Modelling - Logistic Regression, Linear Discriminant Analysis, Random Forest, K-Nearest Neighbour, Naive Bayes, Bagging, Boosting

Text Analytics

21-FEB-2021

MACHINE LEARNING

PGP DSBA JULY B - GROUP 2
CHETAN.DUDHANE@GMAIL.COM

CHETAN DUDHANE

INTRODUCTION

This report consists of two problem statements -

- PROBLEM 1 - Election Data for Exit Polls **(Classification Modelling)**
- PROBLEM 2 - **Text Analytics** of US Presidents' speeches

Please find the Jupyter Code Notebook here. Analysis code is in Python. Datasets used are in the same directory.

PROBLEM 1 - Election Data for Exit Polls

You are hired by one of the leading news channel 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 Description :

Variable	Description
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.A DATA EXPLORATION

vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
Labour	43	3	3	4	1	2	2	female
Labour	36	4	4	4	4	5	2	male
Labour	35	4	4	5	2	3	2	male
Labour	24	4	2	2	1	4	0	female
Labour	41	2	2	1	1	6	2	male
Labour	47	3	4	4	4	4	2	male
Labour	57	2	2	4	4	11	2	male
Labour	77	3	4	4	1	1	0	male
Labour	39	3	3	4	4	11	0	female
Labour	70	3	2	5	1	11	2	male

Election Data - First 10 rows

```
<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)
```

Election Data - Summary Info

1.B DESCRIPTIVE STATISTICS

	count	mean	std	min	0.25	0.50	0.75	max
age	1525	54.18	15.71	24.00	41.00	53.00	67.00	93.00
economic.cond. national	1525	3.25	0.88	1.00	3.00	3.00	4.00	5.00
economic.cond. household	1525	3.14	0.93	1.00	3.00	3.00	4.00	5.00
Blair	1525	3.33	1.17	1.00	2.00	4.00	4.00	5.00
Hague	1525	2.75	1.23	1.00	2.00	2.00	4.00	5.00
Europe	1525	6.73	3.30	1.00	4.00	6.00	10.00	11.00
political.knowledge	1525	1.54	1.08	0.00	0.00	2.00	2.00	3.00

Election Data - Descriptive Stats of Numeric Variables

	count	unique	top	freq
vote	1525	2.00	Labour	1063.00
gender	1525	2.00	female	812.00

Election Data - Descriptive Stats of Categorical Variables

1.C SYNOPSIS

1. Total No. Of Voter Entries = 1525
2. Total No. Of Variables = 9
3. Target / Response Variable = ‘VOTE’
4. No. Of Missing Values = 0

5. No. Of Duplicate Entries = 8

    - As Duplicate entries don't add any value, WE DROP THEM

6. Final dataset entries for Modelling = 1517

7. **We create multiple Classification Models to predict if voters will vote for 'Labour' or 'Conservative'.**

8. **Data Balance** - Here, data is 70:30 in favour of Labour. In experience, this is a fairly good balance for Classification models. Hence, we will not use any Balancing techniques

9. **Scaling** - Its not necessary to scale for each model, but some models perform better with Scaling.

Hence, we test each model on data scaled 2 ways -

a. Only variable 'AGE' - MinMax Scaled
b. Only variable 'AGE' - Zscore Standard Scaled

10. **Threshold** for Classification -

    - Default threshold = 0.5

    - For each model, we check Performance metrics for many thresholds and pick the one which gives the best scores

11. Classification Models used -

a. Naive Bayes
b. Logistic Regression
c. Linear Discriminant Analysis
d. K-Nearest Neighbour - Optimal K=15
e. Adaptive Boosting
f. Gradient Boosting
g. Bagging - Random Forest

12 **Bagging using Random Forest and Gradient Boost are top 2 performers** considering Test Score and Test AUC

But here **Bagging using Random Forest should be preferred over Gradient Boost** as -

- Random Forest is easier to tune than Gradient Boost

- Here, in this Problem Statement, **interpretability and feature importance** would greatly help in designing and predicting Exit Polls

- Random Forest outputs Feature Importance very well. It gives weightage of each Feature individually

[Q 1.1] Read the dataset. Do the descriptive statistics and do null value condition check. Write an inference on it. (5 Marks)

- Data is read and stored as Pandas Data Frame for analysis
- Last 5 rows of the data are given below

vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
Conservative	67	5	3	2	4	11	3	male
Conservative	73	2	2	4	4	8	2	male
Labour	37	3	3	5	4	2	2	male
Conservative	61	3	3	1	4	11	2	male
Conservative	74	2	3	2	4	11	0	female

Election Data - Last 5 Rows

✦ Check for Duplicates -

- No. Of Duplicate Records found = 8

    - As they add no value, WE DROP DUPLICATES

✦ Check for Null Values -

- No. Of Missing Values = 0

✦ Descriptive Statistics and Inference -

- 5 point Summary and other Descriptive Stats are given in the Section 1.B above
- age --------> Mean = 54.18, Median = 53.0, CV = 29.0

  economic.cond.national --------> Mean = 3.25, Median = 3.0, CV = 27.14

  economic.cond.household --------> Mean = 3.14, Median = 3.0, CV = 29.61

  Blair --------> Mean = 3.33, Median = 4.0, CV = 35.23

  Hague --------> Mean = 2.75, Median = 2.0, CV = 44.8

  Europe --------> Mean = 6.73, Median = 6.0, CV = 49.01

  political.knowledge --------> Mean = 1.54, Median = 2.0, CV = 70.24

  * CV = Coefficient of Variation

- Maximum Variation is in Voters' Political Knowledge

  - suggests no consistency in Voters' Political Knowledge

  - large gaps in their Political Knowledge

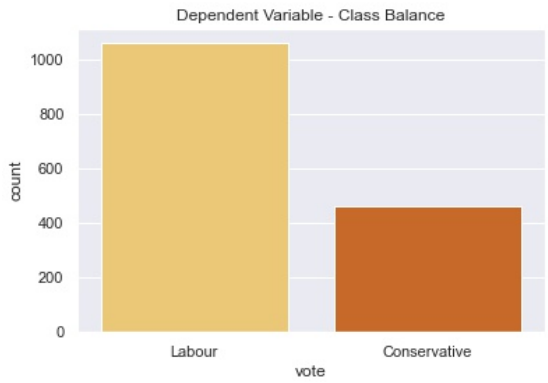
- National and Household Economic Condition ratings have low variation

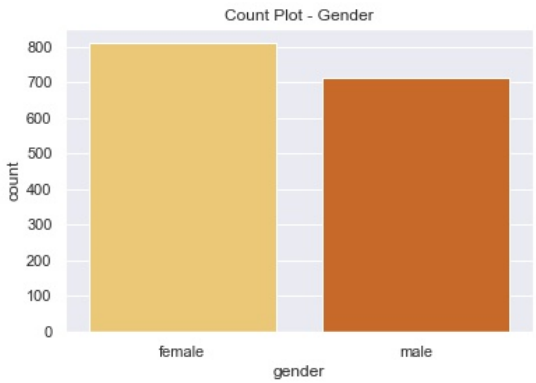
  - suggests Voters have consistent opinion on these 2 parameters

