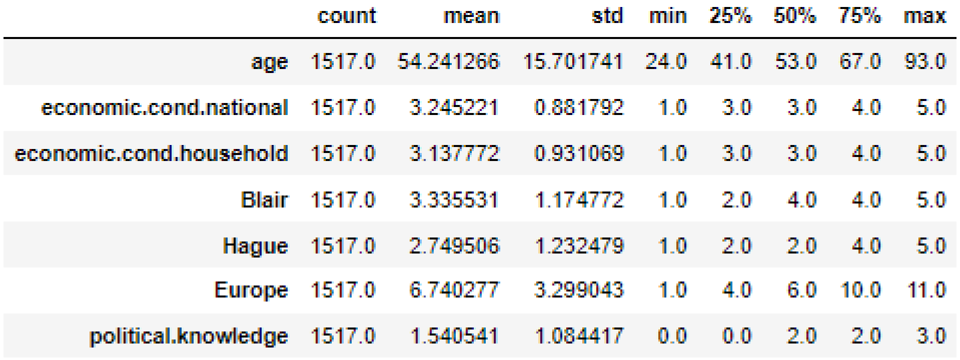
In the given dataset, column 'Unnamed: 0' seems to be of no us hence for the further analysis, we are dropping the particular column from the sample.

Understanding Data and Missing Values

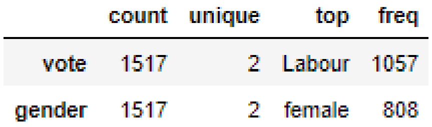


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1525 entries, 0 to 1524
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   vote                     1525 non-null   object
 1   age                      1525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

Table 2: Data Describe



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



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

Duplicates

Number of Duplicate rows: 8

Before: (1525, 9)

After: (1517, 9)

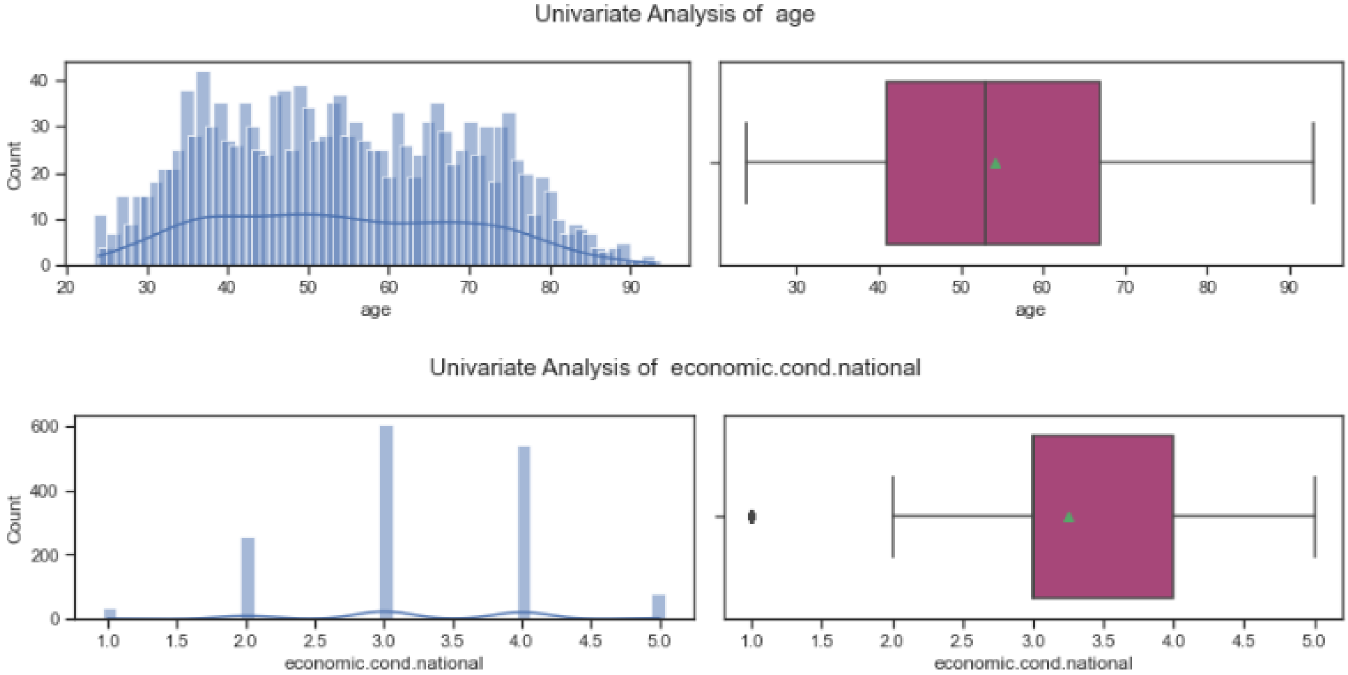
Inference

- Dataset has a total of 1525 observations and 10 columns with 8 features as type integer and 2 as type object.
- Overall, there are no null values present in the data however 8 duplicate values which is 0.52% of our sample.
- Voters age seems to be ranging from 24 to 93 with average voters belongs to age 53 and 75% of voters are of age 67. On the other hand, most of the voters (75%) seems to have rated 4 to national as well as household economic conditions and both seems to have an average rating of 3.
- Majority of the voters seems to be representing 'Eurosceptic' sentiment with rating of 10 though average rating being 6.
- Comparatively, party choice seems to be biased towards Labour with a frequency of 1057 and majority of voters seems to be female with frequency of 808.

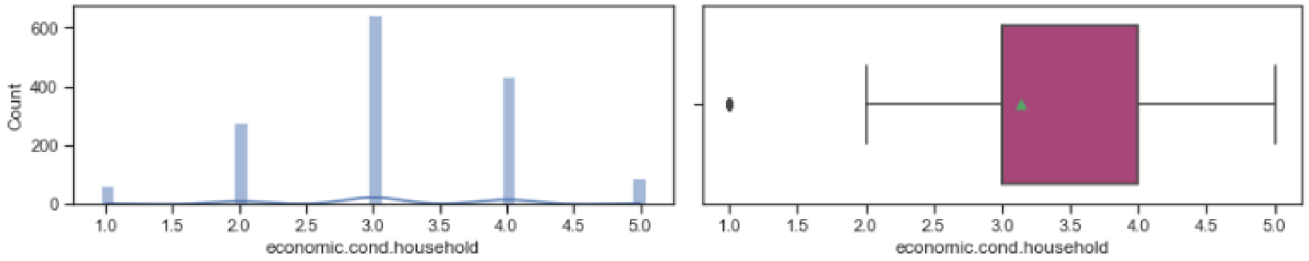
1.2 Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.

Univariate Analysis

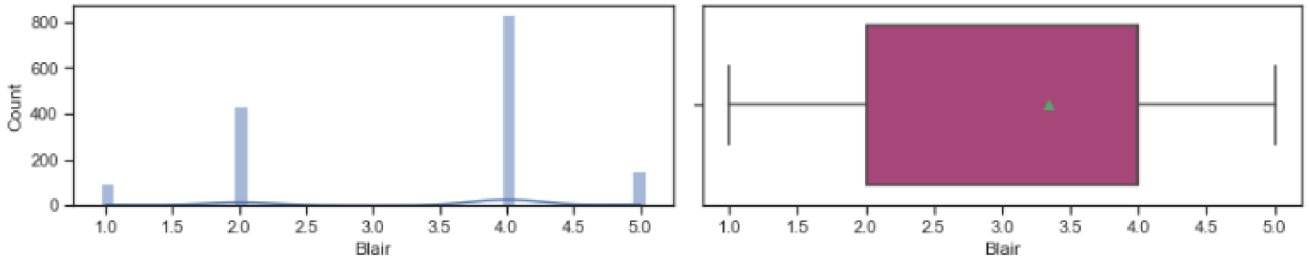
Figure 1Figure 1:Univariate Analysis of Continuous Variables



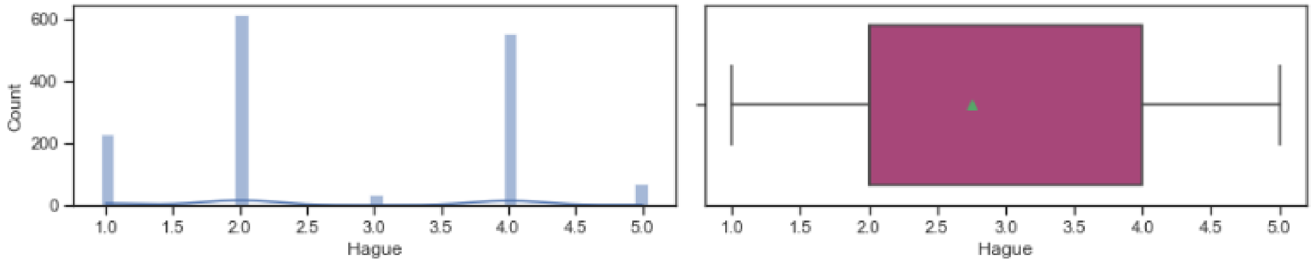
Univariate Analysis of economic.cond.household



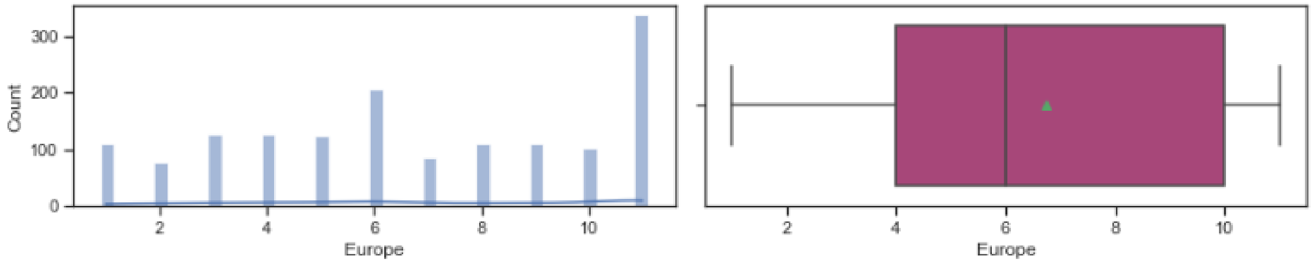
Univariate Analysis of Blair



Univariate Analysis of Hague



Univariate Analysis of Europe



Univariate Analysis of political.knowledge

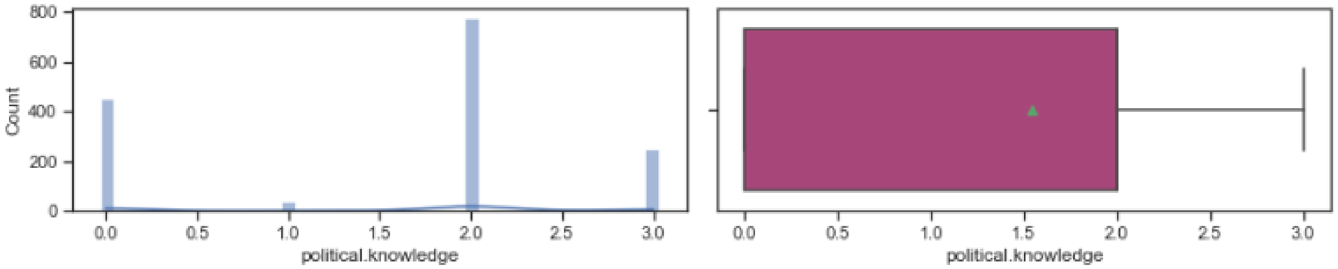
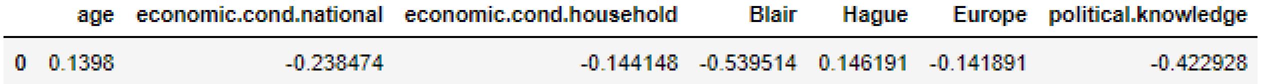
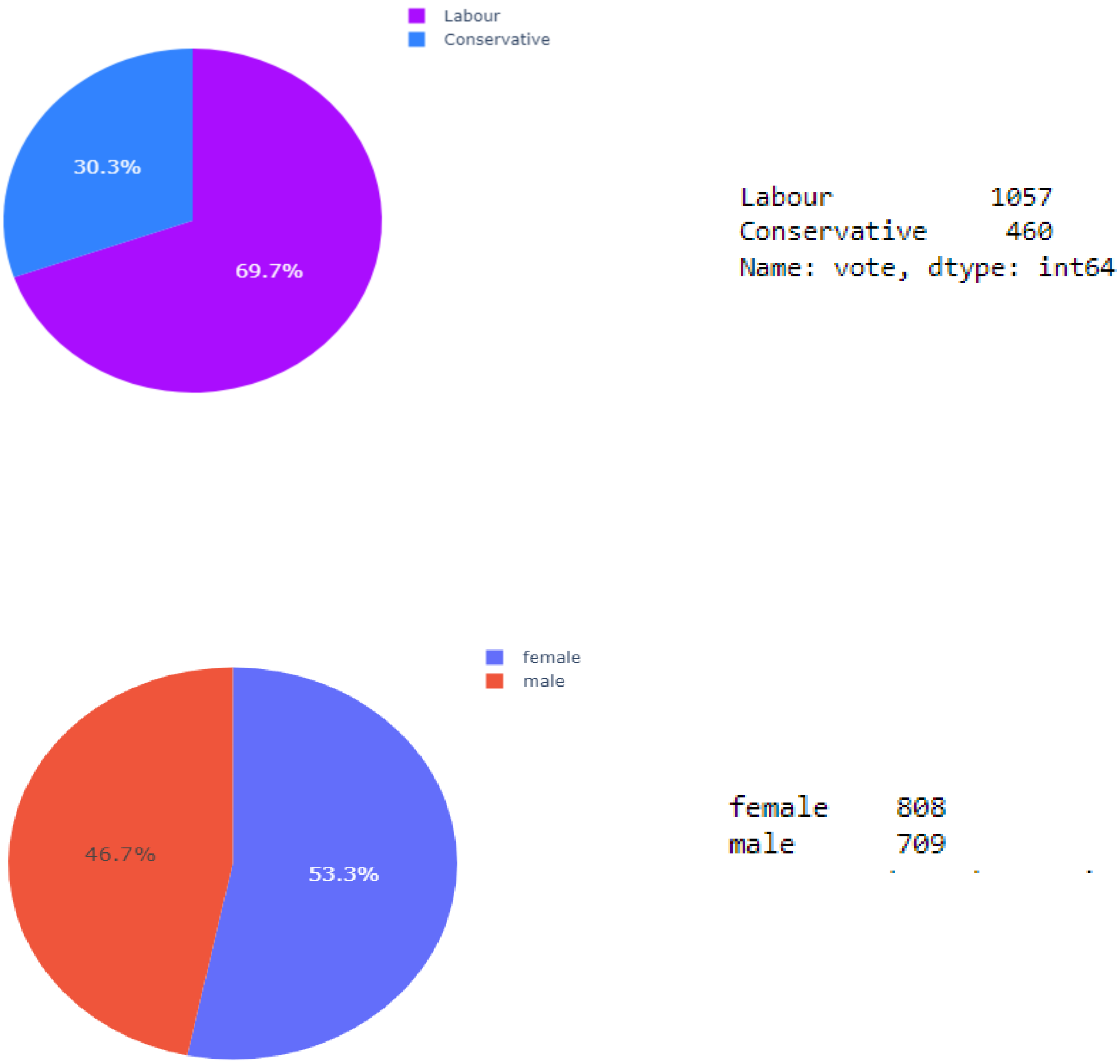


Table 3: Skewness



	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
0	0.1398	-0.238474	-0.144148	-0.539514	0.146191	-0.141891	-0.422928

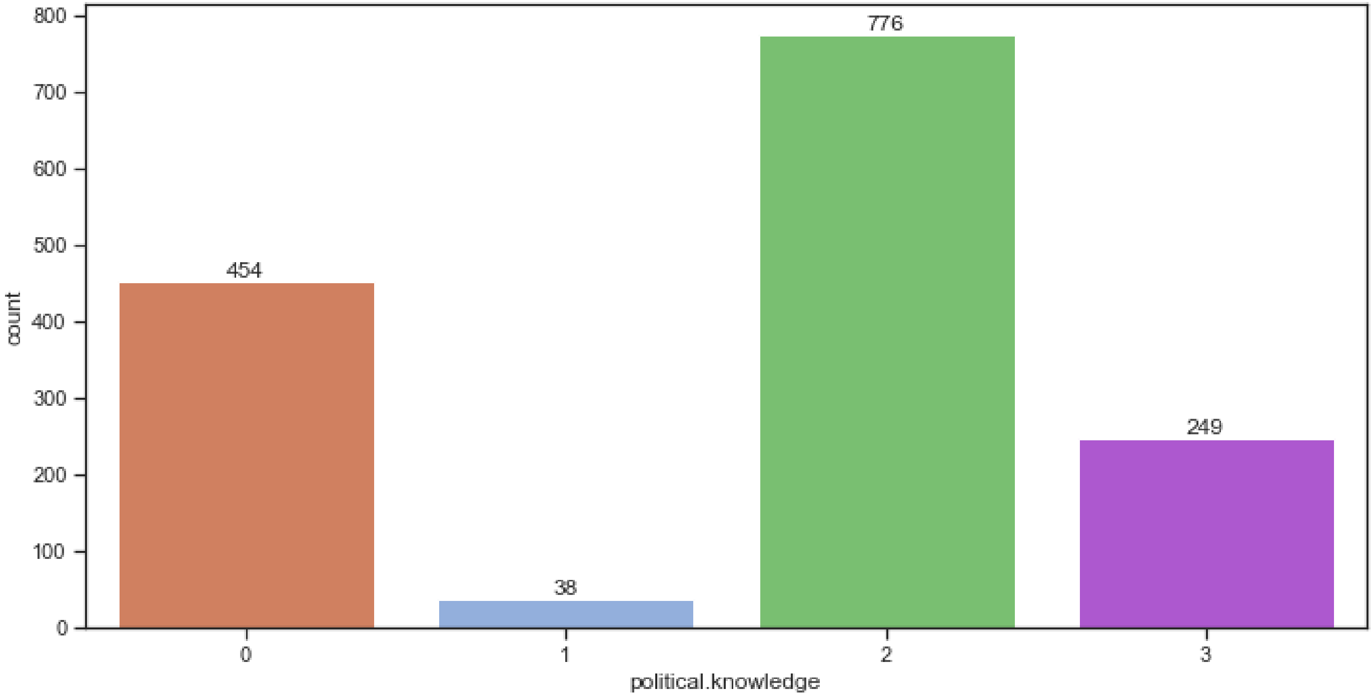
Figure 2: Univariate Analysis of Categorical Variables



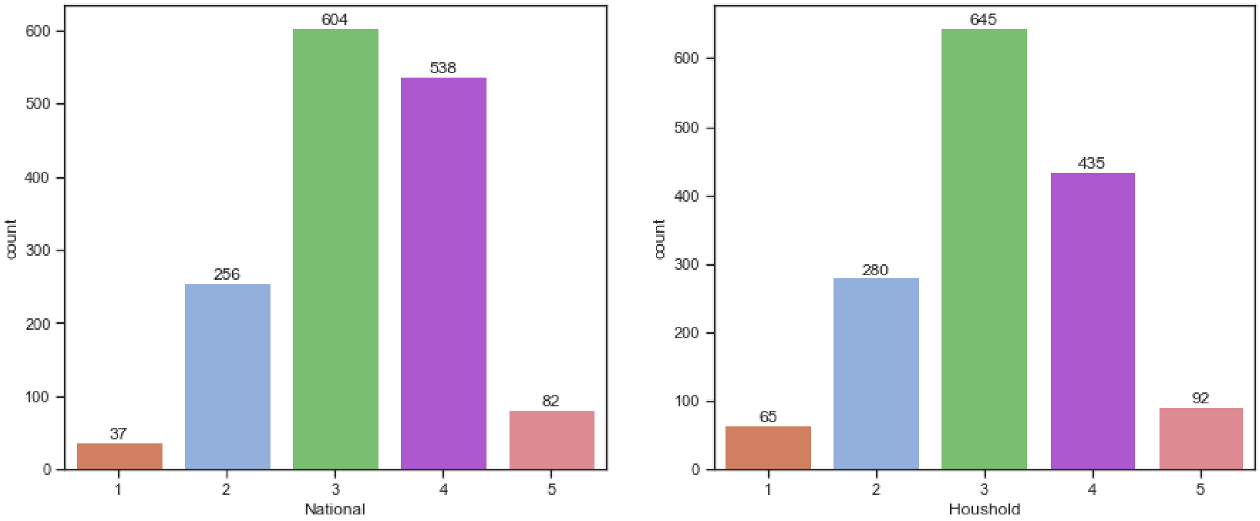
```
Labour          1057
Conservative     460
Name: vote, dtype: int64
```

```
female    808
male      709
```