- 'AGE' has almost same Mean and Median

  - suggests maximum concentration of Voters with age 53-54 yrs

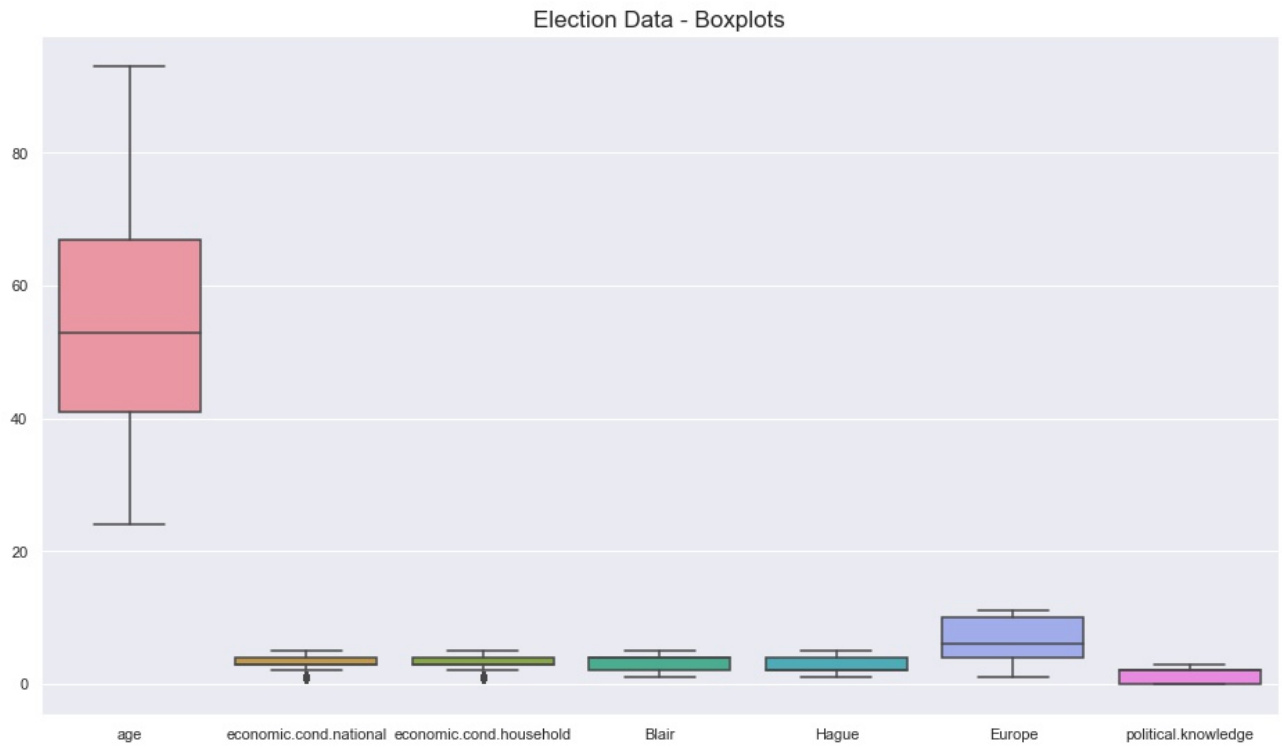
- **Maximum** Voter population in the data is **Female** and those who have voted for **Labour** Party





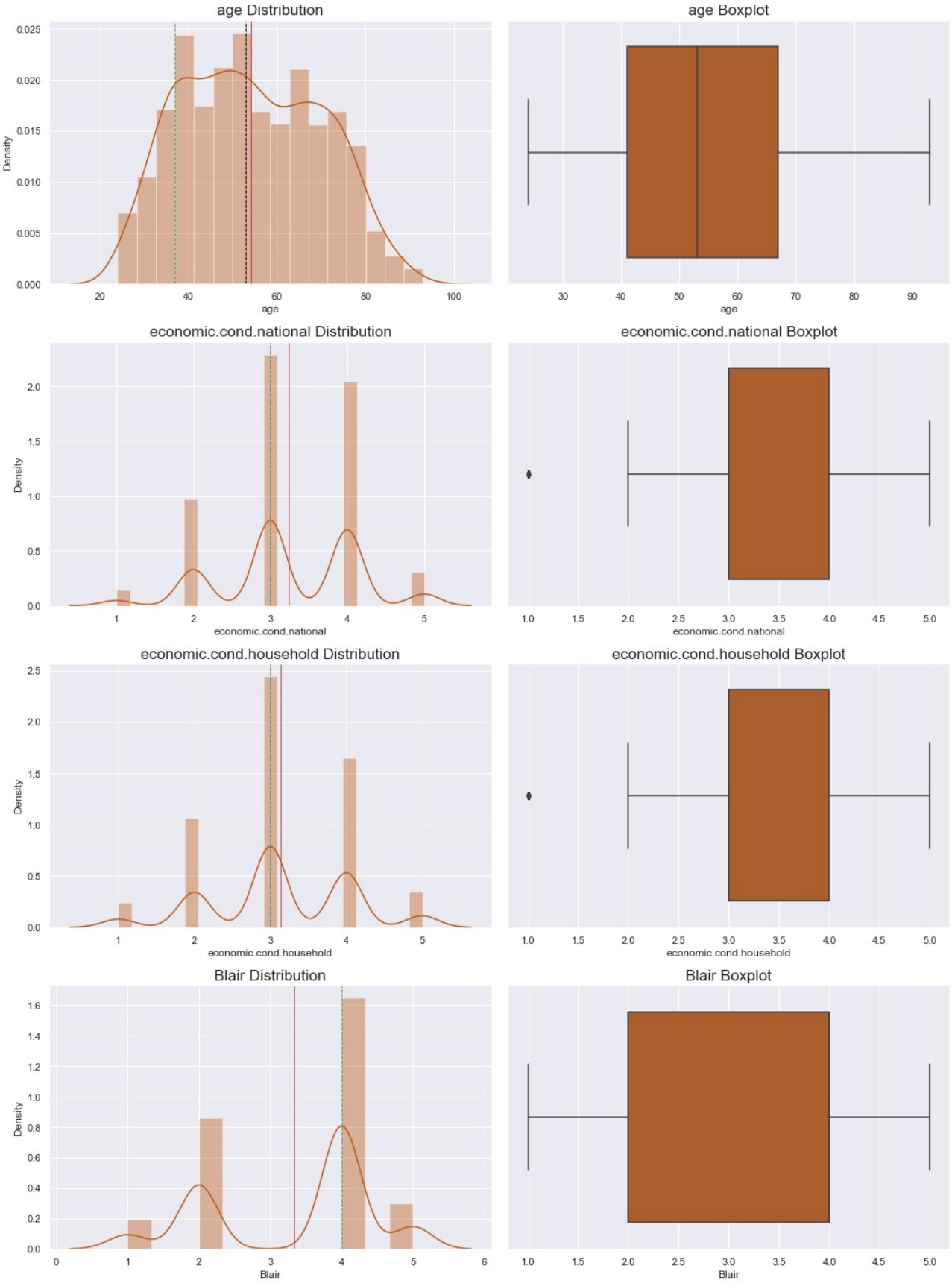
[Q 1.2] Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers. (7 Marks)