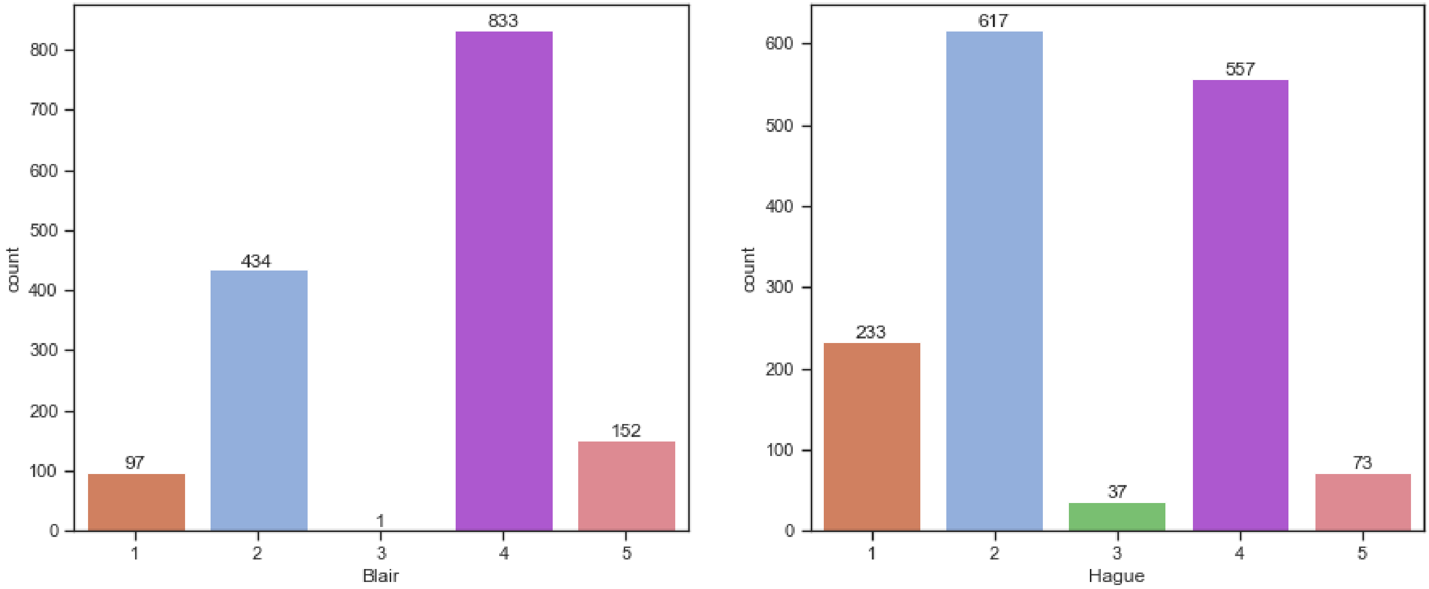
Frequency of Political Knowledge



Frequency of Economic Conditions



Assesments of Leaders



Europian Sentiment Scale

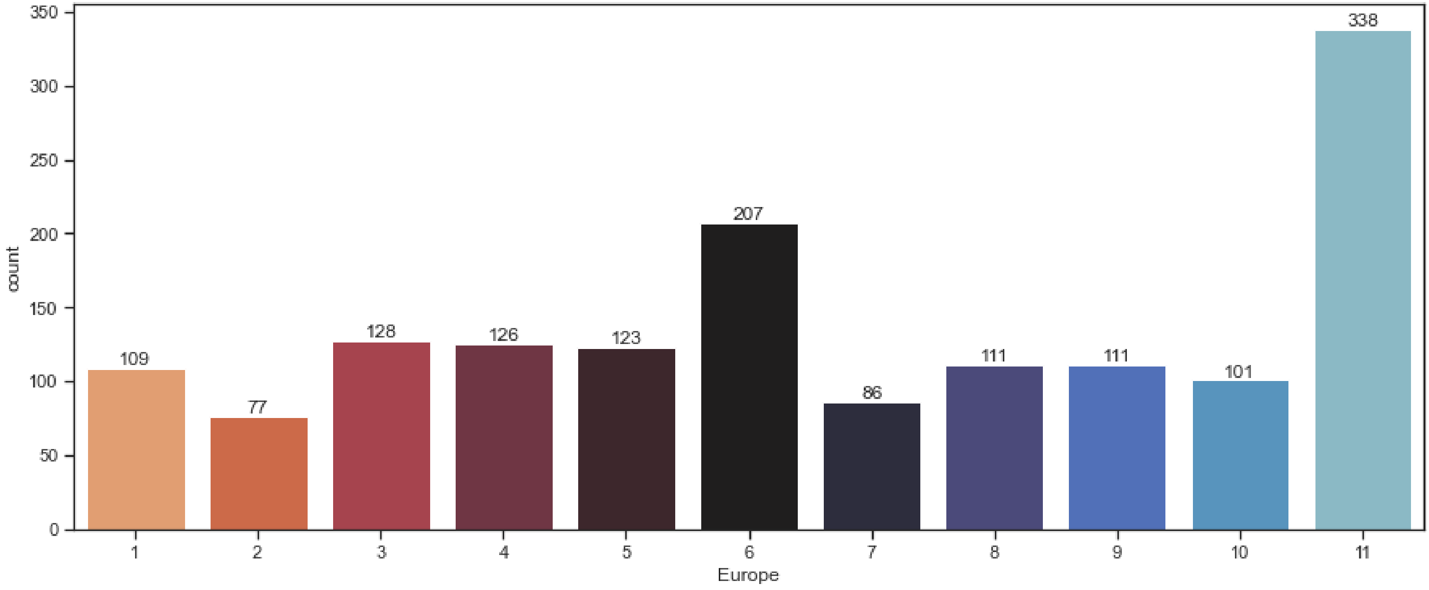
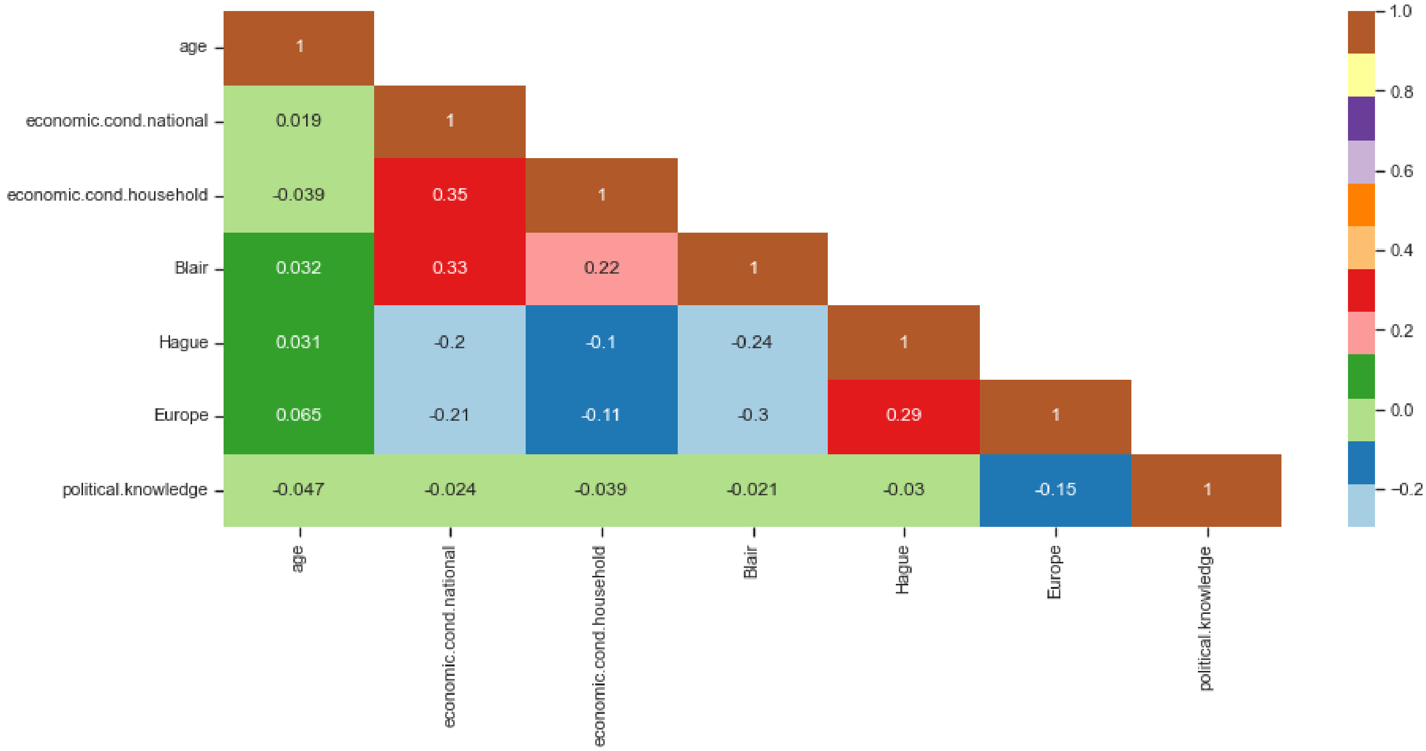
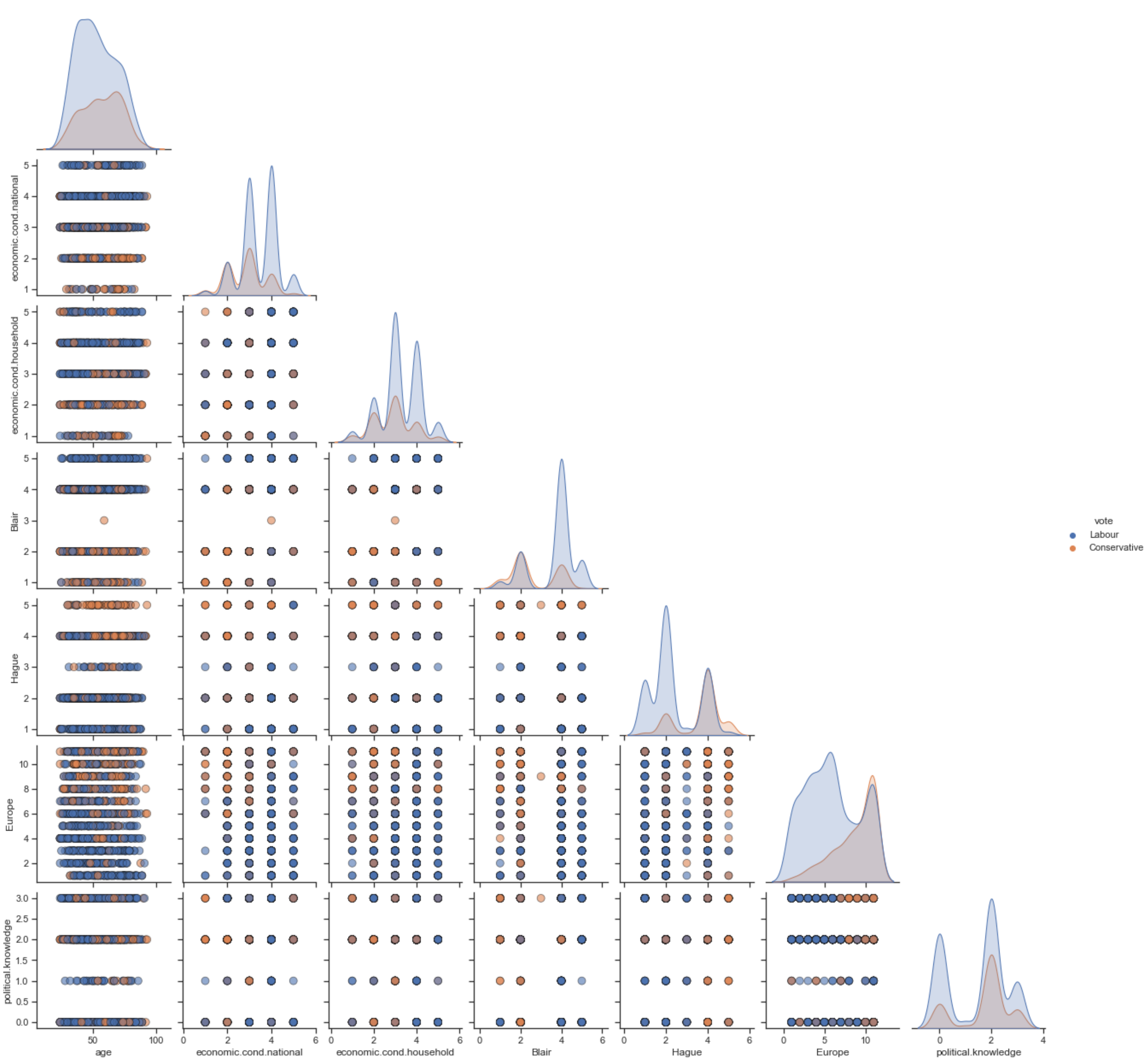


Figure 3: Bi-Variate Analysis

Correlation matrix

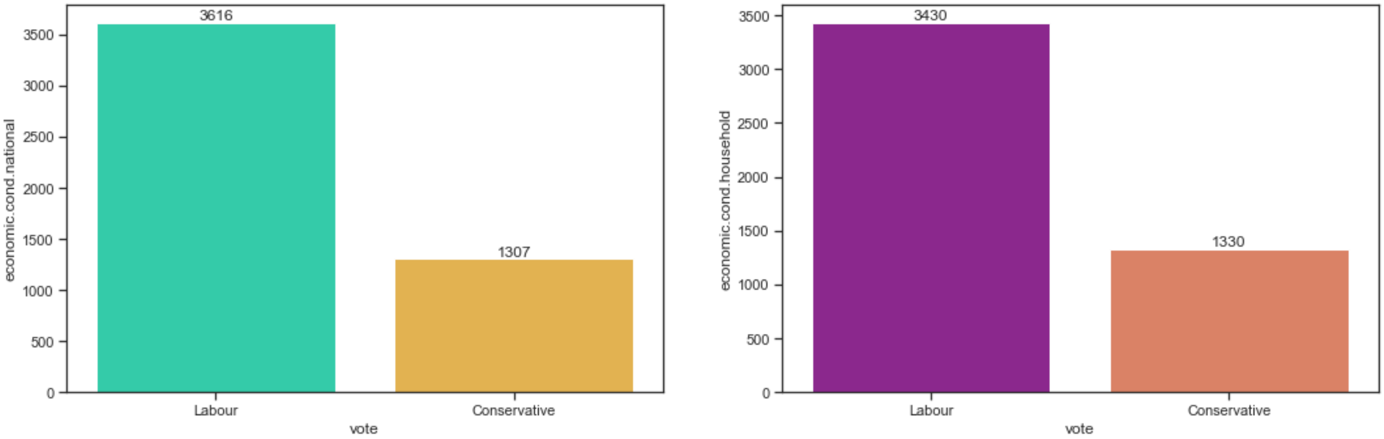


Pairplot Analysis

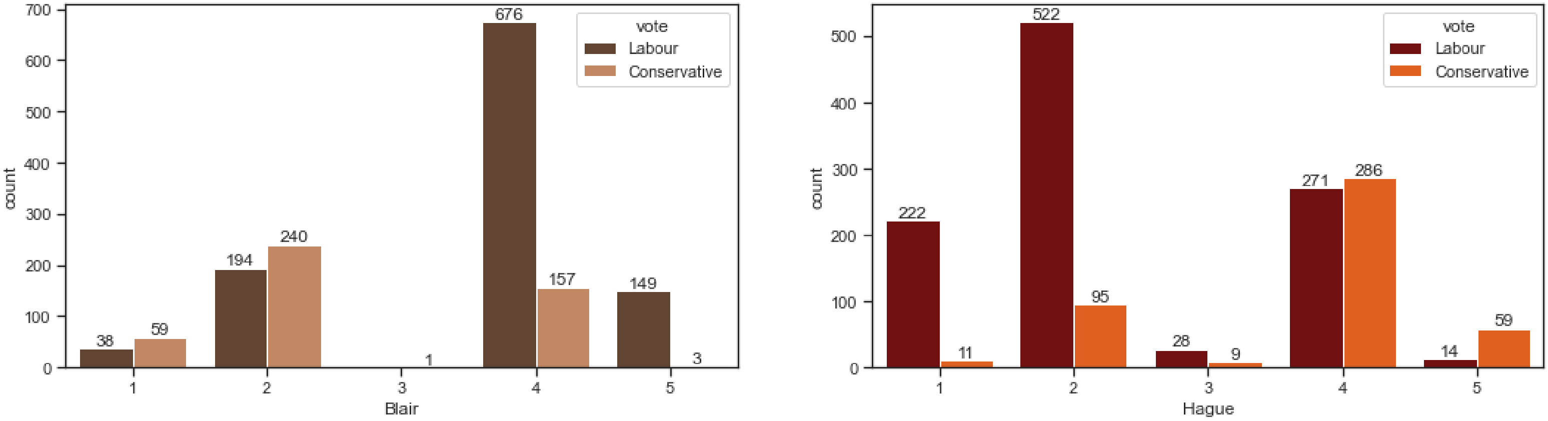


- There is a very less chance of Multicollinearity since many variables does not seem to have strong correlation and linear relation amongst them.

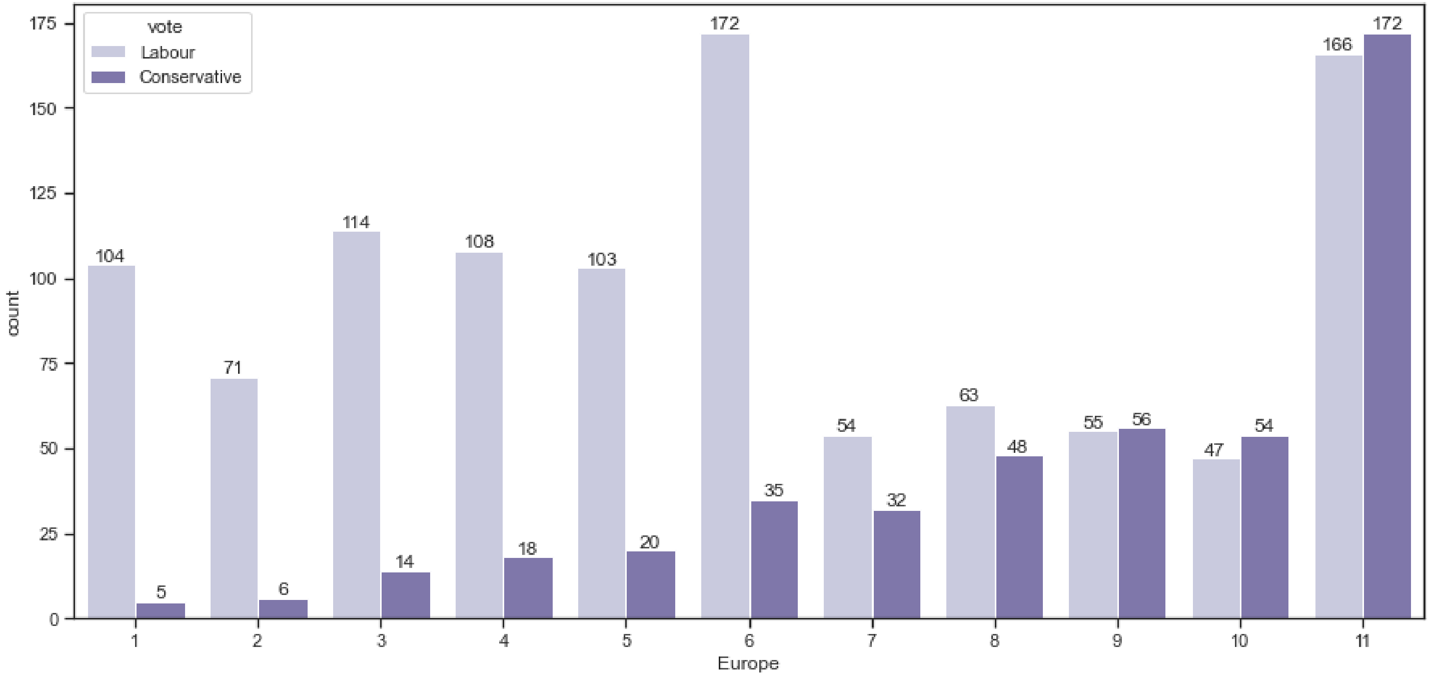
Party Vs. Economic Conditions



Party w.r.t. Assessment



Party w.r.t. European Sentiments



Vote vs. Knowledge scale

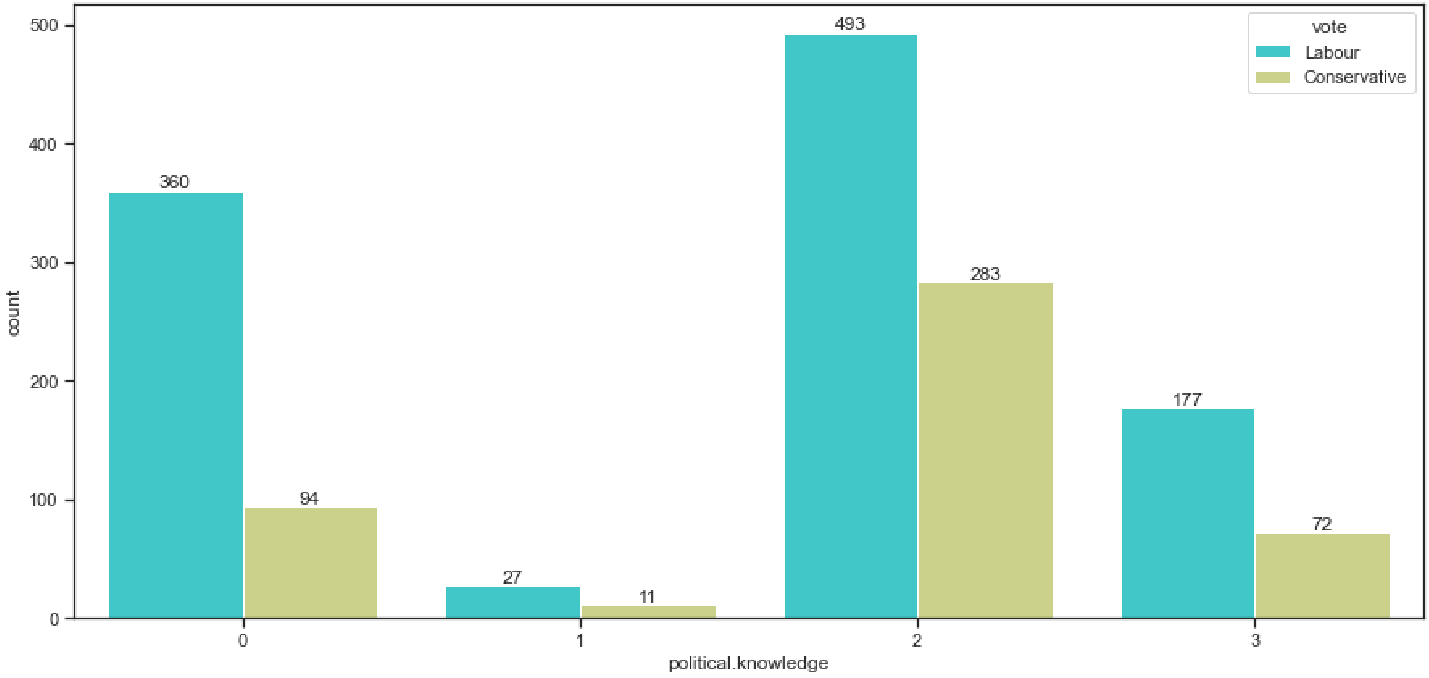
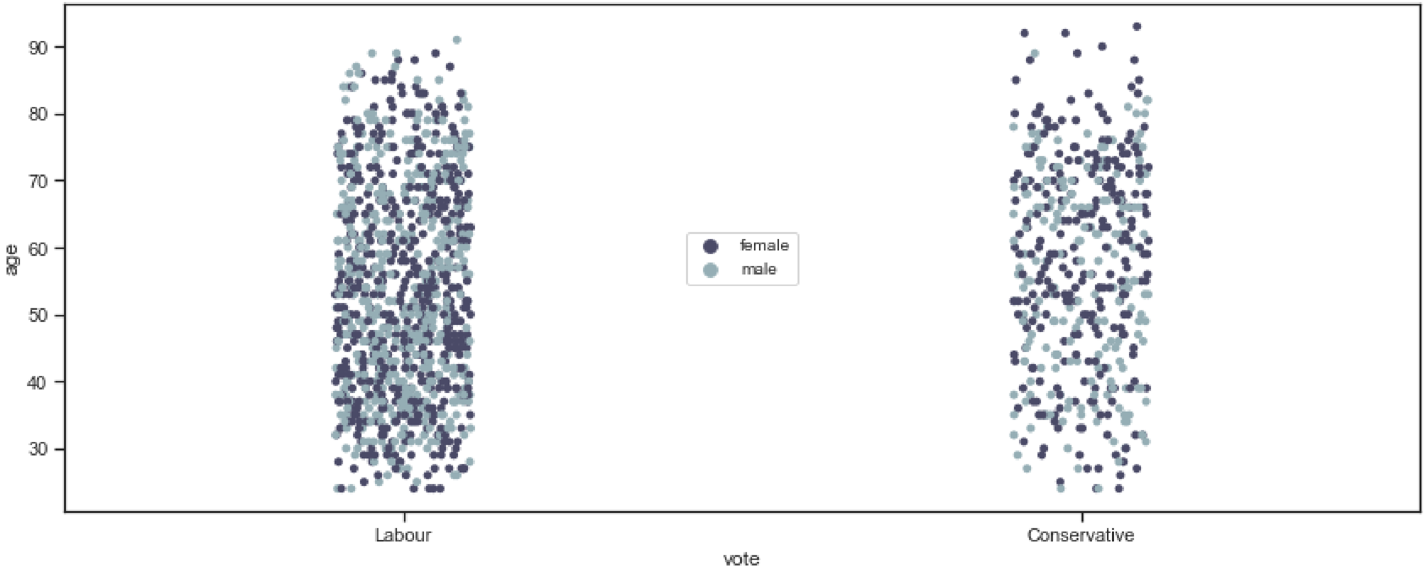


Figure 4: Multivariate Analysis



Inference

- Overall Data seems to be normally distributed with outliers present in economic conditions and majority of the voters lies in the age group of 40-70 years.
- Party choice seems to be more inclined towards Labour (69.7%) with more than double the votes for Conservative class (30.3%).
- Female seems to be have contributed more towards voting with 808 votes i.e. 53.3% of the entire count whereas males seems to be 709 i.e. 46.7%
- For political knowledge, parties seems to uphold good rating of 2 (on a count of 776) however it can be observed that 454 people does not hold any knowledge of parties positions on European market which is 29.93% of the data.
- For both economic conditions i.e. National and Household, majority of people (604 & 645 resp.) have been rated as 3 followed by rating as 4 (538 & 435 resp.).
- Majority of the people has assessed Labour Leader as 4 and Conservative Leader as 2 with an average rating of 3.33 and 2.74 resp.
- Around 338 respondents seems to have be representing 'European Sentiment' with rating as 11 followed by 207 people with rating as 6. Average rating seems to be 6.74.
- There seems to no correlation in the dataset through correlation matrix. Most of the variables such as 'Economic Condition' and 'Blair', 'Economic Condition' and 'Household Condition' as well as 'Hague' and 'Europe' seems to be moderately positively correlated whereas 'Blair' and 'Hague', 'Economic Condition' and 'Europe' seems to be negatively correlated.

- Overall, in every age group, Labour party seems to more votes than Conservative with Female votes are considerably higher than male's.
- For Labour Leader, votes seems to be higher for Labour class and Conservative seems to be more preferred for Conservative Leader with male voters being more in Labour and female voters considerably higher in Conservative. However, out of 152 voters who has given a rating of 5, 149 seems to be from Labour class. Similarly, out of 833 voters with rating of 4, 676 i.e. 81% belongs to class Labour. Nevertheless, lowest rating of 1 seems to coming more from Conservative party. For Labour
- Voters who have accessed National Economic conditions, about 73.45% belongs to Labour party whereas for National Household conditions assessment, about 72% too belongs to Labour party. For National Economic conditions, out of 83 voters with rating of 5, 73 i.e. 87% has voted for Labour party. Similarly, around 83% voters belongs to Labour party with a rating of 4. Rating of 1 and 2 seems to be coming from Conservative Party. Similarly for Household Economic conditions, about 75% of voters with rating of 5 and 80% of voters with rating of 4 belongs to Labour party. Overall, Conservative party seems to be contributing significantly less for accessing Houshold conditions.
- With reference to Eurpoean Sentiment, rating of 10 and 11 seems to be coming from Conservation party whereas rating of 9 seems to be relatively identical for both the parties. Moreover, we can infer that, with less European Sentiment, probability of voters belonging to Labour party are relatively higher.
- Overall, Labour party seems to have more knowledge on European Integration than Conservative. Labour party, on one hand, with an average rating of 1.46 has highest rating of 2 with 493 people followed by 360 people with lowest rating. Conservative, however, seems to have only 72 people with highest rating, 283 people with highest, nonetheless.

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).

Encoding:

Feature 1: Vote

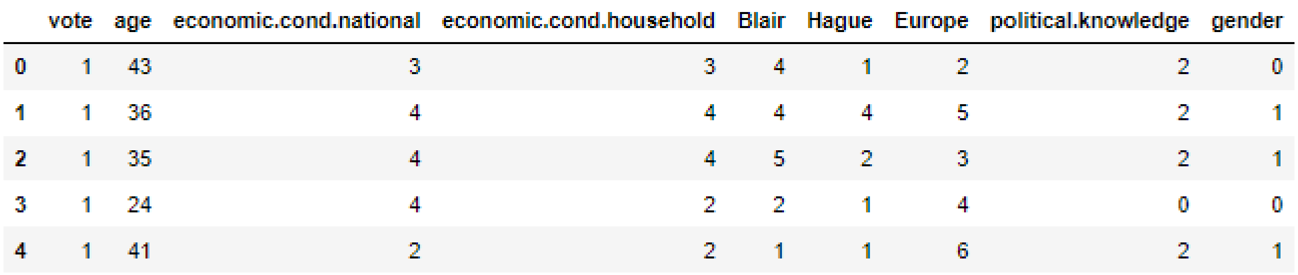
[Labour, Conservative]

Categories: [0: Conservative, 1: Labour]

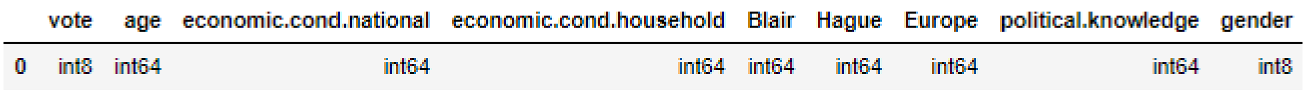
Feature 2: Gender

[Male, Female]

Categories: [0: Female, 1: Male]



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



	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	int8	int64	int64	int64	int64	int64	int64	int64	int8

Scaling

Scaling is a necessary step before you proceed with modelling since most of the ML models utilizes Euclidean Distance, one needs to ensure that there should be none or minimum variance in the magnitudes. Scaling ensures the balance is maintained between the variables and data fits within a scale.

Normalised scaling techniques such as Standard Scaler/Z-Score can be effectively utilized to perform scaling on the features and can be calculated as:

$$Z = \frac{(X - \bar{X})}{S}$$

Where X is the original value of the sample and X-bar is the mean value and S is the standard deviation. Z is the new value which is scaled and will be later stored in the Dataframe using transform.

Table 4: Scaling Comparison

	Mean		Standard Deviation		Variance	
	Before Scaling	After Scaling	Before Scaling	After Scaling	Before Scaling	After Scaling
Age	54.24	0	15.7	1	246.54	1
economic.cond.national	3.25	0	0.88	1	0.78	1
economic.cond.household	3.14	0	0.93	1	0.87	1
Blair	3.34	0	1.17	1	1.38	1
Hague	2.75	0	1.23	1	1.52	1
Europe	6.74	0	3.3	1	10.88	1
political.knowledge	1.54	0	1.08	1	1.18	1
gender	0.47	0	0.5	1	0.25	1

Train-Test-Split

Train_test_split concepts is a preliminary step taken before any model is built. It is effective in building an optimum model which neither underfits nor overfits the data in such a fashion that it partitions the data into trained set and testing set.

Training Set or labelled data is an example given to the model to analyses and learn the patterns in the data which is about **70% of our total data**.

Testing set can be treated as 'unseen data' and is used for building the model. In other words, it is used to test the hypothesis generated by the model. To know the performance of any model, it should be tested on unseen data which is about **30% of our total data**.

Parameters for Train_test_split():

- Test_size – Helps to determine the size of test set.
- Train_size – Represents the proportion of data set in training set.
- Stratify – Split dataset in such a fashion that the ratio of class labels is constant. Specially used when the data is imbalanced.
- Random_state – Controls the shuffling of the dataset before splitting and makes the outcome reproducible with same result.

However, before applying splitting(), we need to ensure that no object type data is present and encoding needs to be applied otherwise as model only accepts integer data type.

Additionally, target variable needs to be dropped from original dataset and stored in a separate variable to use later for testing.

Further, splitting() has been applied keeping **30% as testing dataset** and following variables has been generated:

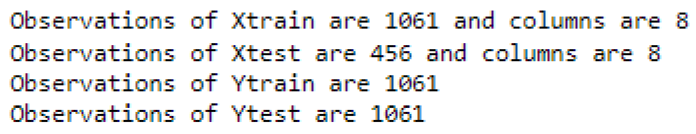
1. Xtrain – Training dataset without target variable

2. Xtest – Testing dataset without target variable

3. Ytrain – Training dataset with target variable which needs to be predicted

4. Ytest - Training dataset with target variable which needs to be predicted

After Train-Test-Split:



```
Observations of Xtrain are 1061 and columns are 8
Observations of Xtest are 456 and columns are 8
Observations of Ytrain are 1061
Observations of Ytest are 1061
```

1.4) Apply Logistic Regression and LDA (Linear Discriminant Analysis). Interpret the inferences of both models. Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting).

Model 1: Logistic Regression

Logistic Regression, also known as 'Logit' and/or Maximum-Entropy classifier, is another method of supervised learning for classification. Unlike Linear Regression, it accepts dependent variable as binary categories, however as integer type.

Logit assigns probabilities to different classes to which a particular data point is likely to belong.

It establishes relation between dependent and independent variables using regression with linear combination as follows:

$$Z = w.x + b$$

Wherein Z is our independent variable, w_i is the optimal weight assigned to an input feature which is determined using **'cross-entropy loss function'** followed by the approach of Gradient Descent, representing how pivotal the feature is for classification.

W is positive - Direct correlation with the class of interest

W is negative – Inverse relation with the class of interest

X being independent variable with b is bias (error).

Parameters used for Logit Model

- **Max_iter** – Defines the maximum number of iterates until convergence during fitting the data.
- **Solver** – Used to identify the parameters weights to minimize the loss function. By default, we are utilizing lib-linear since it is more suitable for smaller datasets.

- **Penalty** - It shrinks the coefficients of those dependent variables which are less contributive toward zero. This helps in optimizing our model for efficiency.
- **Tolerance**: Threshold for the optimization. Lower tolerance has less risk of no convergence of the algorithm.
- **Random-State:** Allows an algorithm to run multiple times with same output. Takes any integer user input.

Step 1: Calling Logit model and fitting the data

Logit model has been called using the tuning parameters fitting where loss function and sigmoid has been utilized to form best s-curve:

Random_state:	1
Max_Iter:	100
Solver:	Lib-Linear
Penalty:	L2
Tol:	0.001

Step 2: Checking Accuracy Scores

After applying sigmoid and cross-entropy to fit the training data, following scores can be observed:

```
Accuracy for Trained Logit Model is: 0.83
Accuracy for Test Logit Model is:    0.86
```

Model 2 – Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is another classification method used in Supervised Learning technique to predict observations where classes are **known and fixed.** It also utilizes approach of forming a linear combination and observing data from low-dimensional space by maximizing the between class scatter and minimize class scatter.

LDA constructs linear equation which minimizes the possibility of misclassification of cases into their respective classes i.e. building a best fit line which separates the 0 and 1 with least chances of classes being wrongly classified to their respective class.

It can be also be used as a dimensionality reduction technique by focusing on maximizing the separability of known classes in the target variable.

Calling LDA model and fitting into trained data

LDA model has been called using the below tuning parameters to form a best line for separating probabilities:

- Solver = ‘svd’ i.e. Single Value Decomposition. As it does not compute the covariance matrix, it is recommended for data with many features.
- Tol = 0.001
- Training class prediction with a cut-off of 0.5
- Testing class prediction with a cut-off of 0.5

Checking Accuracy Scores

```
Accuracy for Trained LDA Model is:   0.82
Accuracy for Test LDA Model is:      0.85
```

Inference

- Overall, Logistic regression seems to be performing better than LDA.
- Trained accuracy of Logit stands at 83% and LDA at 82% which suggests that model neither underfits nor overfits. However, test scores seems to have improved for both models which suggests that model has performed well and able to capture test dataset based on training model.

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

KNN Model

KNN stands for K-Nearest Neighbor is a non-parametric and lazy learning algorithm effective with data of large data points and less dimensions.

In KNN, term ‘k’ is a parameter which refers to the number of nearest neighbors.

Classifies predictor data point by building on a tree through separating axes based on median. Using distance measure techniques henceforth, it determines the neighbors in the dataset closest to predictor data point thus allocating the class.

Identifying the ‘K’ value

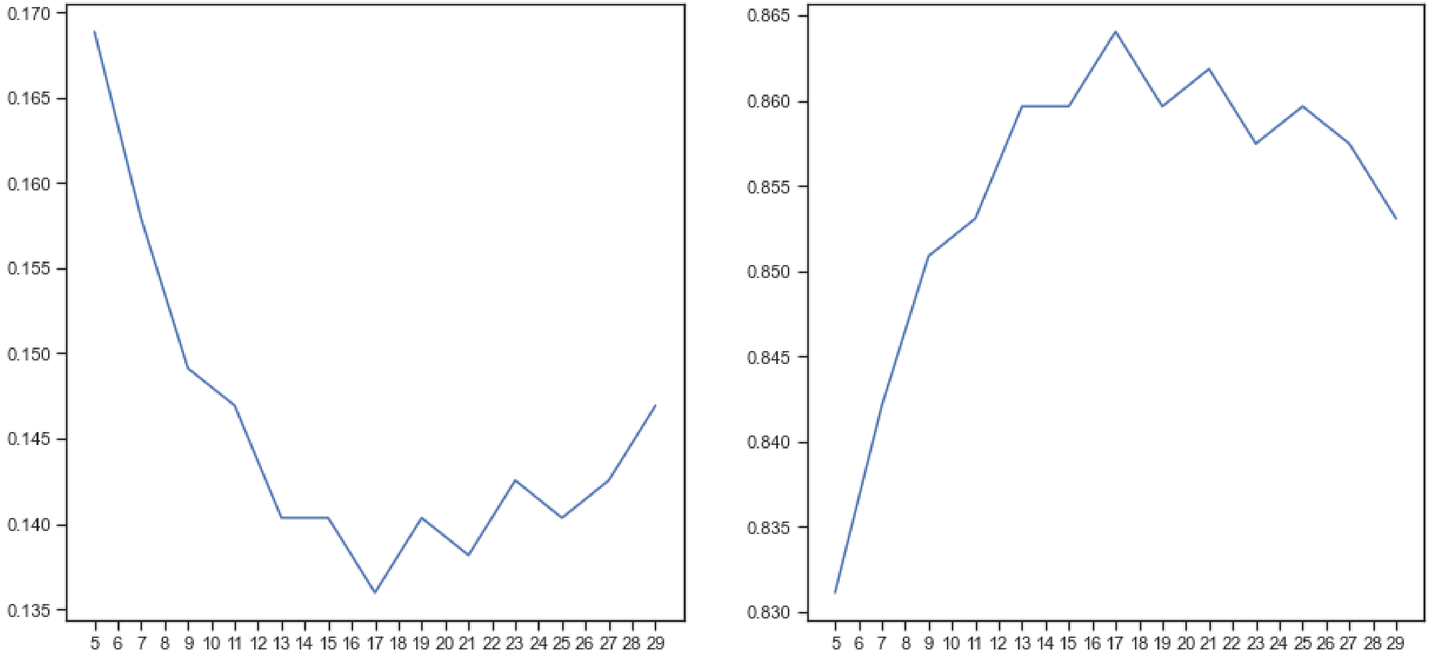
Identifying K-value with high accuracy and low misclassification error can help us find an optimal value of K.

Using too small value might overfit the training data and large value will train model as unferfit.

```
For K= 5

Testing Scores are: 0.831140350877193
Classification error are: 0.16885964912280704


For K= 7

Testing Scores are: 0.8421052631578947
Classification error are: 0.1578947368421053


For K= 9

Testing Scores are: 0.8508771929824561
Classification error are: 0.14912280701754388


For K= 11
```

```
Testing Scores are: 0.8530701754385965
Classification error are: 0.14692982456140347


For K= 13

Testing Scores are: 0.8596491228070176
Classification error are: 0.14035087719298245


For K= 15

Testing Scores are: 0.8596491228070176
Classification error are: 0.14035087719298245


For K= 17

Testing Scores are: 0.8640350877192983
Classification error are: 0.13596491228070173


For K= 19

Testing Scores are: 0.8596491228070176
Classification error are: 0.14035087719298245


For K= 21

Testing Scores are: 0.8618421052631579
Classification error are: 0.13815789473684215


For K= 23

Testing Scores are: 0.8574561403508771
Classification error are: 0.14254385964912286


For K= 25

Testing Scores are: 0.8596491228070176
Classification error are: 0.14035087719298245


For K= 27

Testing Scores are: 0.8574561403508771
Classification error are: 0.14254385964912286


For K= 29

Testing Scores are: 0.8530701754385965
Classification error are: 0.14692982456140347
```

Figure 5: Classification Error and Scoring Plot



[At k = 17, we can observe highest scores i.e. 86% approx. with almost 14% error post of which data seems to be dropping, hence, we can choose optimal value as 17.]

Building KNN Classifer:

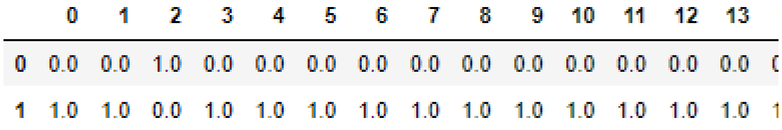
Following parameters has been used to build classifier:

1. **Metric:** Euclidean – Preferred distance metric due to not much Multicollinearity between variables.
2. **N_neighbors :** 17
3. **Weights:** Distance – KNN works well with choosing nearest neighbor (data point) through their distance. Other method would be 'uniform'.
4. **N_jobs:** 1

After fitting the model, we can observe:

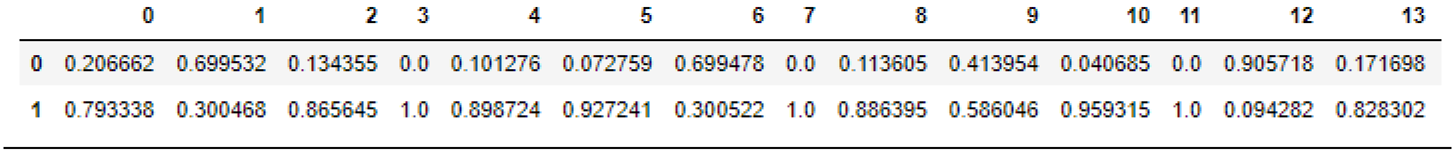
```
Accuracy for Trained KNN Model is:   1.0
Accuracy for Test KNN Model is:      0.86
```

Table 5: Probability of Training set in KNN



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	(
1	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1

Table 6: Probability of Testing set in KNN



	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0.206662	0.699532	0.134355	0.0	0.101276	0.072759	0.699478	0.0	0.113605	0.413954	0.040685	0.0	0.905718	0.171698
1	0.793338	0.300468	0.865645	1.0	0.898724	0.927241	0.300522	1.0	0.886395	0.586046	0.959315	1.0	0.094282	0.828302

Model 2 – Gaussian Naïve-Bayes

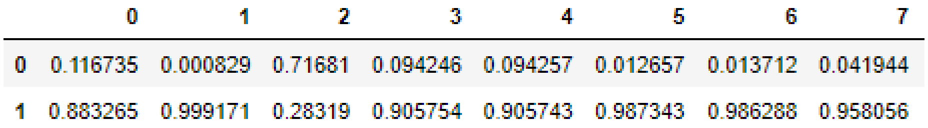
Gaussain Naïve-Bayes (NB) is commonly used as a benchmark model and is a probabilistic model based on Bayes Theorem where 'Naïve' means that there is an assumption that features in the dataset are mutually independent. However, it tends to perform well even if there is a dependency.