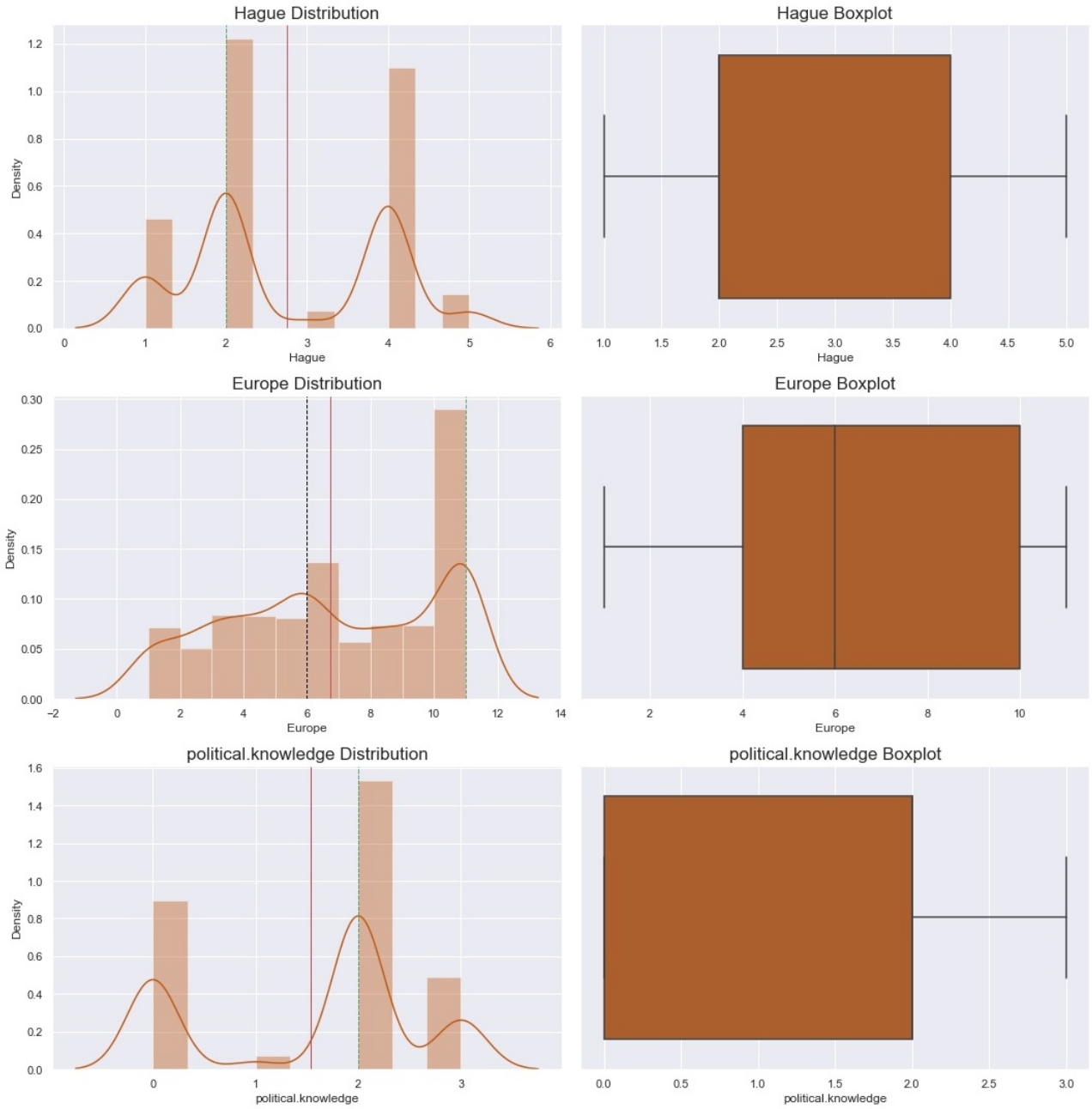
✦ Check for Outliers -



- There are very few Outliers.
- Outliers are present in 2 variables - 'economic.cond.household' and 'economic.condition.national'.
- But **both these 2 variables contain ratings**, which makes them ordinal categorical variables.
- We treat outliers only for continuous variables and not for categorical variables.
- Hence here, **we don't treat outliers .**

✦ Univariate Analysis -

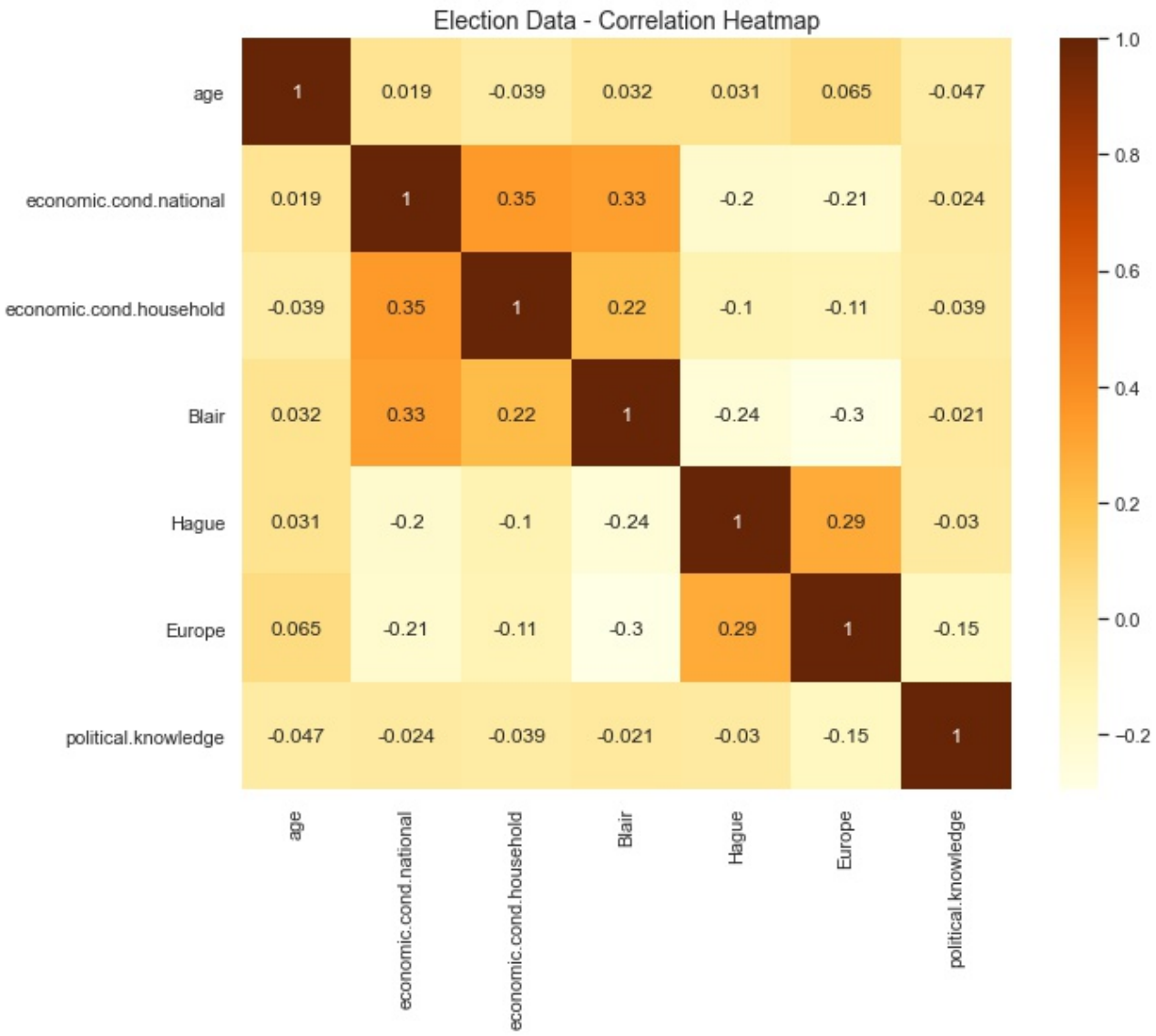




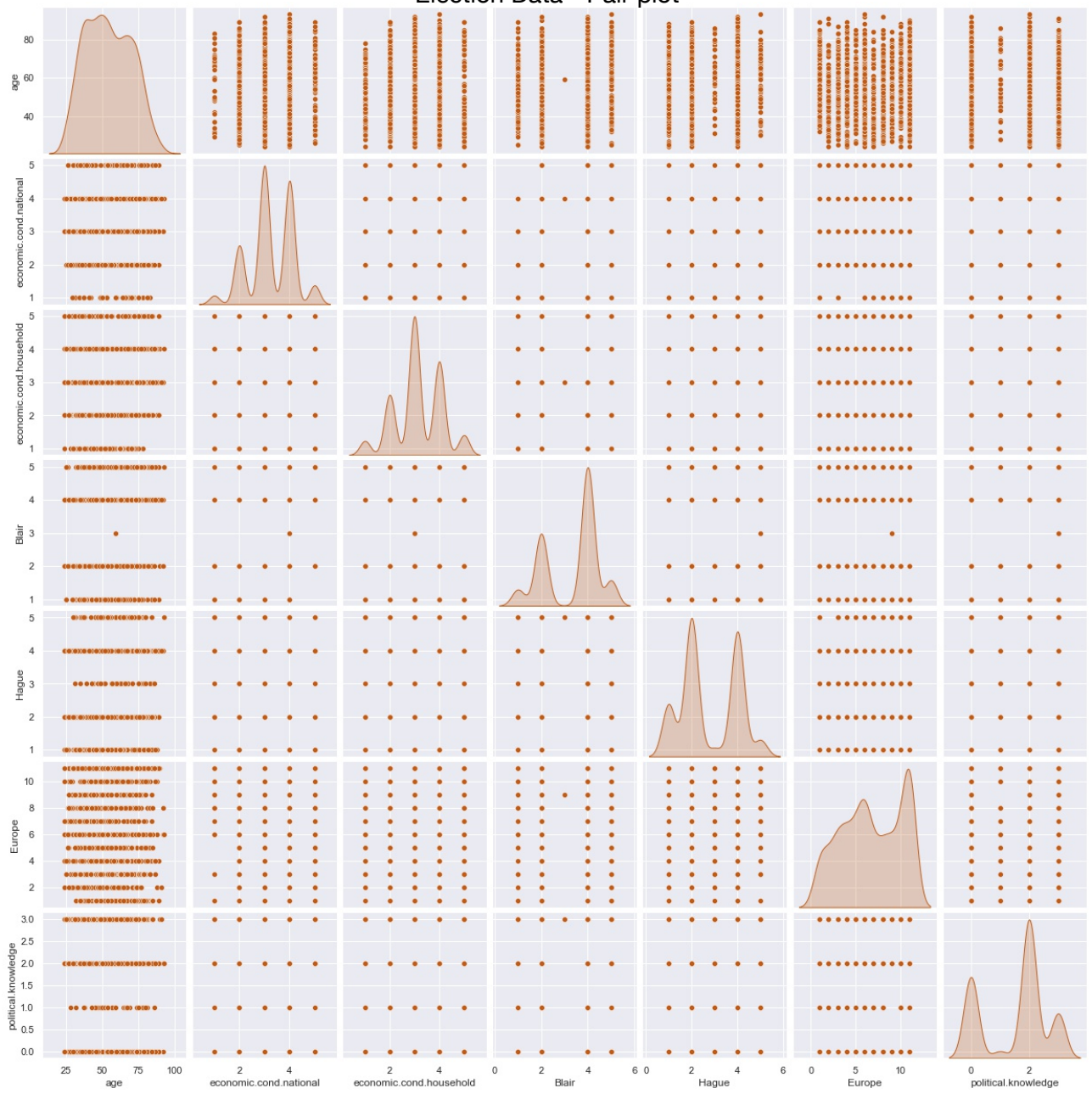
- '**age**' - shows a symmetric distribution
  - has mean and median almost overlapping at 53-54
  - shows **high concentration of voters in age group of 40-60 yrs**
- '**economic.condition.national**' & '**economic.condition.household**'
  - shows high density of voters with 3, 4 & 5 ratings for both these variables
  - **suggests general feeling of well-being amongst voters**

- '**Blair**' & '**Hague**' - Labour leader **Blair has higher proportion of ratings >3**
  - Conservative leader **Hague has higher proportion of ratings <3**
  - **suggests better approval rating for Blair**
- '**Europe**' - a shoot-up in density of voters with ratings 10-11 (around 30%)
  - around 50% of voters have ratings >6
  - **suggests high percentage of voters showing Euro-sceptic sentiments**
- '**political.knowledge**' - its a scale of 0 to 3
  - **75% show good Political Knowledge** with ratings 2 and 3
  - whereas almost **25% show zero Political Knowledge**

✦ Bivariate Analysis -



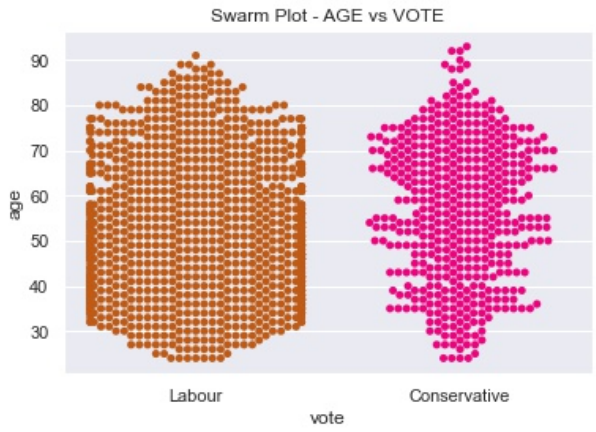
Election Data - Pair plot

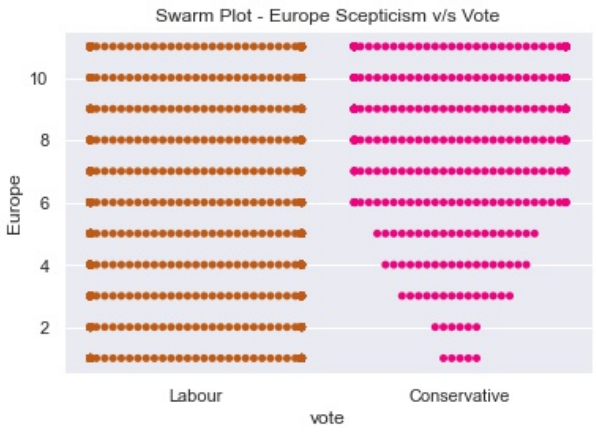


- There is no significant correlation between variables
- Pair-plot shows distribution of Voters who have voted for Labour or Conservative Party
- Distribution doesn't show any clear trend or relation.