Supports continuous values well and does very well with noisy and missing data.

After building the model and fitting the trained data, we can observe:

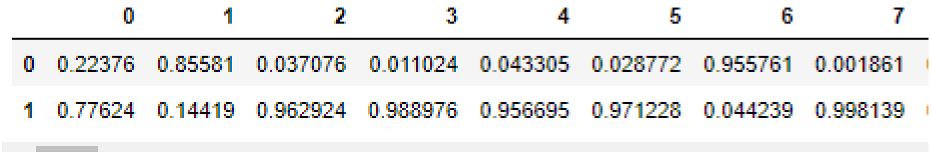
```
Accuracy for Trained GB Model is:    82.0
Accuracy for Test GB Model is:       86.0
```

Table 7: Probability of Training set in Gaussian



	0	1	2	3	4	5	6	7
0	0.116735	0.000829	0.71681	0.094246	0.094257	0.012657	0.013712	0.041944
1	0.883265	0.999171	0.28319	0.905754	0.905743	0.987343	0.986288	0.958056

Table 8: Probability of Testing set in Gaussian



	0	1	2	3	4	5	6	7
0	0.22376	0.85581	0.037076	0.011024	0.043305	0.028772	0.955761	0.001861
1	0.77624	0.14419	0.962924	0.988976	0.956695	0.971228	0.044239	0.998139

Inference

- KNN model seems to be overfitted with 100 % accuracy on the train data and 86% on the test data signifying a gap of more than 10% hence overfit.

- Gaussian seems to be neither overfit nor underfitted with an accuracy of 82% for trained and 86% for test data.

- K = 17 seems to be a optimal value of neighbors for modelling however GridSearch can be applied to choose between different k-values with high accuracy yet high MCE too.

- Probability of an observation belonging to class 1 is way more than class 0 for Gaussian NB.

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

Grid Search Cross-Validation (GridSearchCV()) comes in handy to perform tuning by doing an exhaustive search over specified parameter values for an estimator. It helps to determine which parameters are best suited to perform a model with efficient results by providing a list of different range of values for associated tuning parameters.

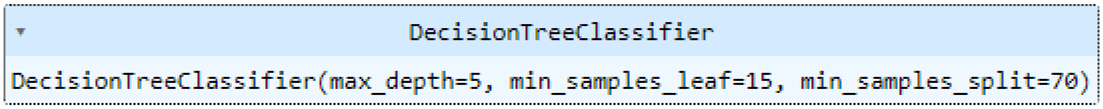
Note – Values provided inside lists can be relooked in case testing models generates less effective results.

Model 1 – Tuned Decision Tree Classifier

Tuning Parameters used in GridSearchCV for DT Model:

1. **Max-Depth –** Helps in uniform the branching and reach an optimum terminal nodes. Optimal numbers depends to business-to-business and can be chosen based on Gini-Gain and sample size of a particular node.
2. **Min-Sample-Leaf** - Ensures minimum number of observations in a particular child node after splitting is done thus may have an effect of smoothing the model. Generally, **1-3%** is selected at random out of total dataset as threshold values for splitting.
3. **Min-Sample-Split –** Ensures minimum number of observations in a particular child node before splitting is done and values are generally **three-times** the values of min-sample-leaf.
4. **Cross-validation (CV) -** Helps in avoiding over-fitting and greedy nature of model by following a k-fold cross validation wherein K = number of times we want to run a model via different iterations.

After applying best_estimator() with **cv = 7**, following values has been obtained:



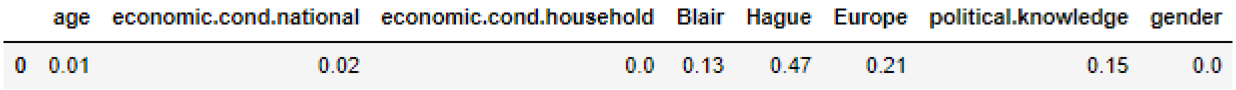
```
▾ DecisionTreeClassifier
DecisionTreeClassifier(max_depth=5, min_samples_leaf=15, min_samples_split=70)
```

Accuracy

```
Accuracy for Tuned Trained DT Model is:     82.0
Accuracy for Tuned Test DT Model is:        84.0
```

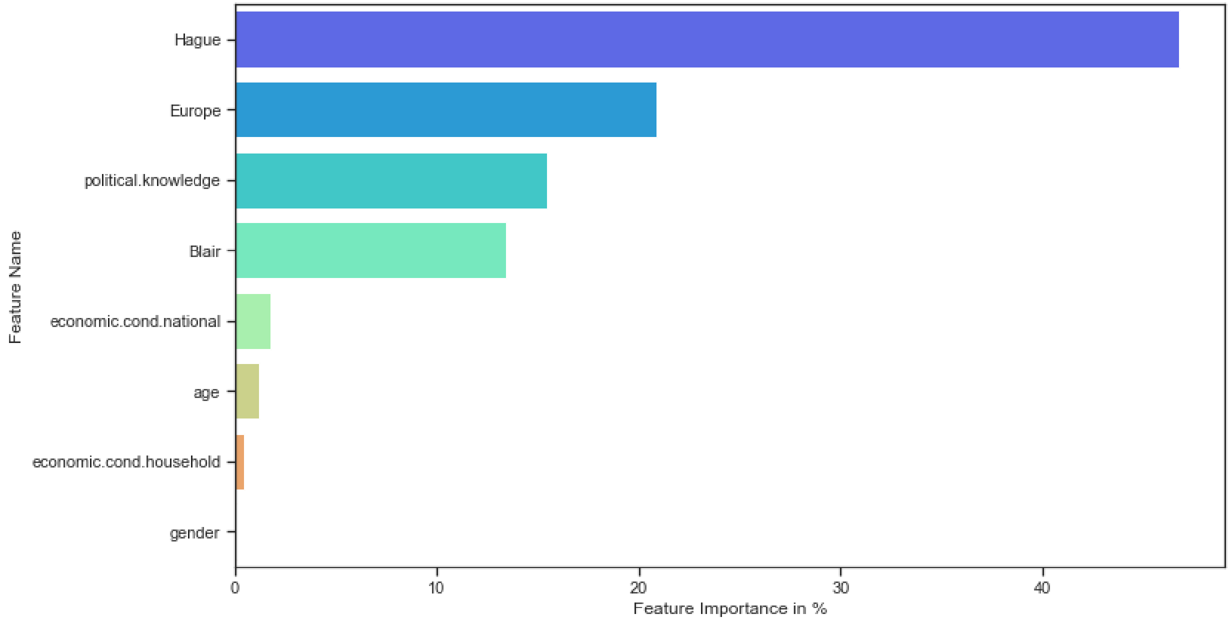
Feature Importance

Table 9: Feature Importance of DT Model



	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	0.01	0.02	0.0	0.13	0.47	0.21	0.15	0.0

Figure 6: Feature Importance of DT



- **Hague, Europe sentiment, Political Knowledge and Blair seems to be contributing the most in building the model and predicting Party.**

Model 2 – Tuned Bagging Random Forest Classifier

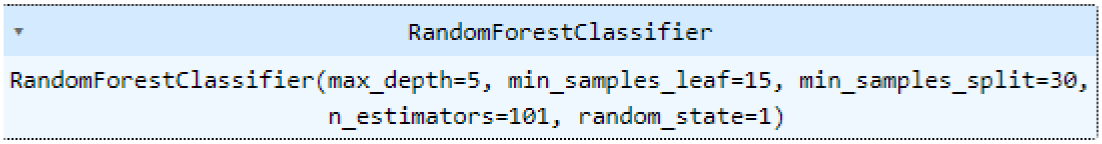
Random Forest is an Ensemble Machine Learning Bagging technique that constructs several decision trees on training set and combines it into one class that is the mean prediction of the individual trees.

Parameters used for RF model:

- max_depth
- min_samples_leaf
- min_samples_split
- n_estimators = number of trees to be build. (preferred to take sq. root of number of observations)
- random_state

- cross_validation (CV)

After applying best_estimator() with **cv = 7**, following values has been obtained:



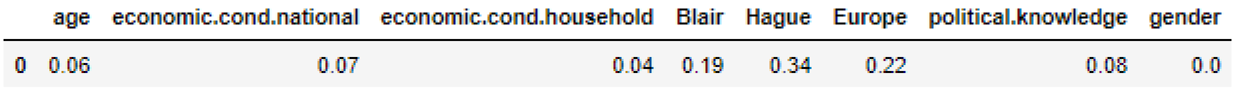
```
RandomForestClassifier

RandomForestClassifier(max_depth=5, min_samples_leaf=15, min_samples_split=30,
                       n_estimators=101, random_state=1)
```

Accuracy

```
Accuracy for Tuned Trained RFModel is:      0.835
Accuracy for Tuned Test RF Model is:        0.86
```

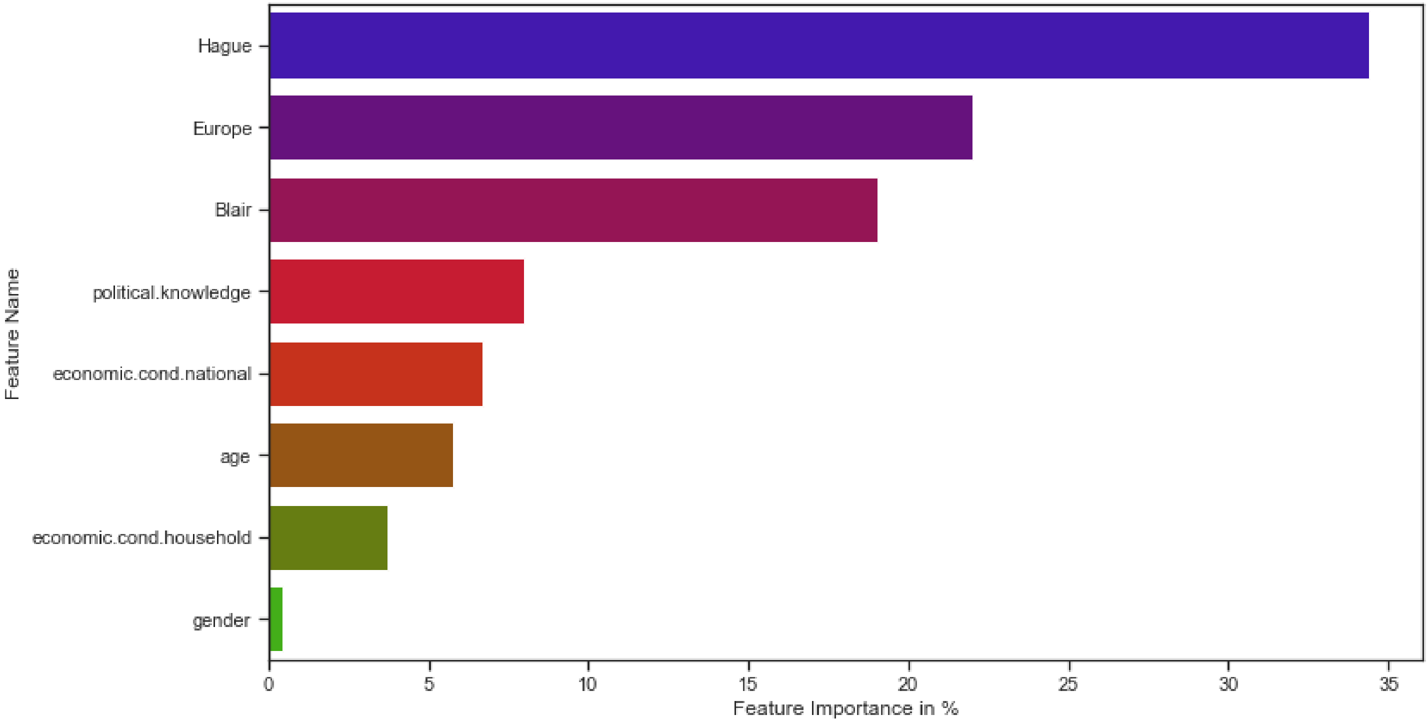
Feature Importance

Table 10: Feature Importance of RF Model



	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	0.06	0.07	0.04	0.19	0.34	0.22	0.08	0.0

Figure 7: Feature Importance of RF Classifier



- **Hague, Europe, Blair, Political Knowledge seems to be most imperative for RF modelling.**

Model 3 – Boosting

Boosting is another Ensemble 'Sequential' technique that consecutively builds on weak models to generate one stronger leaner by minimizing the errors from previous model made during prediction.

ADA Boosting

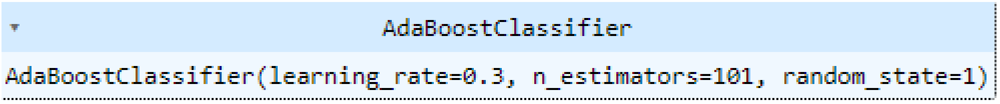
Known as Adaptive Boosting, this technique takes previous model and provides over-weightage to the predictor data points with more significance that has been left.

By default method takes simple model to perform best or input needs to be provided as same. Complex model might affect the performance on test data

Parameters used for building Boosting model:

- **Algorithm**: SAMME.R - The SAMME.R algorithm typically converges faster than SAMME, achieving a lower test error with fewer boosting iterations.
- **Learning Rate**: A higher learning rate increases the contribution of each classifier.
- **N-Estimators:** number of trees to be build. (preferred to take sq. root of number of observations).

After applying GridSearchCV with **cv = 7**, we can observe the below final values:



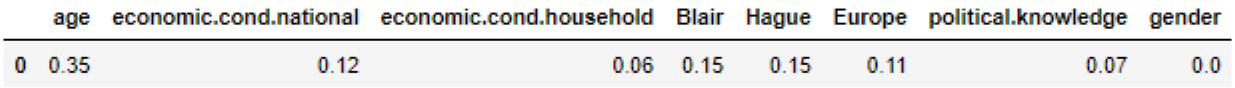
```
AdaBoostClassifier
AdaBoostClassifier(learning_rate=0.3, n_estimators=101, random_state=1)
```

Accuracy

```
Accuracy for Trained AdaBoosting Model is:  0.84
Accuracy for Test AdaBoosting Model is:     0.84
```

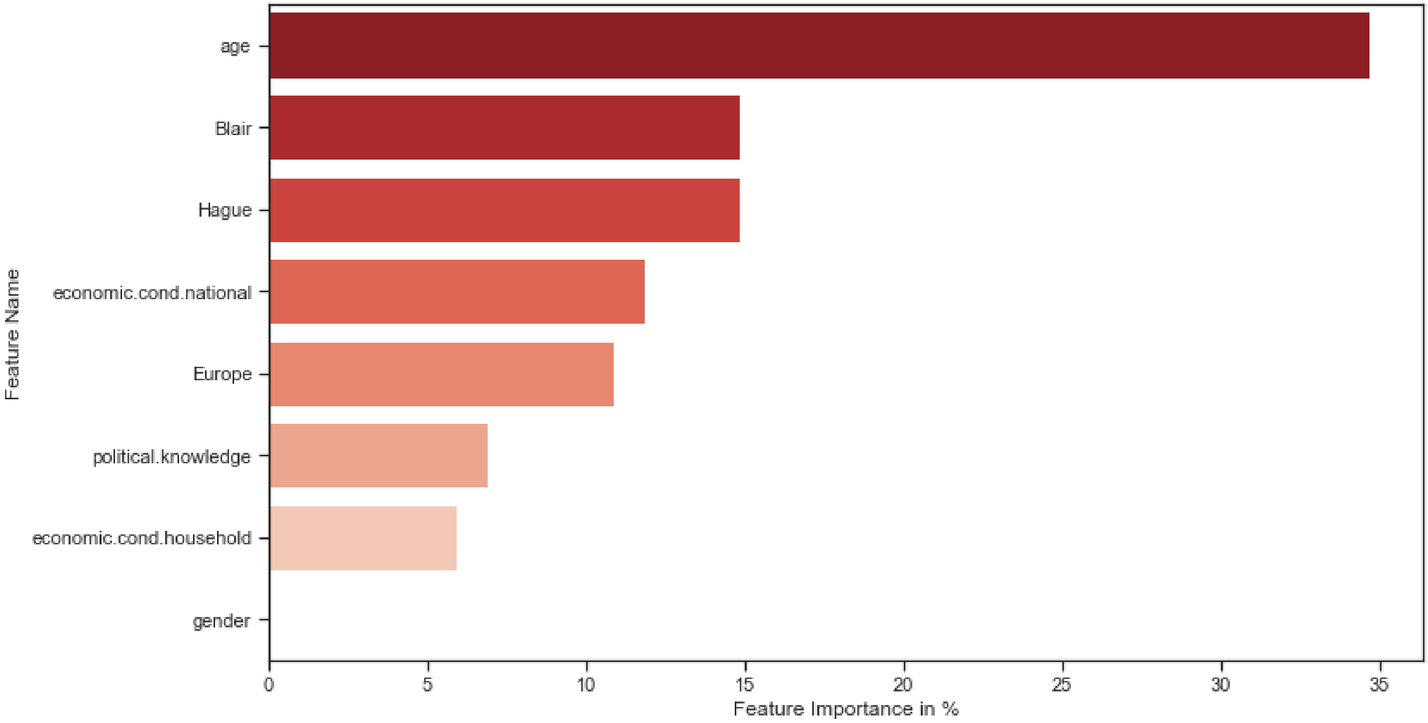
Feature Importance

Table 11: Feature Importance of Ada Boost Classifier



	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	0.35	0.12	0.06	0.15	0.15	0.11	0.07	0.0

Figure 8: Feature Importance of Ada Boosting Classifier



- **Age, which has contributed much lower in RF, Logistic and LDA, can be seen to have improved in Ada Boosting.**

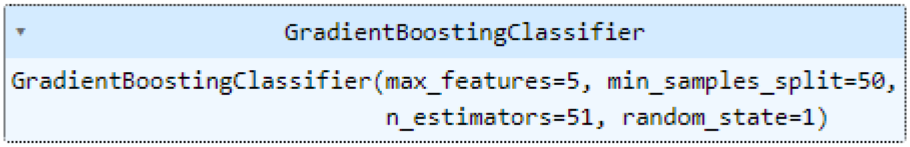
Gradient Boosting

This technique focus on the residual (Predicted – Actual point) from the previous model and works on improving the efficiency thus. However, this type of boosting is more suited for Regression based models with continuous nature where we have to average the best outcomes to generate one strong learner model.

Parameters used for building Boosting model:

- **Loss:** Estimating how good the model is at making predictions with minimal error. 'log_loss' refers to binomial and multinomial deviance, the same as used in logistic regression. It is a good choice for classification.
- **Criteria:** To measure the quality of the split. For Gradient Boosting, ‘friedman_mse’ can be used often for best approximation.
- **Learning Rate**: A higher learning rate increases the contribution of each classifier.
- **N-Estimators:** number of trees to be build. (preferred to take sq. root of number of observations).
- **Max_Features**
- **Min_sample_split**

After applying GridSearchCV with **cv = 7**, we can observe the below final values:



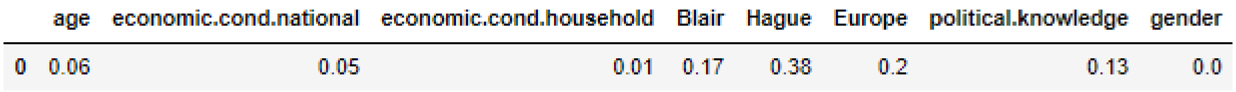
```
GradientBoostingClassifier
GradientBoostingClassifier(max_features=5, min_samples_split=50,
                           n_estimators=51, random_state=1)
```

Accuracy

```
Accuracy for Trained AdaBoosting Model is:  0.87
Accuracy for Test AdaBoosting Model is:     0.85
```

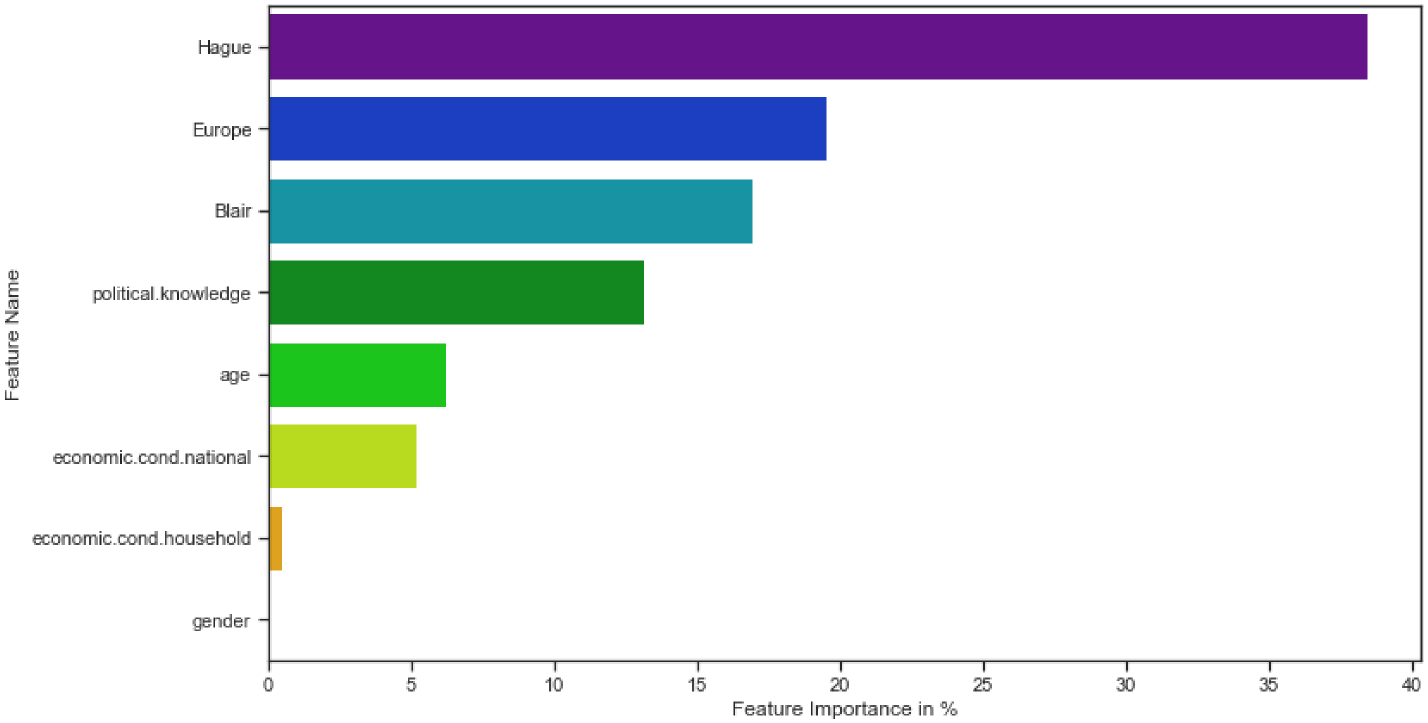
Feature Importance

Table 12: Feature Importance of Gradient Boost Classifier



	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	0.06	0.05	0.01	0.17	0.38	0.2	0.13	0.0

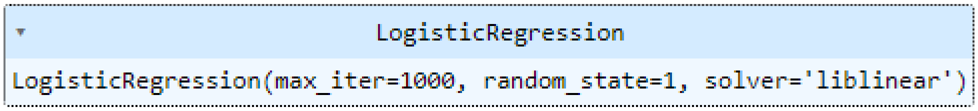
Figure 9: Feature Importance of Gradient Boost Classifier



- **Hague, Europe, Blair and political knowledge still seems to be pivotal for making predictions on party.**
- **Overall Accuracy on the trained data seems to have improved using Boosting.**

Tuned Logistic Model

After applying GridSearchCV with cv = 7, output has been observed:



```
LogisticRegression
LogisticRegression(max_iter=1000, random_state=1, solver='liblinear')
```

Accuracy

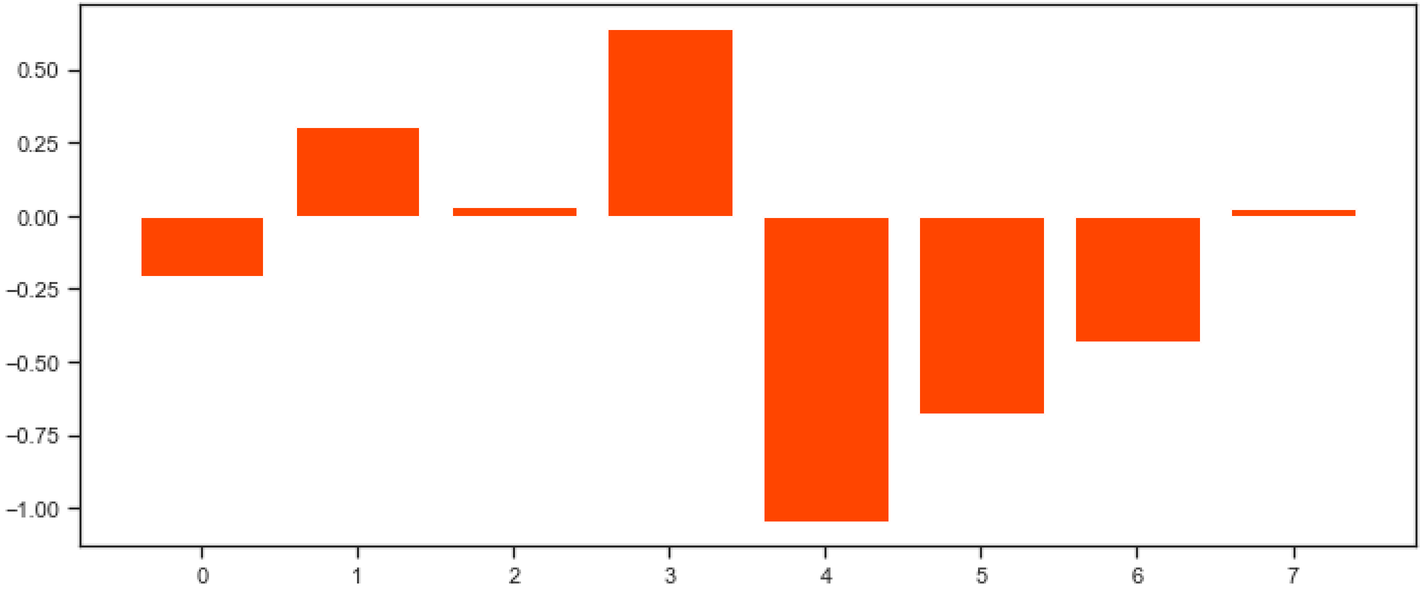
```
Accuracy for Tuned Trained Logit Model is:  0.83
Accuracy for Tuned Test Logit Model is:     0.86
```

Intercept

```
Intercept for Logit Model is: [1.30480547]
```

Feature Importance

Figure 10: Coefficient Importance of Logit Model

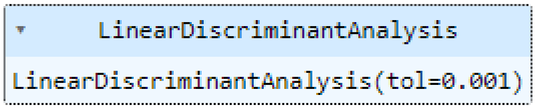


```
Feature: 0, Score: -0.21
Feature: 1, Score: 0.31
Feature: 2, Score: 0.03
Feature: 3, Score: 0.64
Feature: 4, Score: -1.05
Feature: 5, Score: -0.68
Feature: 6, Score: -0.43
Feature: 7, Score: 0.02
```

- **The trained accuracy has slightly increased after tuning the data however testing data remains the same.**
- **Regression co-efficients shows the feature importance and features such as economic.cond.national with highest co-efficiency seems to have positive effect on the target variable as one increases, other increases too.**

Tuned LDA Model

After applying GridSearchCV with cv = 7, output has been observed:



```
▾ LinearDiscriminantAnalysis
LinearDiscriminantAnalysis(tol=0.001)
```

Accuracy

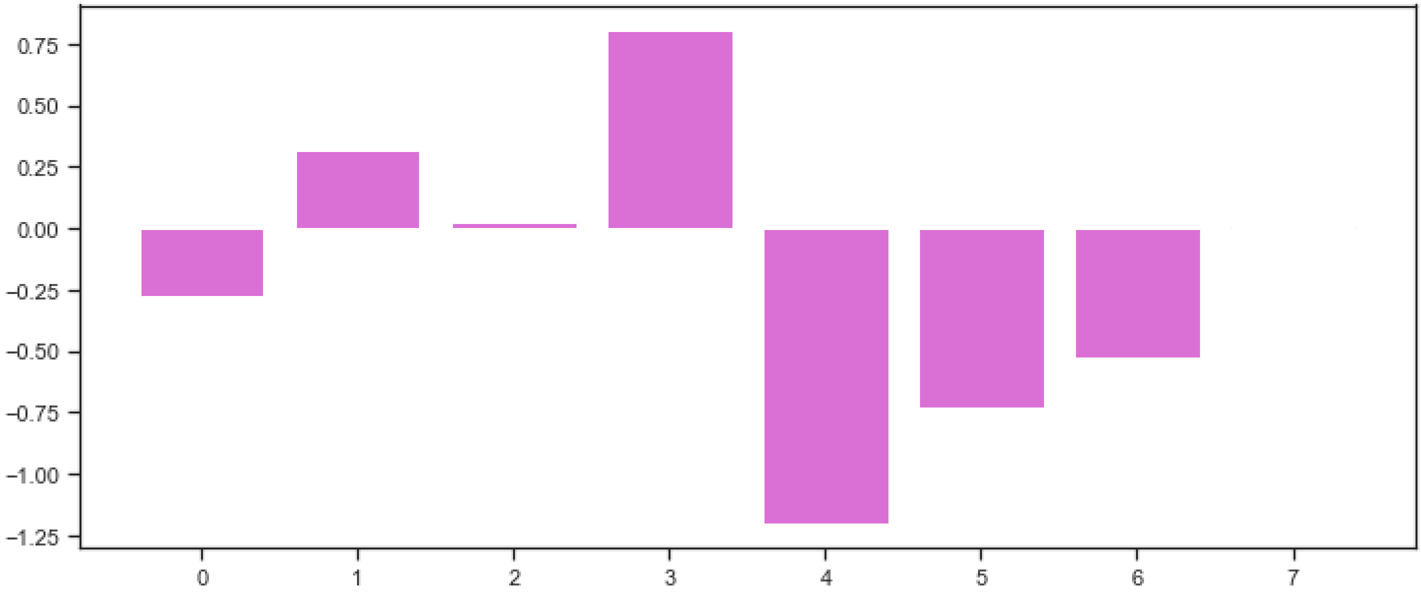
```
Accuracy for Tuned Trained Logit Model is:  0.82
Accuracy for Tuned Test Logit Model is:     0.85
```

Intercept

```
Intercept for LDA Model is: [1.38940367]
```

Feature Importance

Figure 11: Coefficient Importance of LDA Model

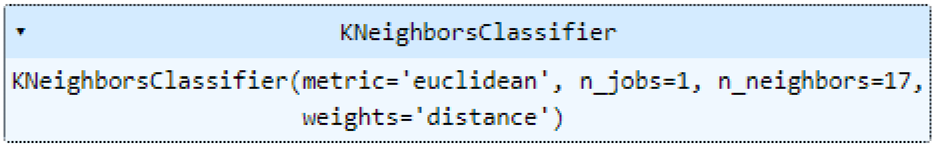


```
Feature: 0, Score: -0.27
Feature: 1, Score: 0.32
Feature: 2, Score: 0.03
Feature: 3, Score: 0.81
Feature: 4, Score: -1.20
Feature: 5, Score: -0.73
Feature: 6, Score: -0.52
Feature: 7, Score: 0.01
```

- **Model doesn’t seems to have improved post tuning as accuracy remains the same.**

Tuned KNN Model

Best-Estimators:



```
▾ KNeighborsClassifier
KNeighborsClassifier(metric='euclidean', n_jobs=1, n_neighbors=17,
                     weights='distance')
```

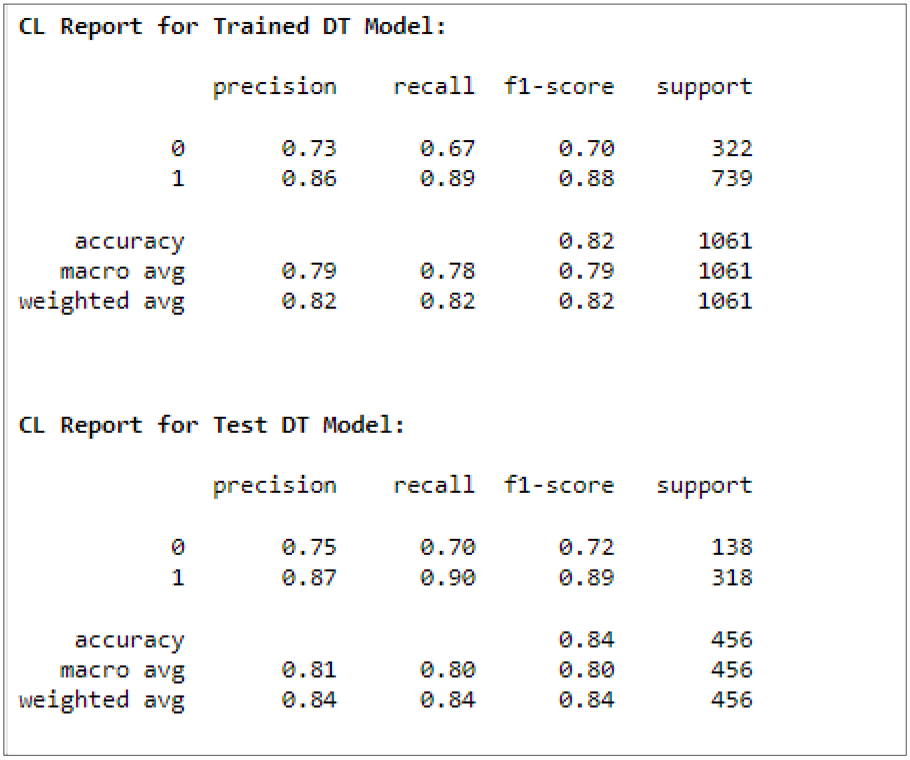
Accuracy

```
Accuracy for Tuned Trained KNN Model is:    1.0
Accuracy for Tuned Test KNN Model is:       0.86
```

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.

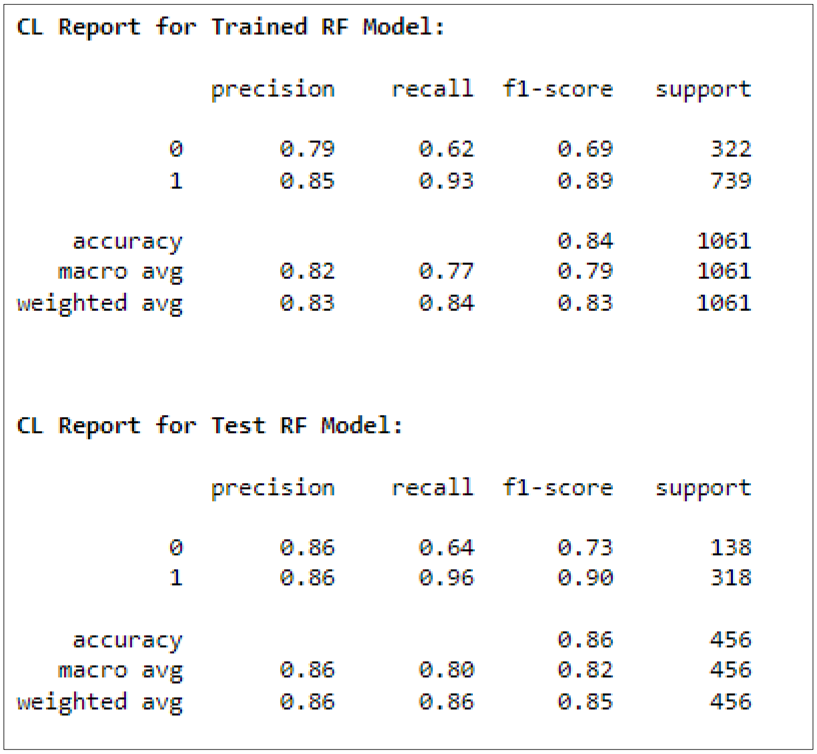
Classification Report

Table 13: Classification Report for DT



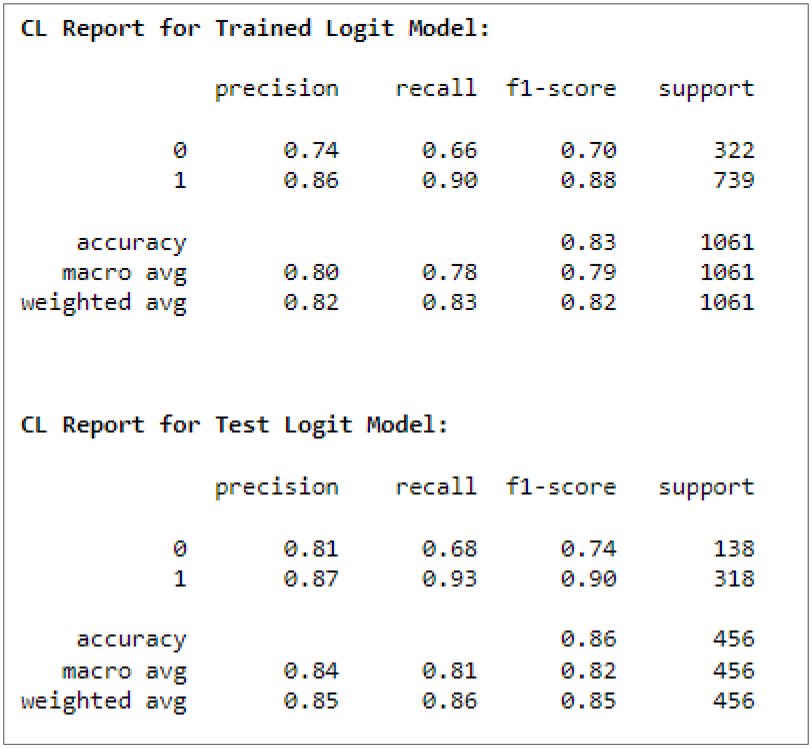
```
CL Report for Trained DT Model:

              precision    recall  f1-score   support

           0       0.73      0.67      0.70       322
           1       0.86      0.89      0.88       739

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



CL Report for Test DT Model:

              precision    recall  f1-score   support

           0       0.75      0.70      0.72       138
           1       0.87      0.90      0.89       318

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

Table 14: Classification Report for RF Classifier



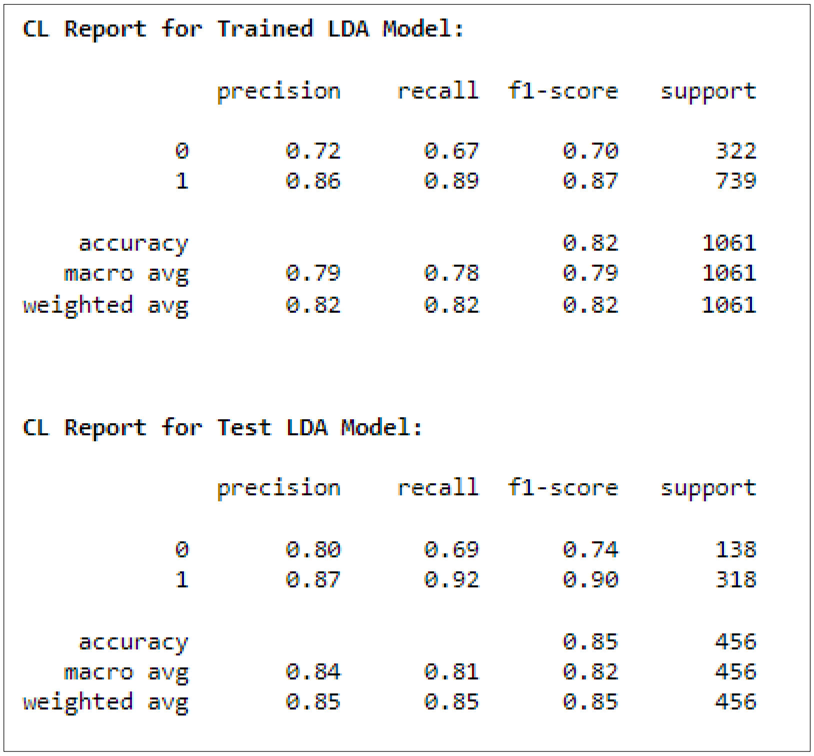
```
CL Report for Trained RF Model:

              precision    recall  f1-score   support

           0       0.79      0.62      0.69       322
           1       0.85      0.93      0.89       739

    accuracy                           0.84      1061
   macro avg       0.82      0.77      0.79      1061
weighted avg       0.83      0.84      0.83      1061



CL Report for Test RF Model:

              precision    recall  f1-score   support

           0       0.86      0.64      0.73       138
           1       0.86      0.96      0.90       318

    accuracy                           0.86       456
   macro avg       0.86      0.80      0.82       456
weighted avg       0.86      0.86      0.85       456
```

Table 15: Classification Report for Logit Model



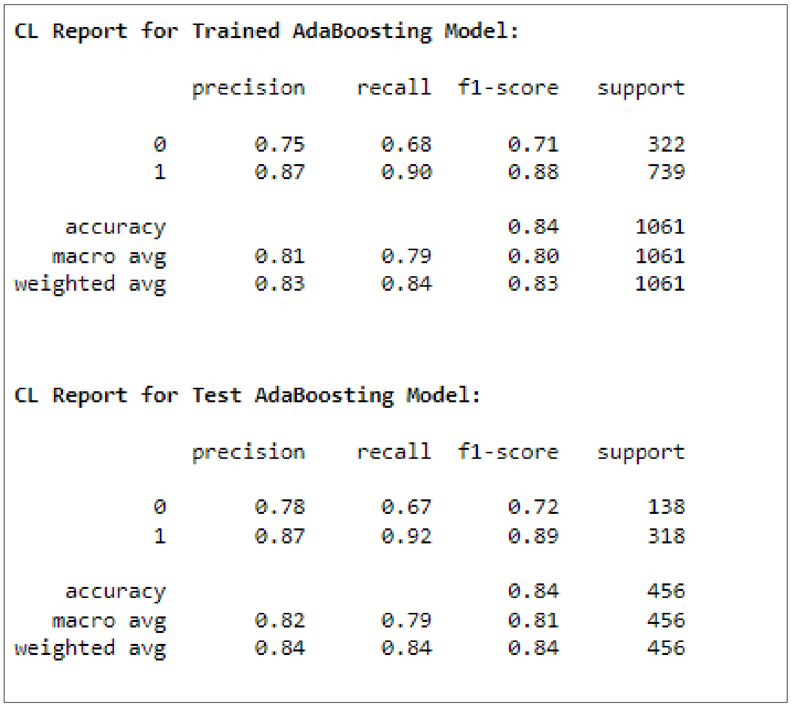
```
CL Report for Trained Logit Model:

              precision    recall  f1-score   support

           0       0.74      0.66      0.70       322
           1       0.86      0.90      0.88       739