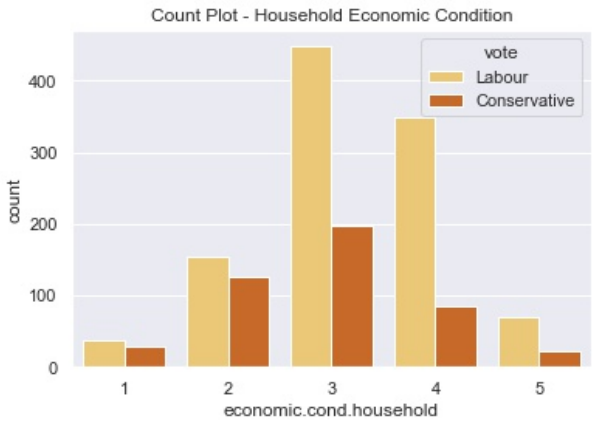
✦ Exploratory Data Analysis -

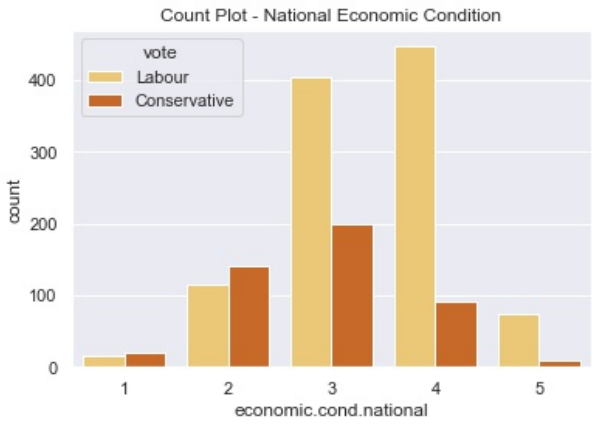
- **Swarm Plots -**

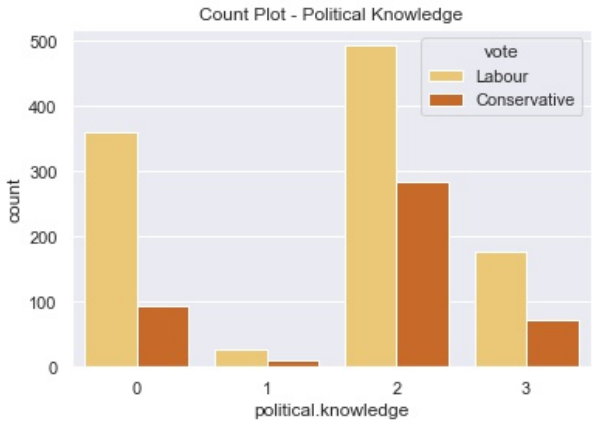


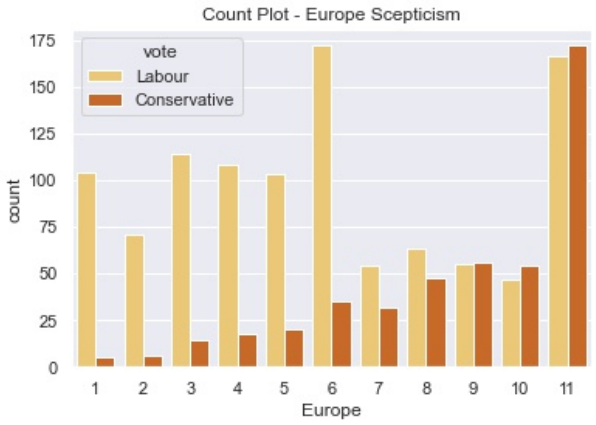


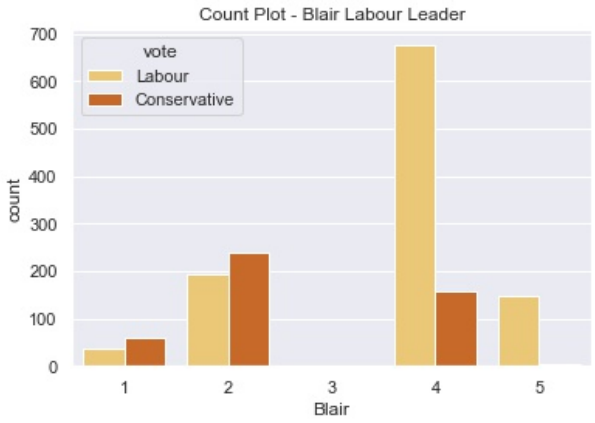
- **Count Plots -**

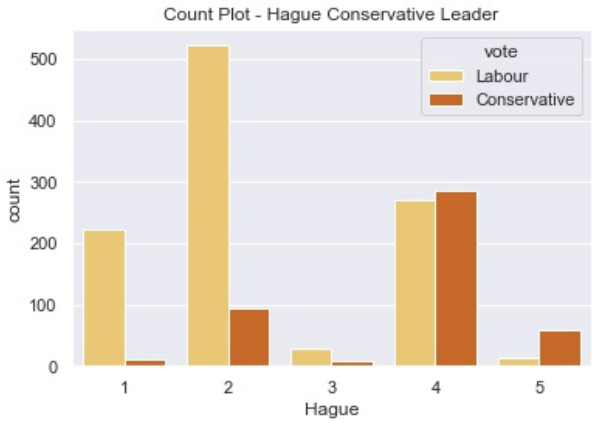












- **Analysis -**
  - Labour party voters uniformly belong to all age groups
  - Conservative party voters are mostly seniors, aged around 50+. Also, 90+ voters are seen to specifically prefer Conservative party
  - Voters rating 3 or more for the National and Household economic condition are seen to prefer labour party more than Conservative
  - Proportionately, voters with zero Political Knowledge are seen to prefer Labour party more than Conservatives.
  - Voters with lower ratings for Euro-sceptic sentiments largely favour Labour party

    Whereas, voters with higher ratings are seen to favour Conservative increasingly more, but here overall both parties have similar vote share.
  - And obviously, voters giving higher approval ratings to Blair vote for Labour party and those giving higher ratings to Hague vote for Conservatives

[Q 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). (5 Marks)

✦ Data Encoding -

- Almost all Classification Models need all attributes in the Numeric format
- Hence, We Encode the data containing String values into Numeric Values
- In our Election Data, Variables to be encoded are - 'VOTE' and 'GENDER'

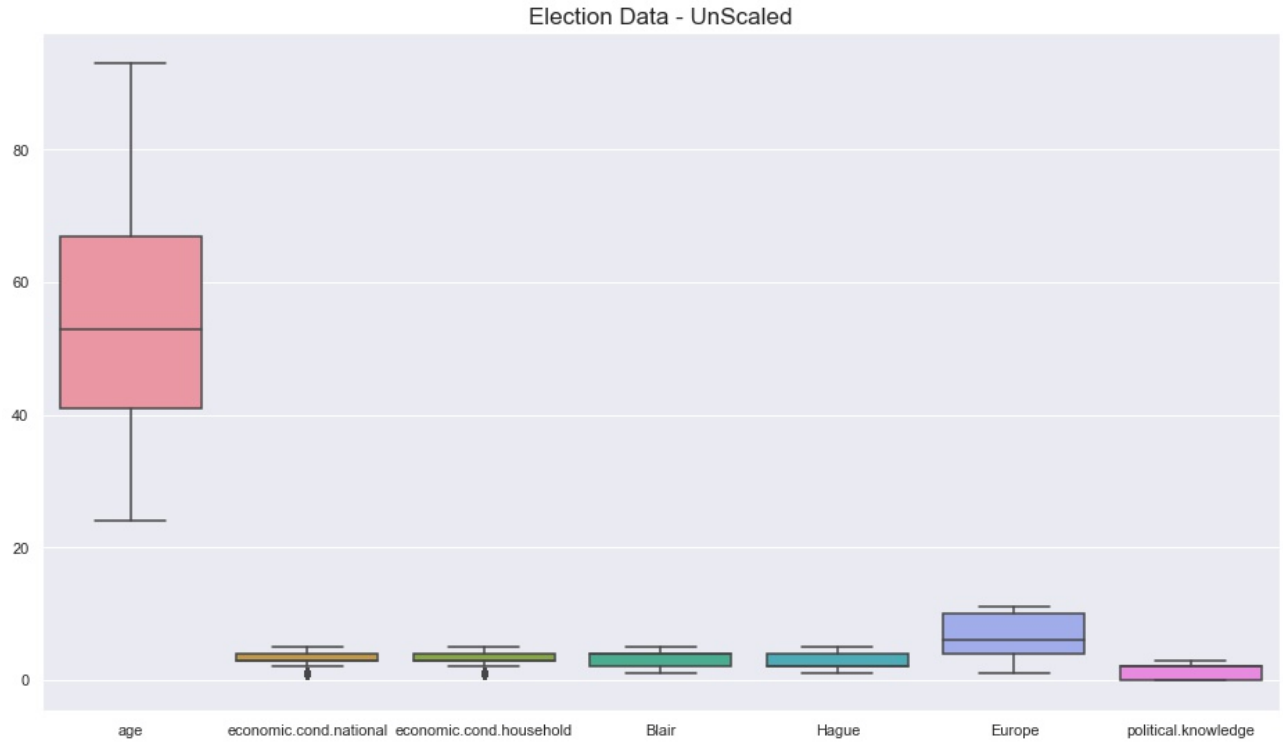
- These 2 Variables have 2 levels each and are Nominal variables i.e. they don't have an Order from Low to High.
- We can use One Hot Encoding or Categorical Encoding
- We'll use Pandas Categorical Codes method - This method generates random codes
- Given below, Levels and its corresponding codes -

```
feature: vote
['Labour', 'Conservative']
Codes :
[1 0]


feature: gender
['female', 'male']
Codes :
[0 1]
```

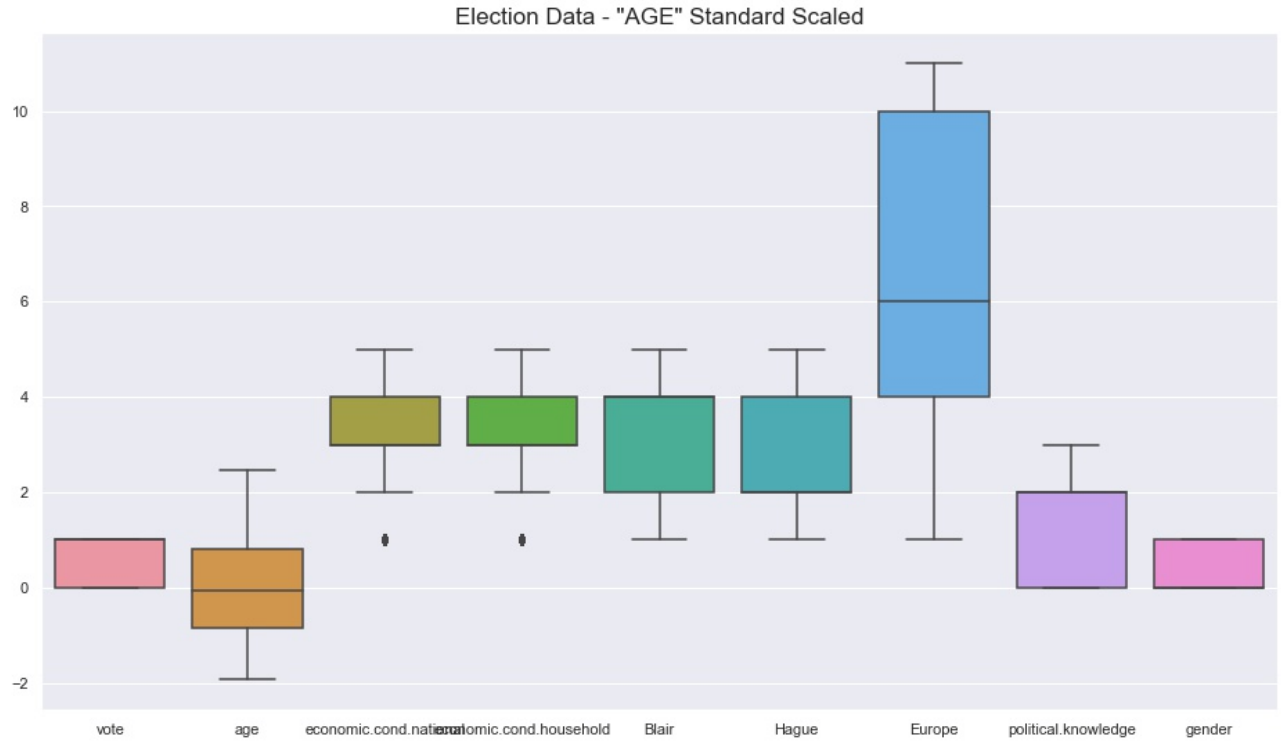
✦ Scaling -

- Scaling all variables to one scale helps in standardising the data to one scale



- If we use Zscore Scaling (same as StandardScaler), then unit of all variables becomes standard deviation

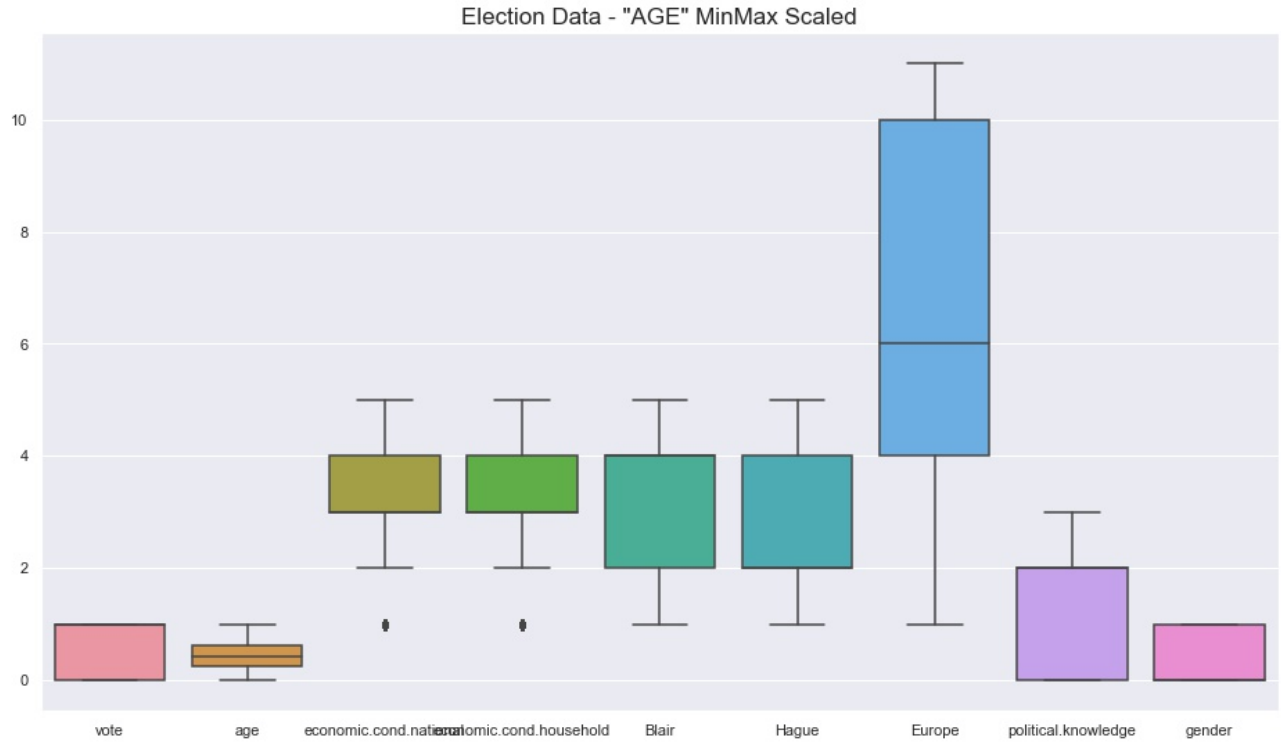
  One unit on scaled data == One Standard Deviation in the original data