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



CL Report for Test Logit Model:

              precision    recall  f1-score   support

           0       0.81      0.68      0.74       138
           1       0.87      0.93      0.90       318

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

Table 16: Classification Report for LDA Model



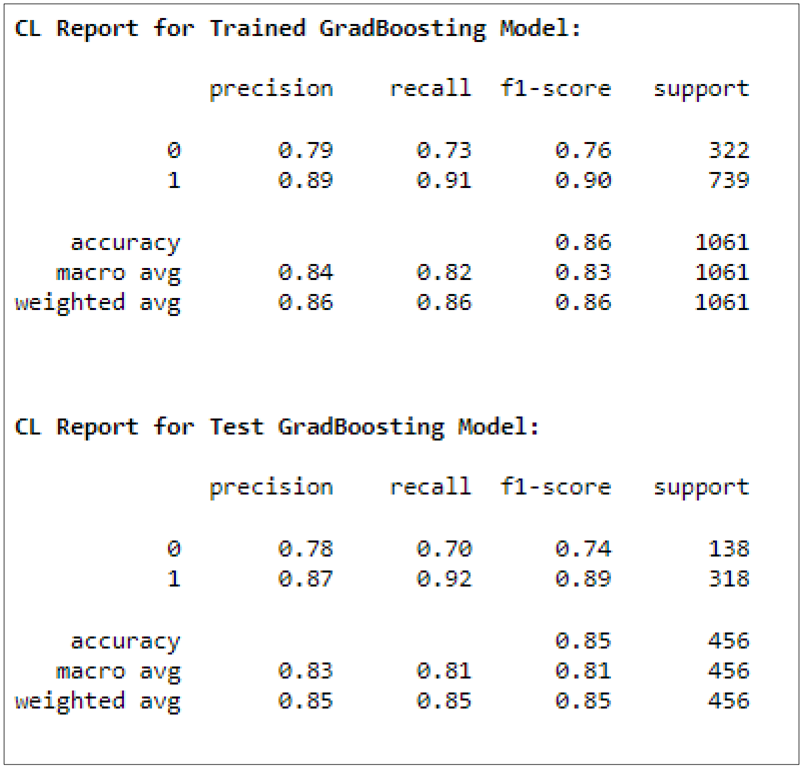
```
CL Report for Trained LDA Model:

              precision    recall  f1-score   support

           0       0.72      0.67      0.70       322
           1       0.86      0.89      0.87       739

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


CL Report for Test LDA Model:

              precision    recall  f1-score   support

           0       0.80      0.69      0.74       138
           1       0.87      0.92      0.90       318

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

Table 17: Classification Report for Ada Boosting Model



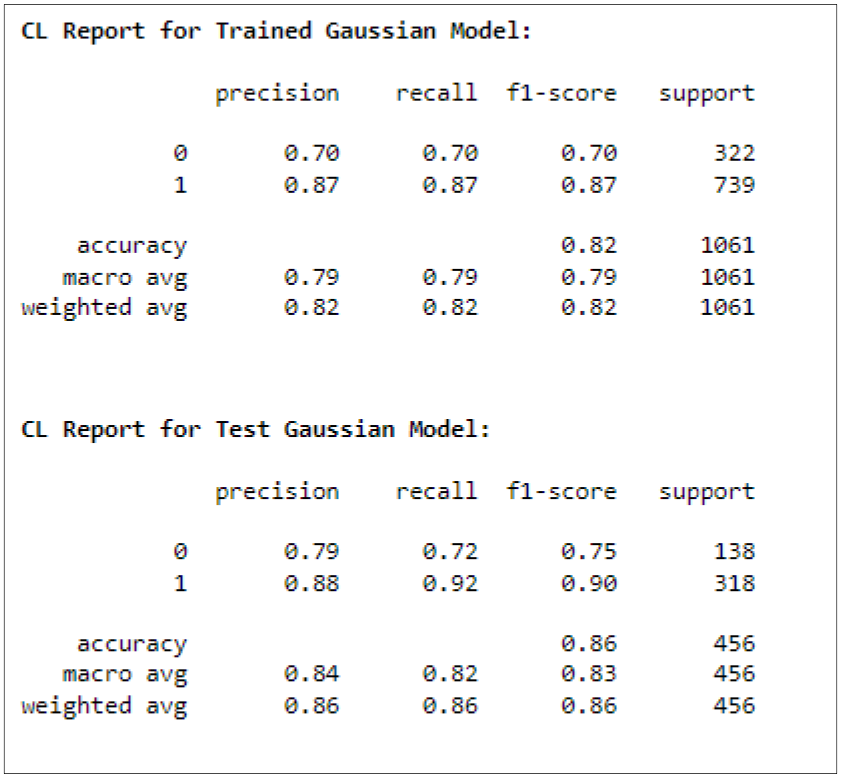
```
CL Report for Trained AdaBoosting Model:

              precision    recall  f1-score   support

           0       0.75      0.68      0.71       322
           1       0.87      0.90      0.88       739

    accuracy                           0.84      1061
   macro avg       0.81      0.79      0.80      1061
weighted avg       0.83      0.84      0.83      1061


CL Report for Test AdaBoosting Model:

              precision    recall  f1-score   support

           0       0.78      0.67      0.72       138
           1       0.87      0.92      0.89       318

    accuracy                           0.84       456
   macro avg       0.82      0.79      0.81       456
weighted avg       0.84      0.84      0.84       456
```

Table 18: Classification Report for Gradient Boosting Model



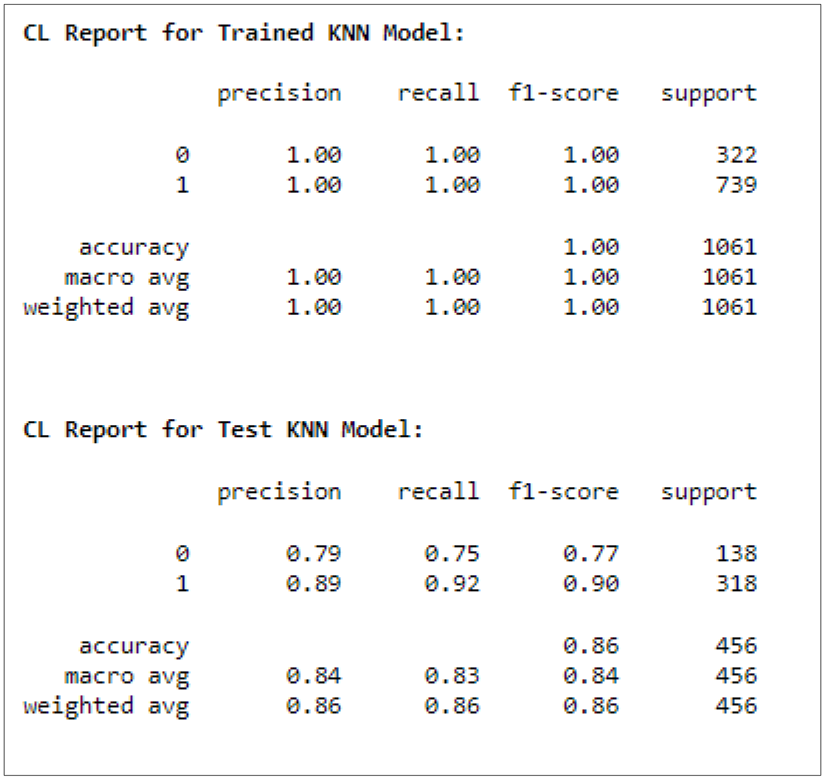
```
CL Report for Trained GradBoosting Model:

              precision    recall  f1-score   support

           0       0.79      0.73      0.76       322
           1       0.89      0.91      0.90       739

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



CL Report for Test GradBoosting Model:

              precision    recall  f1-score   support

           0       0.78      0.70      0.74       138
           1       0.87      0.92      0.89       318

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

Table 19: Classification Report for Gaussian NB Model



```
CL Report for Trained Gaussian Model:

              precision    recall  f1-score   support

           0       0.70      0.70      0.70       322
           1       0.87      0.87      0.87       739

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



CL Report for Test Gaussian Model:

              precision    recall  f1-score   support

           0       0.79      0.72      0.75       138
           1       0.88      0.92      0.90       318

    accuracy                           0.86       456
   macro avg       0.84      0.82      0.83       456
weighted avg       0.86      0.86      0.86       456
```

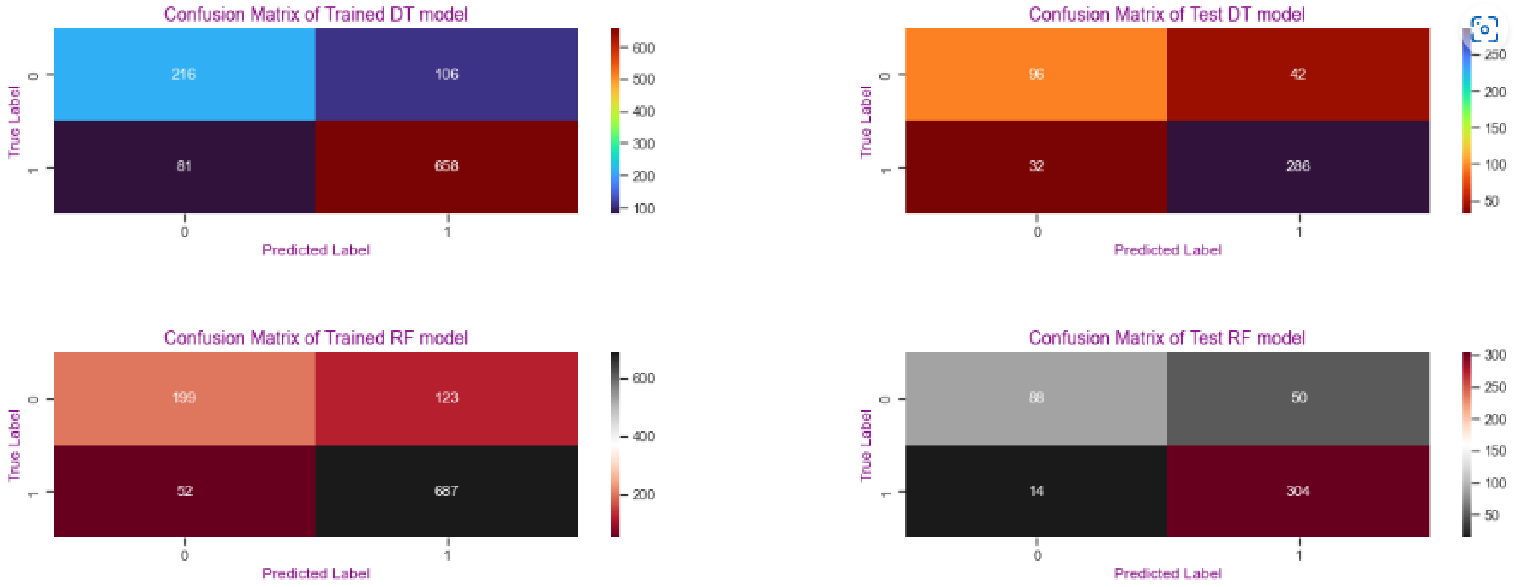
Table 20: Classification Report for KNN Model

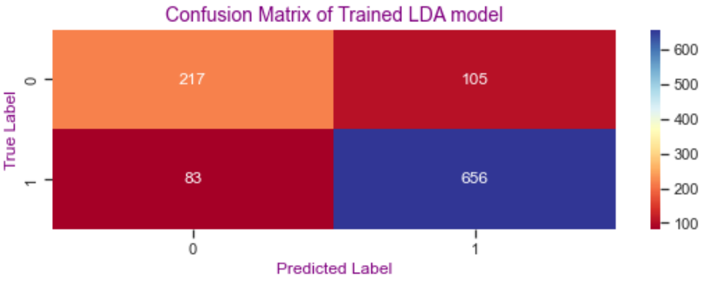


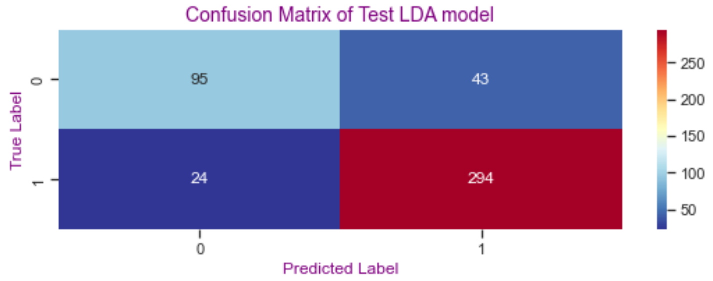
```
CL Report for Trained KNN Model:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       322
           1       1.00      1.00      1.00       739

    accuracy                           1.00      1061
   macro avg       1.00      1.00      1.00      1061
weighted avg       1.00      1.00      1.00      1061



CL Report for Test KNN Model:

              precision    recall  f1-score   support

           0       0.79      0.75      0.77       138
           1       0.89      0.92      0.90       318

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

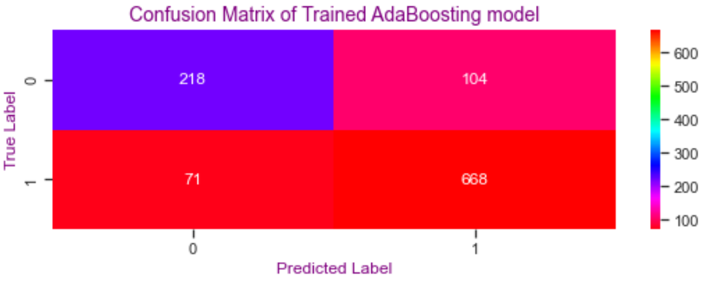
Confusion Matrix

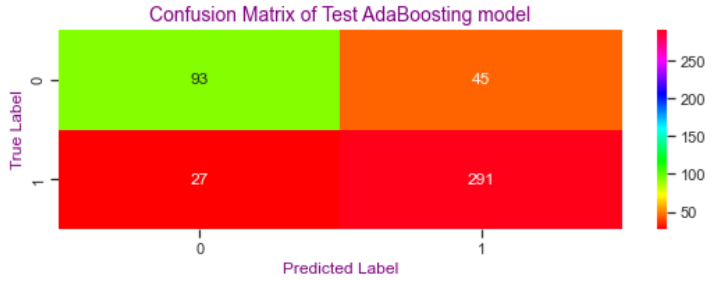
Figure 12: Confusion Matrix of Models

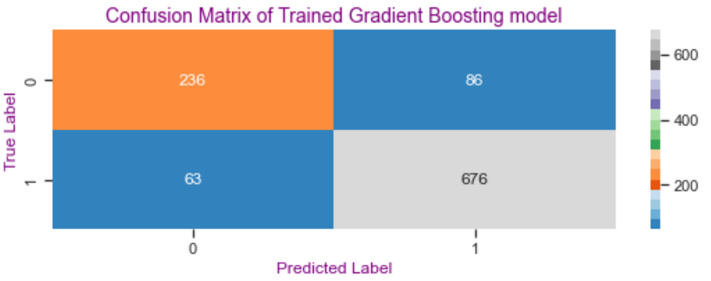


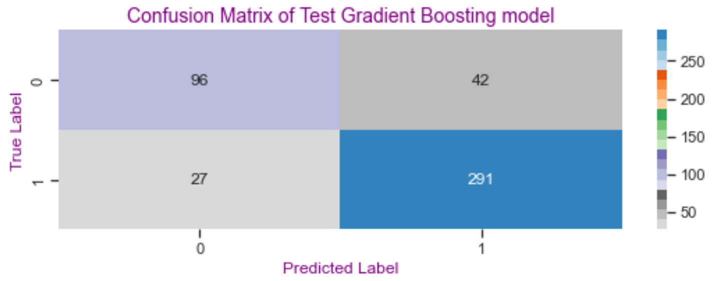

Confusion Matrix of Trained LDA model

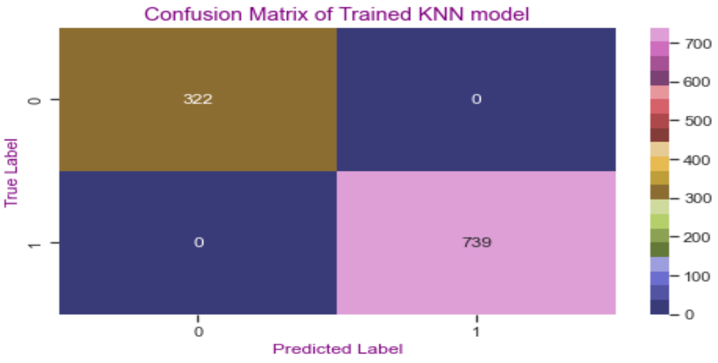

Confusion Matrix of Test LDA model

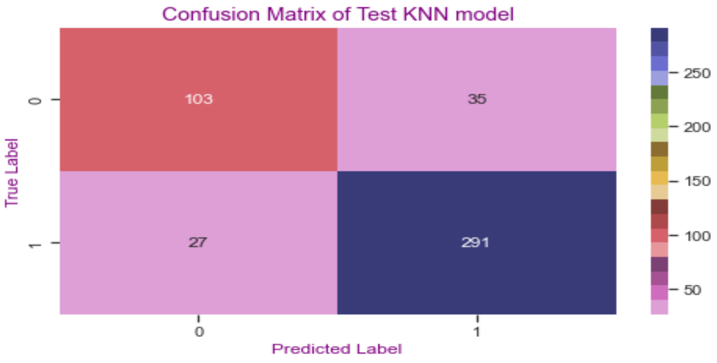

Confusion Matrix of Trained AdaBoosting model

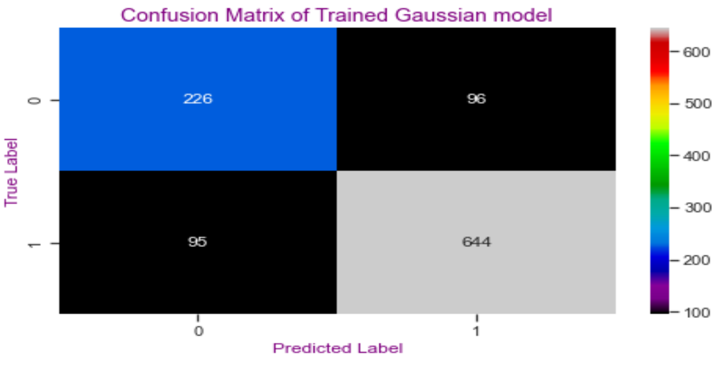

Confusion Matrix of Test AdaBoosting model

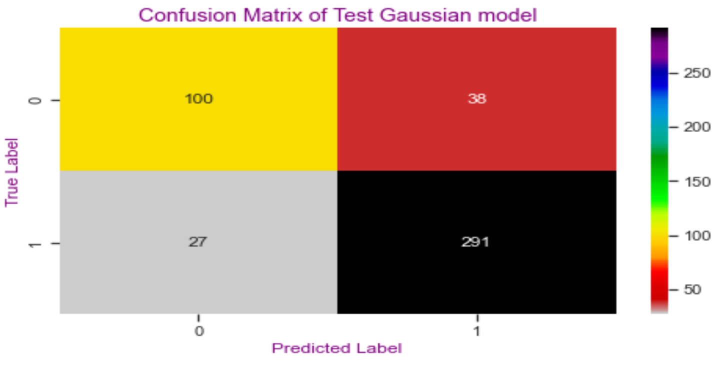

Confusion Matrix of Trained Gradient Boosting model


Confusion Matrix of Test Gradient Boosting model


Confusion Matrix of Trained KNN model

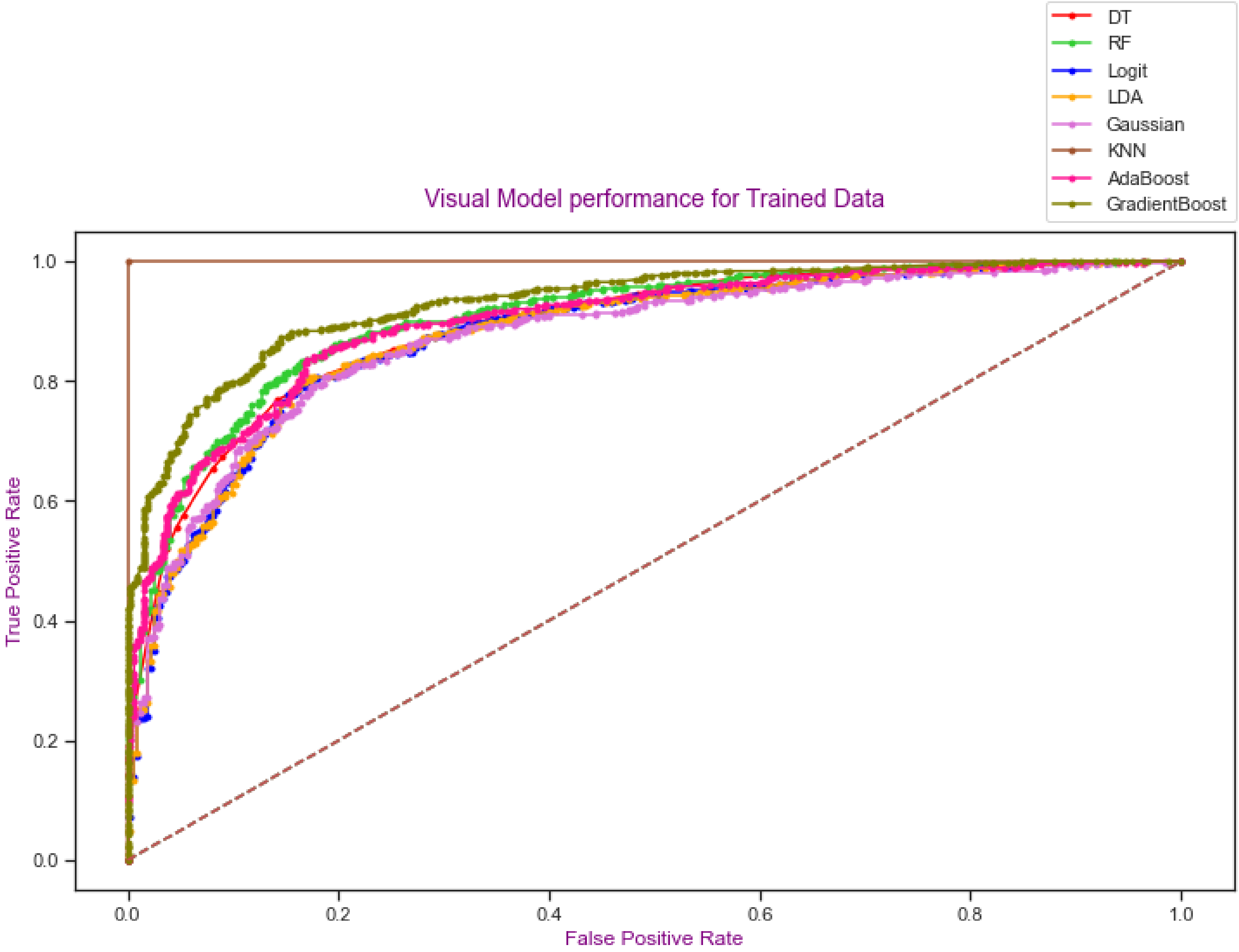

Confusion Matrix of Test KNN model


Confusion Matrix of Trained Gaussian model


Confusion Matrix of Test Gaussian model

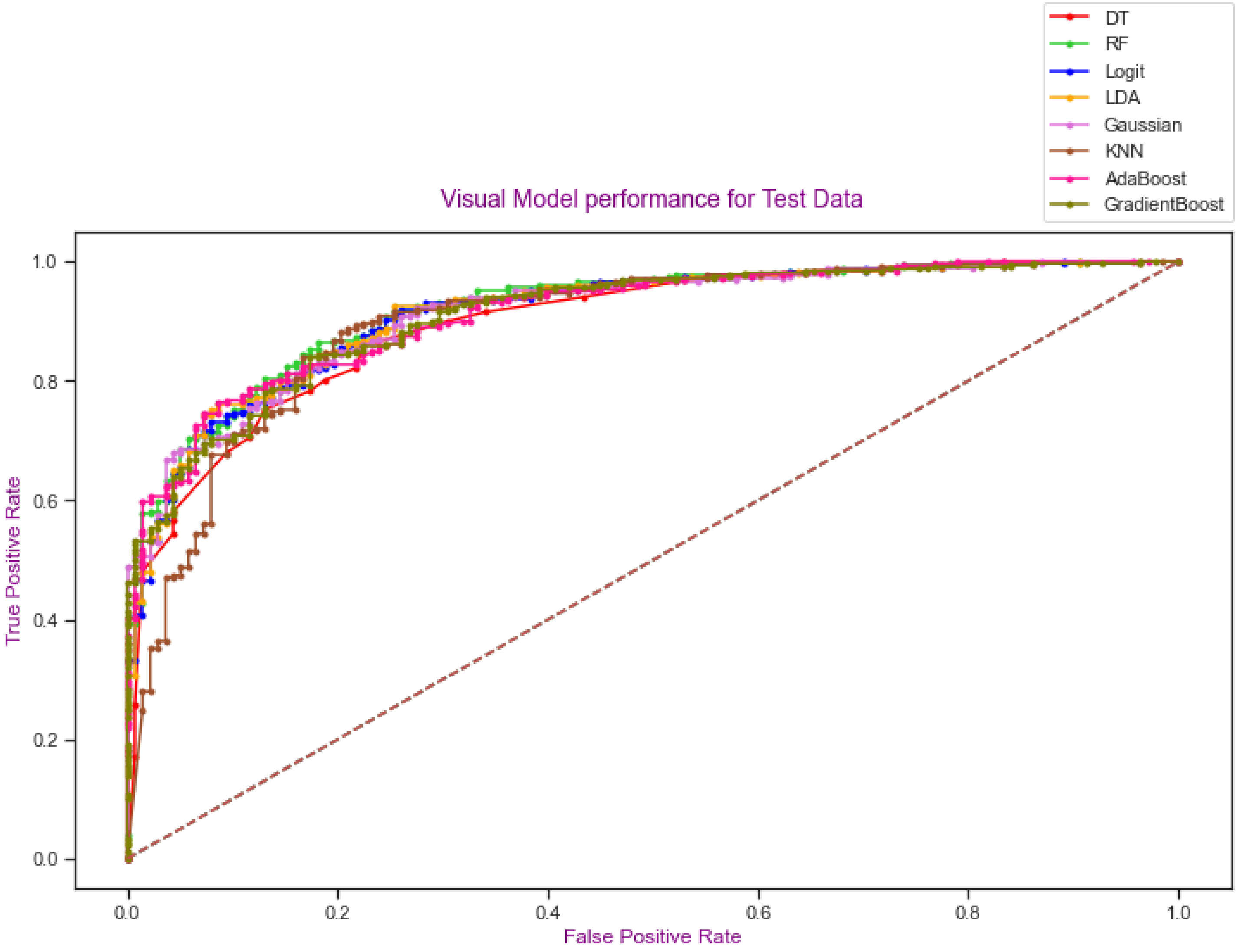
ROC-AUC-Curve

Figure 12: ROC plot for Trained Models



```
Area under the curve for Trained DT Model is       88.92 %
Area under the curve for Trained RF Model is       90.5 %
Area under the curve for Trained Logistic Model is 87.71 %
Area under the curve for Trained LDA Model is      87.69 %
Area under the curve for Trained Gaussian Model is 87.32 %
Area under the curve for Trained KNN Model is      100.0 %
Area under the curve for Trained AdaBoosting Model is 89.83 %
Area under the curve for Trained GradientBoosting Model is 93.05 %
```

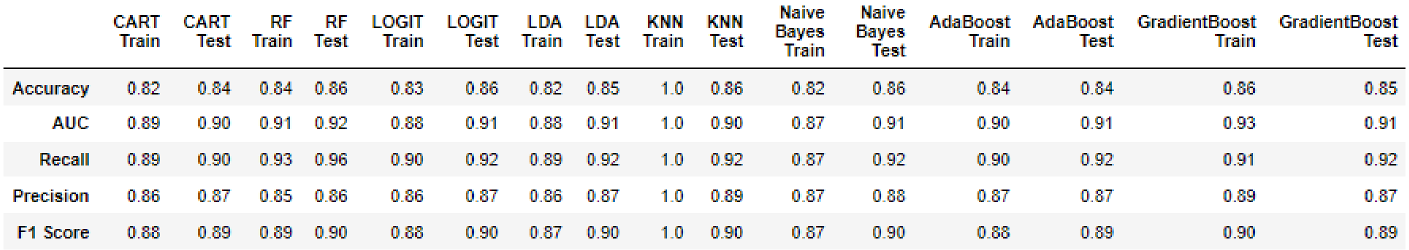
Figure 13: ROC plot for Test Models



```
Area under the curve for Test DT Model is          89.71 %
Area under the curve for Test RF Model is          92.04 %
Area under the curve for Test Logistic Model is    91.29 %
Area under the curve for Test LDA Model is         91.44 %
Area under the curve for Test Gaussian Model is    91.25 %
Area under the curve for Test KNN Model is         89.7 %
Area under the curve for Test AdaBoosting Model is 91.27 %
Area under the curve for Test GradientBoosting Model is 91.02 %
```

Model Comparison

Table 21: Model Comparison



	CART Train	CART Test	RF Train	RF Test	LOGIT Train	LOGIT Test	LDA Train	LDA Test	KNN Train	KNN Test	Naive Bayes Train	Naive Bayes Test	AdaBoost Train	AdaBoost Test	GradientBoost Train	GradientBoost Test
Accuracy	0.82	0.84	0.84	0.86	0.83	0.86	0.82	0.85	1.0	0.86	0.82	0.86	0.84	0.84	0.86	0.85
AUC	0.89	0.90	0.91	0.92	0.88	0.91	0.88	0.91	1.0	0.90	0.87	0.91	0.90	0.91	0.93	0.91
Recall	0.89	0.90	0.93	0.96	0.90	0.92	0.89	0.92	1.0	0.92	0.87	0.92	0.90	0.92	0.91	0.92
Precision	0.86	0.87	0.85	0.86	0.86	0.87	0.86	0.87	1.0	0.89	0.87	0.88	0.87	0.87	0.89	0.87
F1 Score	0.88	0.89	0.89	0.90	0.88	0.90	0.87	0.90	1.0	0.90	0.87	0.90	0.88	0.89	0.90	0.89

Inference

- Overall, all models seems to have performed well in terms of accuracy as well as other metrics except KNN which seems to be overfitted.
- Gradient Boosting seems to have worked well in elevating the accuracy to 86%.
- Recall value seems to be the highest for RF classifier with 96% and only 14 considered as False positives.
- AUC score for RF stands at 92% whereas Gradient Boost seems to 93% and more liberal as seen in roc_auc_curve.
- Precision seems to be at 87% for Testing models, highest for RF classifier with 304 correctly classified.
- Overall, gender doesn't seems to be contributing much while building any models and making predictions.

Overall Best/Optimized model:

- With high recall, precision and accuracy on the test model, statistically, it is safe to conclude that RF classifier can be used as a model to predict the voting party.
- Hague, Europe, Political knowledge seems to have highest contribution to predict the voting party.

1.8 Based on these predictions, what are the insights?

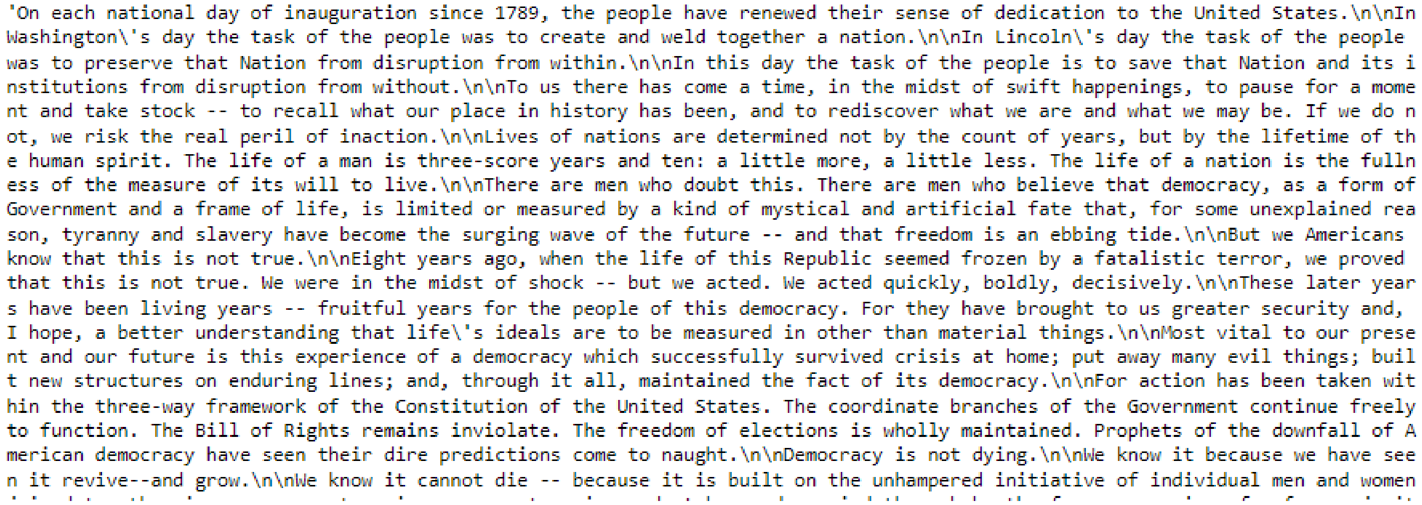
- Data seems to be quite balanced with 30.3% of voters belonging to Conservating party and rest 69.7% to Labour however more data can be collected to generate stronger predictive powers.
- Hague, Europe and Blair seems to be good predictor for voter parties however Gender doesn't happen to be an influencing factor.
- Voters with less Eurosceptic sentiments, seems to have increased as supporters of Labour party.
- Voters between the ages of 30 to 70 has given more votes comparatively and those in the 20's, 80's and 90's has voted significantly less.
- Using RF Classifier model, there are 86% chances of Labour Party being elected as party's choice with 96% probability of voters who are correctly classified of voting Labour party.
- Exit polls can be further created using the ML models along with seats covered by a party.

Problem 2: Text Analysis

Executive Summary

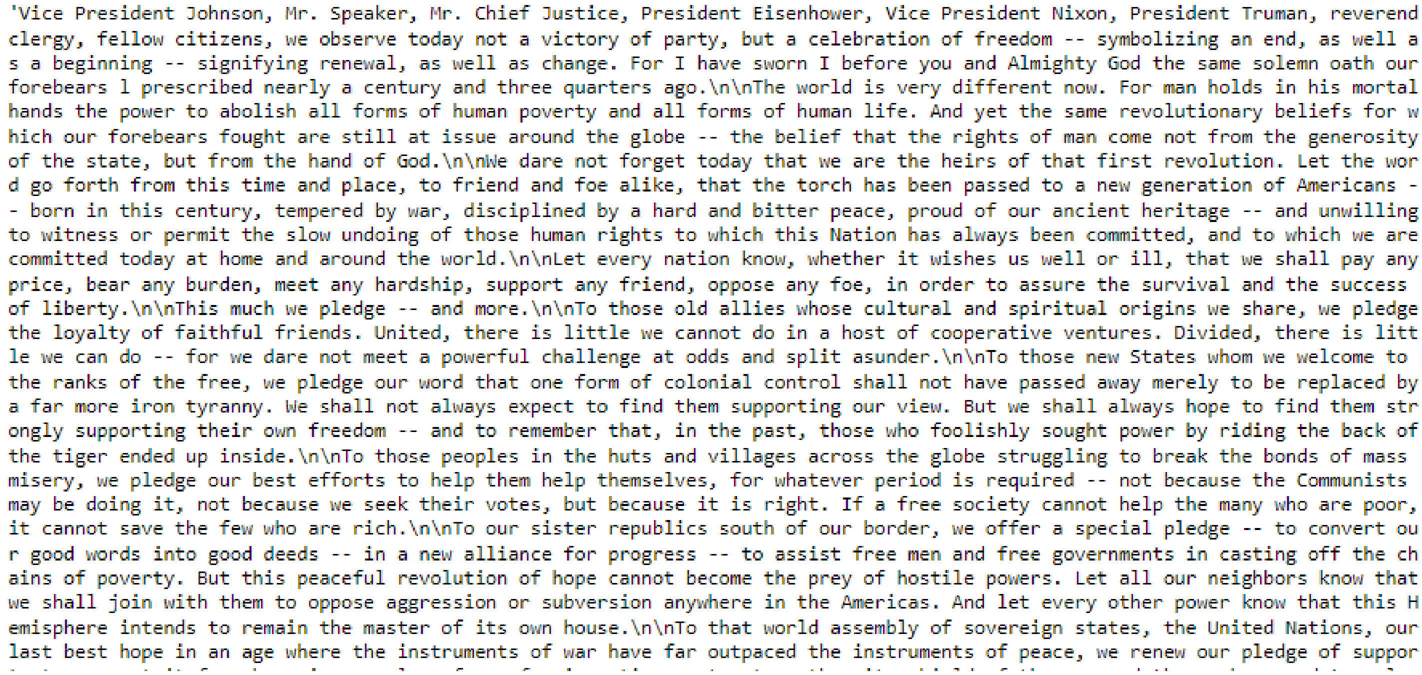
Extracting speeches given by three different presidents of the United States of America and analyzing texts frequencies.

Sample of Inaugural Corpus of Roosevelt Speech



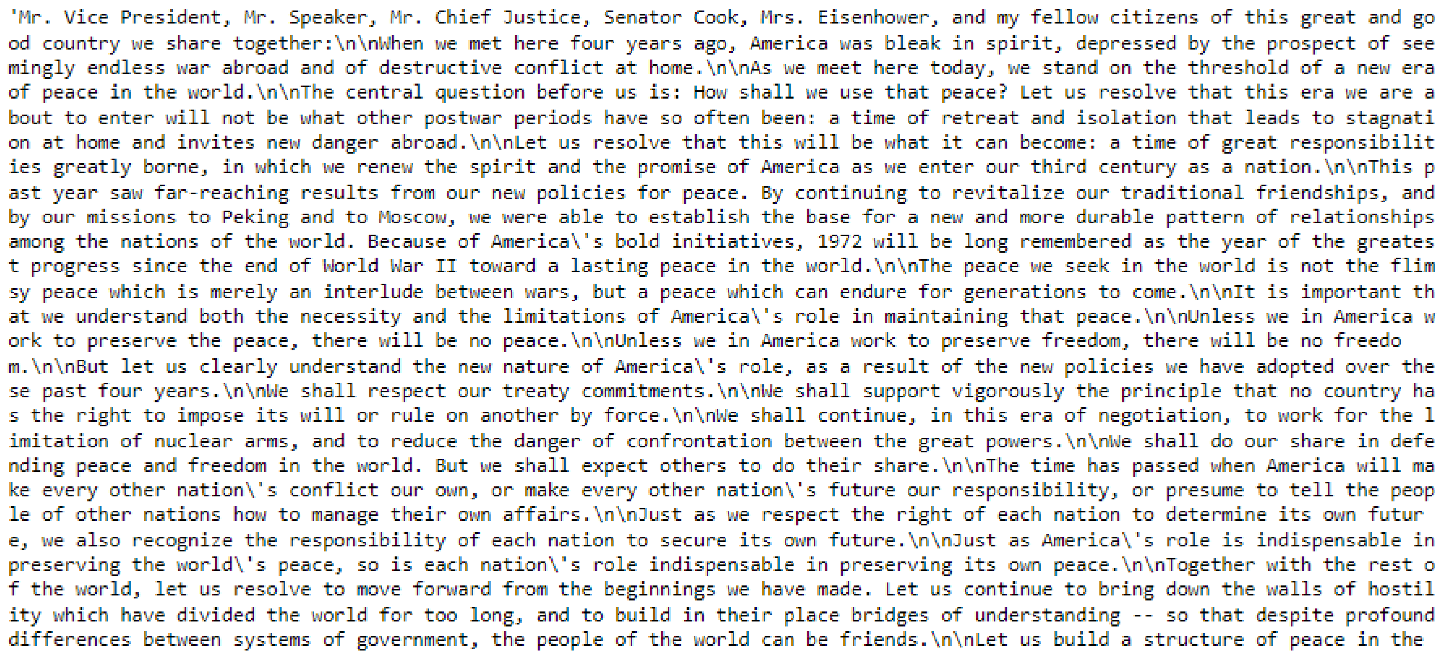
```
'On each national day of inauguration since 1789, the people have renewed their sense of dedication to the United States.\n\nIn
Washington\'s day the task of the people was to create and weld together a nation.\n\nIn Lincoln\'s day the task of the people
was to preserve that Nation from disruption from within.\n\nIn this day the task of the people is to save that Nation and its i
nstitutions from disruption from without.\n\nTo us there has come a time, in the midst of swift happenings, to pause for a mome
nt and take stock -- to recall what our place in history has been, and to rediscover what we are and what we may be. If we do n
ot, we risk the real peril of inaction.\n\nLives of nations are determined not by the count of years, but by the lifetime of th
e human spirit. The life of a man is three-score years and ten: a little more, a little less. The life of a nation is the fulln
ess of the measure of its will to live.\n\nThere are men who doubt this. There are men who believe that democracy, as a form of
Government and a frame of life, is limited or measured by a kind of mystical and artificial fate that, for some unexplained rea
son, tyranny and slavery have become the surging wave of the future -- and that freedom is an ebbing tide.\n\nBut we Americans
know that this is not true.\n\nEight years ago, when the life of this Republic seemed frozen by a fatalistic terror, we proved
that this is not true. We were in the midst of shock -- but we acted. We acted quickly, boldly, decisively.\n\nThese later year
s have been living years -- fruitful years for the people of this democracy. For they have brought to us greater security and,
I hope, a better understanding that life\'s ideals are to be measured in other than material things.\n\nMost vital to our prese
nt and our future is this experience of a democracy which successfully survived crisis at home; put away many evil things; buil
t new structures on enduring lines; and, through it all, maintained the fact of its democracy.\n\nFor action has been taken wit
hin the three-way framework of the Constitution of the United States. The coordinate branches of the Government continue freely
to function. The Bill of Rights remains inviolate. The freedom of elections is wholly maintained. Prophets of the downfall of A
merican democracy have seen their dire predictions come to naught.\n\nDemocracy is not dying.\n\nWe know it because we have see
n it revive--and grow.\n\nWe know it cannot die -- because it is built on the unhampered initiative of individual men and women
```

Sample of Inaugural Corpus of Kennedy Speech



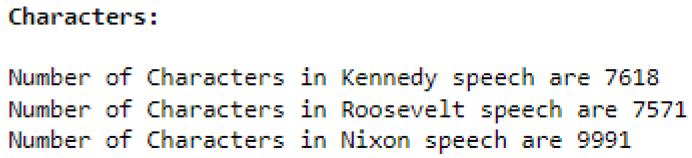
```
'Vice President Johnson, Mr. Speaker, Mr. Chief Justice, President Eisenhower, Vice President Nixon, President Truman, reverend
clergy, fellow citizens, we observe today not a victory of party, but a celebration of freedom -- symbolizing an end, as well a
s a beginning -- signifying renewal, as well as change. For I have sworn I before you and Almighty God the same solemn oath our
forebears l prescribed nearly a century and three quarters ago.\n\nThe world is very different now. For man holds in his mortal
hands the power to abolish all forms of human poverty and all forms of human life. And yet the same revolutionary beliefs for w
hich our forebears fought are still at issue around the globe -- the belief that the rights of man come not from the generosity
of the state, but from the hand of God.\n\nWe dare not forget today that we are the heirs of that first revolution. Let the wor
d go forth from this time and place, to friend and foe alike, that the torch has been passed to a new generation of Americans -
- born in this century, tempered by war, disciplined by a hard and bitter peace, proud of our ancient heritage -- and unwilling
to witness or permit the slow undoing of those human rights to which this Nation has always been committed, and to which we are
committed today at home and around the world.\n\nLet every nation know, whether it wishes us well or ill, that we shall pay any
price, bear any burden, meet any hardship, support any friend, oppose any foe, in order to assure the survival and the success
of liberty.\n\nThis much we pledge -- and more.\n\nTo those old allies whose cultural and spiritual origins we share, we pledge
the loyalty of faithful friends. United, there is little we cannot do in a host of cooperative ventures. Divided, there is litt
le we can do -- for we dare not meet a powerful challenge at odds and split asunder.\n\nTo those new States whom we welcome to
the ranks of the free, we pledge our word that one form of colonial control shall not have passed away merely to be replaced by
a far more iron tyranny. We shall not always expect to find them supporting our view. But we shall always hope to find them str
ongly supporting their own freedom -- and to remember that, in the past, those who foolishly sought power by riding the back of
the tiger ended up inside.\n\nTo those peoples in the huts and villages across the globe struggling to break the bonds of mass
misery, we pledge our best efforts to help them help themselves, for whatever period is required -- not because the Communists
may be doing it, not because we seek their votes, but because it is right. If a free society cannot help the many who are poor,
it cannot save the few who are rich.\n\nTo our sister republics south of our border, we offer a special pledge -- to convert ou
r good words into good deeds -- in a new alliance for progress -- to assist free men and free governments in casting off the ch
ains of poverty. But this peaceful revolution of hope cannot become the prey of hostile powers. Let all our neighbors know that
we shall join with them to oppose aggression or subversion anywhere in the Americas. And let every other power know that this H
emisphere intends to remain the master of its own house.\n\nTo that world assembly of sovereign states, the United Nations, our
last best hope in an age where the instruments of war have far outpaced the instruments of peace, we renew our pledge of suppor
```

Sample of Inaugural Corpus of Kennedy Speech



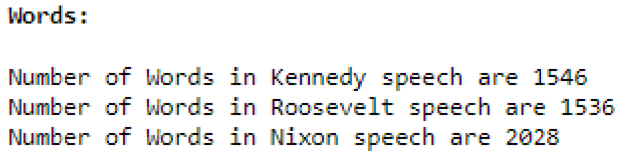
```
'Mr. Vice President, Mr. Speaker, Mr. Chief Justice, Senator Cook, Mrs. Eisenhower, and my fellow citizens of this great and go
od country we share together:\n\nWhen we met here four years ago, America was bleak in spirit, depressed by the prospect of see
mingly endless war abroad and of destructive conflict at home.\n\nAs we meet here today, we stand on the threshold of a new era
of peace in the world.\n\nThe central question before us is: How shall we use that peace? Let us resolve that this era we are a
bout to enter will not be what other postwar periods have so often been: a time of retreat and isolation that leads to stagnati
on at home and invites new danger abroad.\n\nLet us resolve that this will be what it can become: a time of great responsibilit
ies greatly borne, in which we renew the spirit and the promise of America as we enter our third century as a nation.\n\nThis p
ast year saw far-reaching results from our new policies for peace. By continuing to revitalize our traditional friendships, and
by our missions to Peking and to Moscow, we were able to establish the base for a new and more durable pattern of relationships
among the nations of the world. Because of America\'s bold initiatives, 1972 will be long remembered as the year of the greates
t progress since the end of World War II toward a lasting peace in the world.\n\nThe peace we seek in the world is not the flim
sy peace which is merely an interlude between wars, but a peace which can endure for generations to come.\n\nIt is important th
at we understand both the necessity and the limitations of America\'s role in maintaining that peace.\n\nUnless we in America w
ork to preserve the peace, there will be no peace.\n\nUnless we in America work to preserve freedom, there will be no freedo
m.\n\nBut let us clearly understand the new nature of America\'s role, as a result of the new policies we have adopted over the
se past four years.\n\nWe shall respect our treaty commitments.\n\nWe shall support vigorously the principle that no country ha
s the right to impose its will or rule on another by force.\n\nWe shall continue, in this era of negotiation, to work for the l
imitation of nuclear arms, and to reduce the danger of confrontation between the great powers.\n\nWe shall do our share in defe
nding peace and freedom in the world. But we shall expect others to do their share.\n\nThe time has passed when America will ma
ke every other nation\'s conflict our own, or make every other nation\'s future our responsibility, or presume to tell the peop
le of other nations how to manage their own affairs.\n\nJust as we respect the right of each nation to determine its own futur
e, we also recognize the responsibility of each nation to secure its own future.\n\nJust as America\'s role is indispensable in
preserving the world\'s peace, so is each nation\'s role indispensable in preserving its own peace.\n\nTogether with the rest o
f the world, let us resolve to move forward from the beginnings we have made. Let us continue to bring down the walls of hostil
ity which have divided the world for too long, and to build in their place bridges of understanding -- so that despite profound
differences between systems of government, the people of the world can be friends.\n\nLet us build a structure of peace in the
```

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



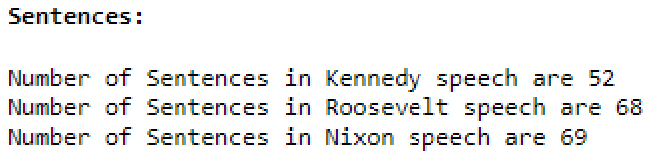
Characters:

```
Number of Characters in Kennedy speech are 7618
Number of Characters in Roosevelt speech are 7571
Number of Characters in Nixon speech are 9991
```



Words:

```
Number of Words in Kennedy speech are 1546
Number of Words in Roosevelt speech are 1536
Number of Words in Nixon speech are 2028
```



Sentences:

```
Number of Sentences in Kennedy speech are 52
Number of Sentences in Roosevelt speech are 68
Number of Sentences in Nixon speech are 69
```

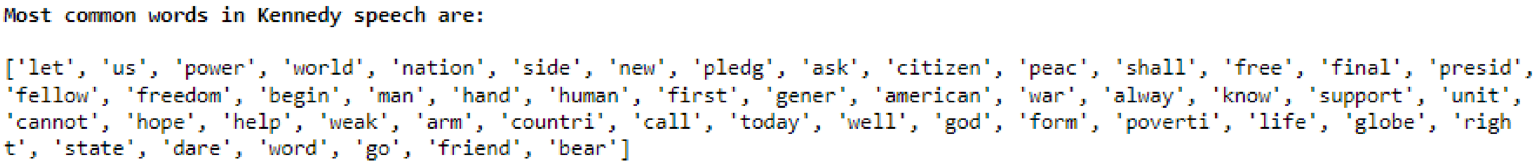
2.2 Remove all the stopwords from all three speeches.

Stopwords are defined as ‘useless’ such as (‘a’, ‘an’, ‘the’, ‘in’) words in NLP which are very common and can get in the way while conducting text mining and taking up valuable processing time.

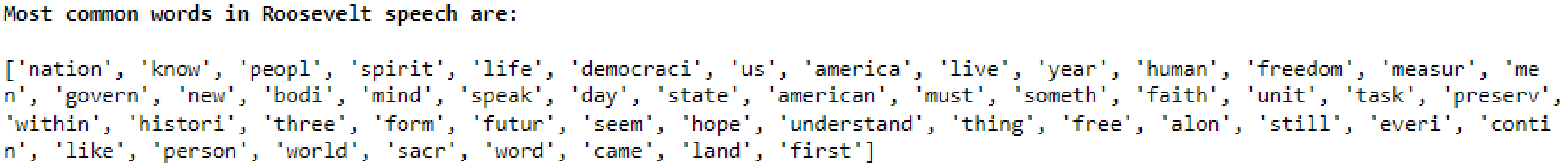
NLTK (Natural Language Tool Kit) is an inbuilt library comprising of stopwords that we can further utilize for cleaning of our data and extend any values that we wouldn’t want to be considered as stopwords.

Stemming is another method of cleaning data by the process of reduced inflection in words to their root forms as mapping a group of words to the same stem.

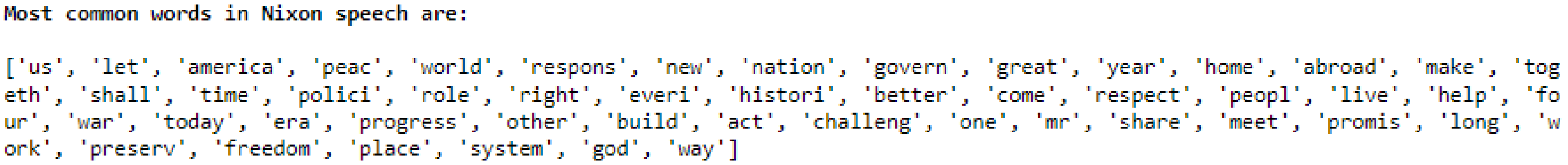
For further analysis, we shall be using ‘PorterStemmer’ – an inbuilt library for Stemming.



```
Most common words in Kennedy speech are:

['let', 'us', 'power', 'world', 'nation', 'side', 'new', 'pledg', 'ask', 'citizen', 'peac', 'shall', 'free', 'final', 'presid',
'fellow', 'freedom', 'begin', 'man', 'hand', 'human', 'first', 'gener', 'american', 'war', 'alway', 'know', 'support', 'unit',
'cannot', 'hope', 'help', 'weak', 'arm', 'countri', 'call', 'today', 'well', 'god', 'form', 'poverti', 'life', 'globe', 'righ
t', 'state', 'dare', 'word', 'go', 'friend', 'bear']
```

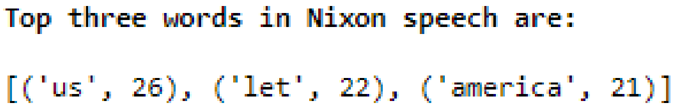


```
Most common words in Roosevelt speech are:

['nation', 'know', 'peopl', 'spirit', 'life', 'democraci', 'us', 'america', 'live', 'year', 'human', 'freedom', 'measur', 'me
n', 'govern', 'new', 'bodi', 'mind', 'speak', 'day', 'state', 'american', 'must', 'someth', 'faith', 'unit', 'task', 'preserv',
'within', 'histori', 'three', 'form', 'futur', 'seem', 'hope', 'understand', 'thing', 'free', 'alon', 'still', 'everi', 'conti
n', 'like', 'person', 'world', 'sacr', 'word', 'came', 'land', 'first']
```

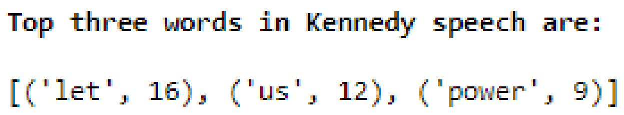


```
Most common words in Nixon speech are:

['us', 'let', 'america', 'peac', 'world', 'respons', 'new', 'nation', 'govern', 'great', 'year', 'home', 'abroad', 'make', 'tog
eth', 'shall', 'time', 'polici', 'role', 'right', 'everi', 'histori', 'better', 'come', 'respect', 'peopl', 'live', 'help', 'fo
ur', 'war', 'today', 'era', 'progress', 'other', 'build', 'act', 'challeng', 'one', 'mr', 'share', 'meet', 'promis', 'long', 'w
ork', 'preserv', 'freedom', 'place', 'system', 'god', 'way']
```

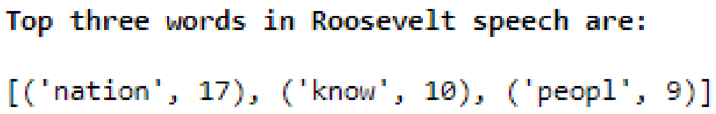
2.3 Which word occurs the most number of times in his inaugural address for each president? Mention the top three words. (after removing the stopwords)



```
Top three words in Nixon speech are:

[('us', 26), ('let', 22), ('america', 21)]
```



```
Top three words in Kennedy speech are:

[('let', 16), ('us', 12), ('power', 9)]
```

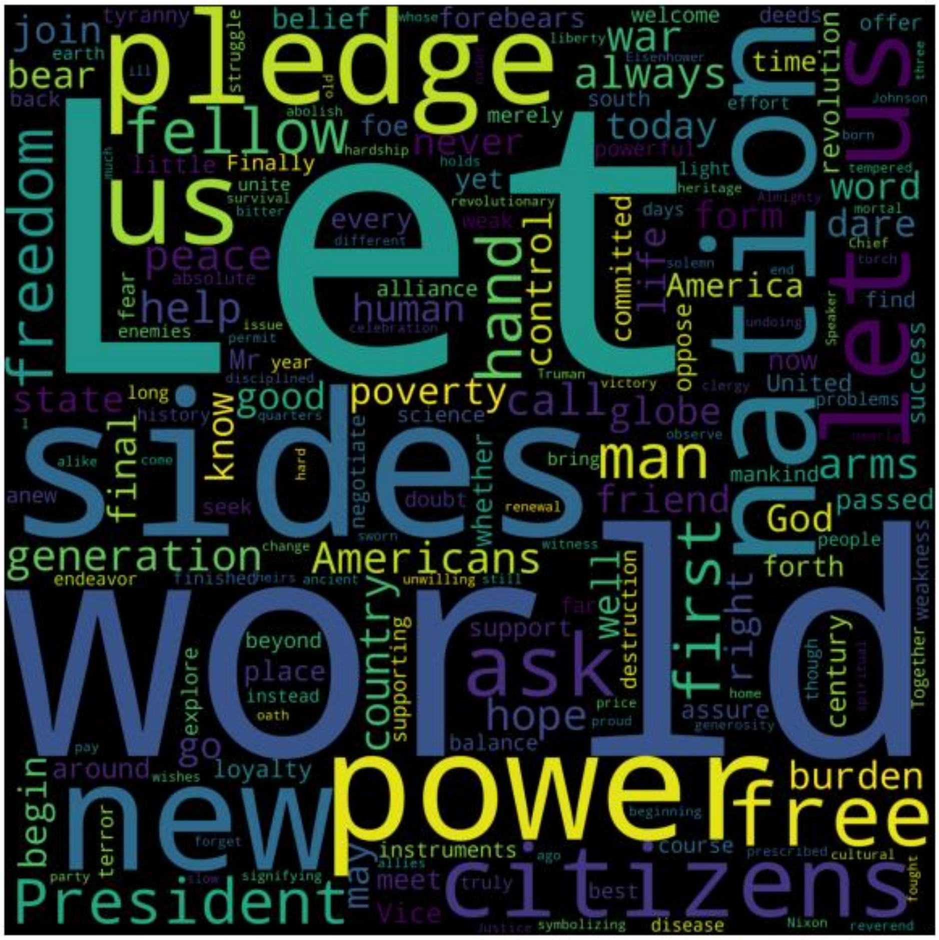


```
Top three words in Roosevelt speech are:

[('nation', 17), ('know', 10), ('peopl', 9)]
```

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

Wordcloud is a cloud filled with plethora of words in different sizes representing the frequencies and importance of each word.

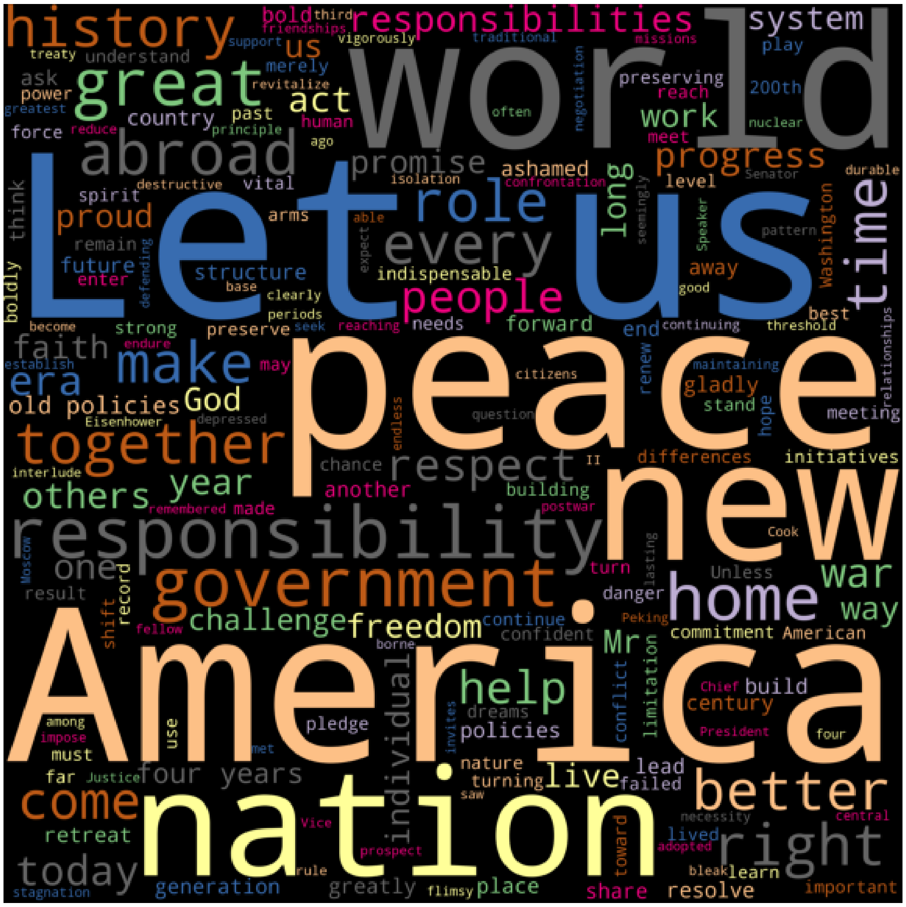
Figure 14: Kennedy Speech



Inference

- Most frequent words seems to be power, pledge, Let, world

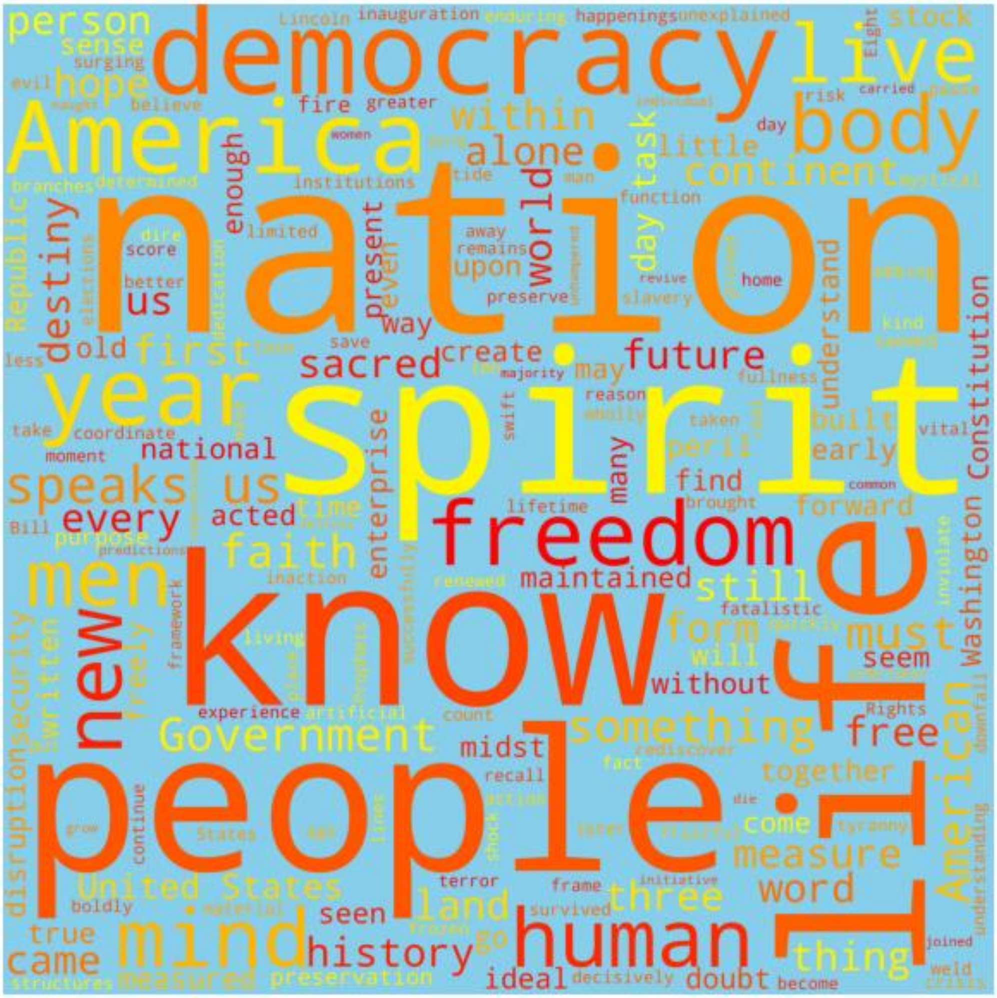
Figure 15: Nixon Speech



Inference

- Most frequent words seems to be America, responsibility, nation, peace, Let, Us

Figure 16: Roosevelt Speech



Inference

- Most frequent words seems to be spirit, people, nation, democracy.

---------------------------THE END ---------------------------