- If we use MinMax Scaling, then

  Scaled Values near 0 indicate actual values near its Minimum

  Scaled Values near 1 indicate actual values near its Maximum



- **Classification Models which use Distance Measure algorithms** like KNN, SVM and maybe Logistic Regression **need Scaling** to avoid classification bias
- Whereas, **other Classification Models** like LDA, Naive Bayes, Boosting, Bagging, etc are **invariant to Scaling**
- **But, Models trained on scaled data are generally known to have significantly higher performance compared to the models trained on unscaled data.**
- Hence Scaling of data is preferred
- Here, in our Election data - only 'AGE' variable is continuous and all other variables are ratings - which are ordinal categorical variables
- Here, **we only scale the continuous 'AGE' variable**
- We scale 2 ways and compare Performance for each model -
    - MinMax Scaling of 'AGE
    - ZScore (Standard Scaling) of 'AGE'
- We found most models showing no performance difference due to different scaling
- Some models performed marginally better with Standard Scaled data

✦ Train and Test Split -

- Final dataset for modelling has **1517 observations and 9 variables**
- We use Encoded data for All Classification Modelling
- We assign **8 Independent Variables** (all except 'vote') to **X**
- We assign **1 Dependent Variable 'vote' to y**
- We perform a **70:30 - Train:Test Split** - with multiple seeds of Random States
- To maintain class balance of 70-30 in favour of Labour - as per the original data, we try for different seeds. We finalise on random_state=23
- **Train data - 1061 observations**
- **Test data - 456 observations**

[Q 1.4] Apply Logistic Regression and LDA (Linear Discriminant Analysis) (3 pts). Interpret the inferences of both models (2 pts)

✦ Logistic Regression (Logit) -

- We call Sklearn Classifier - LogisticRegression()
- We build **2 Logit models** -
  - Logit Model with **Default settings**
  - Logit Model with **best parameters through GridSearchCV**
- We **fine tune** these 2 models by selecting **best Probability Threshold** by comparing **Train Accuracy Scores**.
- Logit Model with **default settings** is built on 2 datasets -
  - **MinMax Scaled** Data and
  - ZScored - **Standard Scaled** Data
- Logit Model with **best parameters through GridSearchCV** is built on -
  - **Only Standard Scaled** Data as this was seen to perform better
- We perform **GridSearchCV** over following attributes over **10 folds** -

```
'penalty': ['l2','none', 'l1'],
'solver': ['lbfgs','liblinear', 'sag', 'saga', 'newton-cg'],
'tol': [0.0001,0.00001],
 max_iter = 10000, n_jobs=2,
 scoring = 'f1'
```

- **Best parameters** through GridSearchCV were found to be -

```
max_iter=10000,
n_jobs=2,
penalty='l1',
solver='saga'
penalty = 0.0001,
scoring = 'f1'
```

- But, the **Logit Model with default settings performed better** on metrics
- Hence, **we finalise Logit Model with default settings and Threshold = 0.5 to take ahead for final All Model Comparison**.

✦ Linear Discriminant Analysis (LDA) -

- We call Sklearn Classifier - LinearDiscriminantAnalysis()
- We build **2 LDA models on 2 differently Scaled Datasets** -
  - LDA Model on **MinMax Scaled** data
  - LDA Model on **Standard Scaled** data
- We **fine tune** these 2 models by **selecting best Probability Threshold** by comparing **Train Accuracy Scores**.
- We find that **difference in scaling has no impact on model performance**. Both models have exactly the same Performance Metrics.
- **Best Threshold** was found to be **0.5** - this gave the best Test Accuracy Score
- We **finalise LDA Model on Standard Scaled Data with Threshold = 0.5 to take ahead for final All Model Comparison**.

✦ Inference on Final Logit and LDA Models -

- We find the final **Logit Model performing marginally better** than the final LDA model
- Scores as follows -

Logit Train = 0.8172,	LDA Train = 0.8191
Logit Test = 0.8821,	**LDA Test = 0.8799**
Logit f1 for Labour = 0.9189,	LDA f1 for Labour = 0.9170
Logit f1 for Conservative = 0.784,	LDA f1 for Conservative = 0.7826

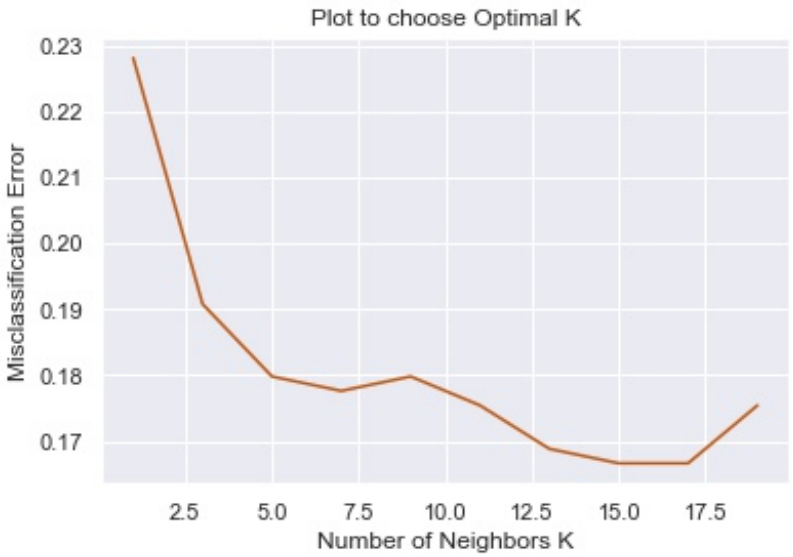
- Also, LDA makes more assumptions about the underlying data whereas **Logit is more robust and forgiving when the assumptions are violated**
- **Logit** can also crudely give us the **important attributes** for classification.
- Considering all above, **we'll choose Final Logistic Regression Model** out of the two

[Q 1.5] Apply KNN Model and Naïve Bayes Model(5 pts). Interpret the inferences of each model (2 pts)

✦ K-Nearest Neighbour (KNN) -

- We call Sklearn Classifier - KNearestNeighbor()
- We build 3 KNN **models** -
  - KNN Model on **MinMax Scaled Data (Default K=5)**
  - KNN Model on **Standard Scaled Data (Default K=5)**
  - KNN Model on **MinMax Scaled Data with Optimal K=15**
- We **fine tune** these 3 models by selecting **best Probability Threshold** by comparing Train Accuracy Scores.
- **MinMax Scaled KNN Model performed marginally better** than Standard Scaled
- Hence, for choosing optimal value of K, we use MinMax Scaled data
- We find Mis-Classification Errors (MCE) on Test Data for K=1, 3, 5, 7, 9, 11, 13, 15, 17, 19

  MCE = 1 - Accuracy Score
- Best K is the one which has the least Error, refer the plot below -



- Optimal K = 15 & 17. Lets **choose K = 15**
- We finalise **KNN Model on MinMax Scaled data with optimal K=15 and Threshold = 0.5 to take ahead for Final All Model Comparison.**

✦ Naive Bayes (NB) -

- We call Sklearn Classifier - GaussianNB()
- We build **2 NB models on 2 differently Scaled Datasets** -
  - NB Model on **MinMax Scaled** data
  - NB Model on **Standard Scaled** data
- We **fine tune** these 2 models by **selecting best Probability Threshold** by comparing **Train Accuracy Scores**.
- We find that **difference in scaling has no impact on model performance**. Both models have exactly the same Performance Metrics.
- **Best Threshold** was found to be **0.4** - this gave the best Train Accuracy Score. We Calculate all other Performance Metrics for this threshold
- We **finalise NB Model on Standard Scaled Data with Threshold = 0.4 to take ahead for final All Model Comparison**.

✦ Inference on Final KNN and NB Models -

- **KNN and NB have almost the same metrics**, with KNN performing just marginally better
- Scores as follows -

KNN Train = 0.8341,	NB Train = 0.8369
KNN Test = 0.8289,	**NB Test = 0.8224**
KNN f1 for Labour = 0.8796,	NB f1 for Labour = 0.8767
KNN f1 for Conservative = 0.7045,	NB f1 for Conservative = 0.6824

- **KNN is a Lazy Learner algorithm** as it does its learning right from the start at every iteration.
- While **NB is an Eager Learner algorithm**. It builds its way up at every iteration as it uses Bayes Theorem to calculate Posterior (Consecutive) probabilities.
- **KNN** gets **computationally more intensive** for larger data whereas NB is much faster here
- O**ur data has weak correlation and mostly categorical variables which works the best for NB**

- Considering all above, **we'll choose Naive Bayes Model** out of the two

[Q 1.6] Model Tuning (2 pts) , Bagging (2.5 pts) and Boosting (2.5 pts).

✦ Adaptive Boosting (AdaBoost) -

- We call Sklearn Classifier - AdaBoostClassifier()
- We build **2 AdaBoost models on 2 differently Scaled Datasets** -
  - AdaBoost Model on **MinMax Scaled** data
  - AdaBoost Model on **Standard Scaled** data
- We **fine tune** these 2 models by **selecting best Probability Threshold** by comparing **Train Accuracy Scores**.
- We find that **difference in scaling has no impact on model performance**. Both models have exactly the same Performance Metrics.
- **Best Threshold** was found to be **0.5**
- We **finalise AdaBoost Model on Standard Scaled Data with Threshold = 0.5 to take ahead for final All Model Comparison**.

✦ Gradient Boosting (GradBoost) -

- We call Sklearn Classifier - GradientBoostingClassifier()
- We build **2 GradBoost models on 2 differently Scaled Datasets** -
  - GradBoost Model on **MinMax Scaled** data
  - GradBoost Model on **Standard Scaled** data
- We **fine tune** these 2 models by **selecting best Probability Threshold** by comparing **Train Accuracy Scores**.
- We find that **difference in scaling has no impact on model performance**. Both models have exactly the same Performance Metrics.
- **Best Threshold** was found to be **0.5**
- We **finalise GradBoost Model on Standard Scaled Data with Threshold = 0.5 to take ahead for final All Model Comparison**.

✦ Bagging using Random Forest (RF) -

- We call Sklearn Classifier - RandomForestClassifier()
- We build **2 RF models** -

  - RF Model with **Default settings**
  - RF Model with **best parameters through GridSearchCV**

- We **fine tune** these 2 models by selecting **best Probability Threshold** by comparing **Train Accuracy Scores**.
- RF Model with **default settings** is built on 2 datasets -

  - **MinMax Scaled** Data and
  - ZScored - **Standard Scaled** Data

- RF Model with **best parameters through GridSearchCV** is built on -

  - **Only Standard Scaled** Data as this was seen to perform better

- We perform **GridSearchCV** over following attributes over **10 folds** -

```
'n_estimators': [51, 201, 301, 501],
'criterion': ['gini', 'entropy'],
'min_samples_leaf': [10, 20, 50, 100],
'min_samples_split': [30 ,60, 150, 300],
'max_features': [‘log2’, 3, 4]
```

- **Best parameters** through GridSearchCV were found to be -

```
'criterion': 'gini',
'max_features': ‘log2’,
'min_samples_leaf': 20,
'min_samples_split': 30,
'n_estimators': 51
```

- **RF Model with default settings was Overfitted** as expected (Train Score = 0.991, Test Score = 0.8224) - **hence rejecting** this
- **Regularised RF model** has **good consistent performance** over train and test both
- **Best Threshold** was found to be **0.6**
- Hence, **we finalise Regularised RF Model and Threshold = 0.6 to take ahead for final All Model Comparison**.

✦ Model Tuning -

- Parameter tuning for each model is done and details mentioned above individually in explanations of each model above.

[Q 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 (4 pts) Final Model - Compare all models on the basis of the performance metrics in a structured tabular manner. Describe on which model is best/ optimized (3 pts)

✦ Logistic Regression -

- Best Threshold = 0.5
- TRAIN Data:

  Accuracy: 84.35%

  AUC: 89.35%

  f1-Score for Labour: 89.01%

  f1-Score for Conservative: 72.88%

- TEST Data:

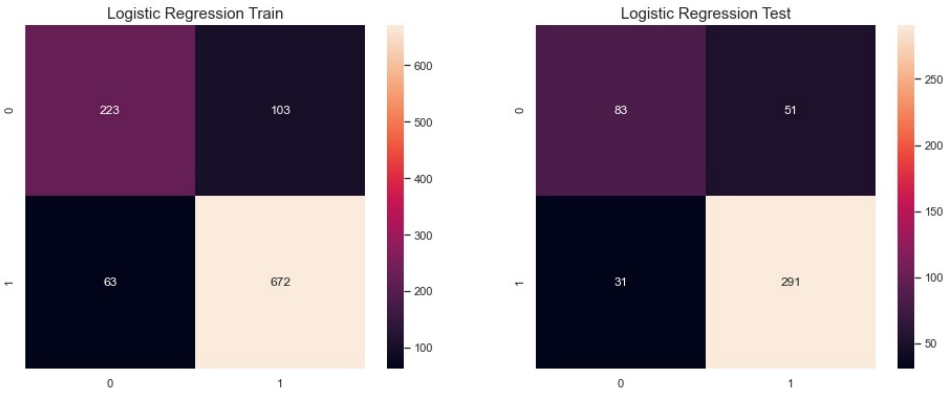
  Accuracy: 82.02%

  AUC: 87.31%

  f1-Score for Labour: 87.65%

  f1-Score for Conservative: 66.94%

- Confusion Matrix :



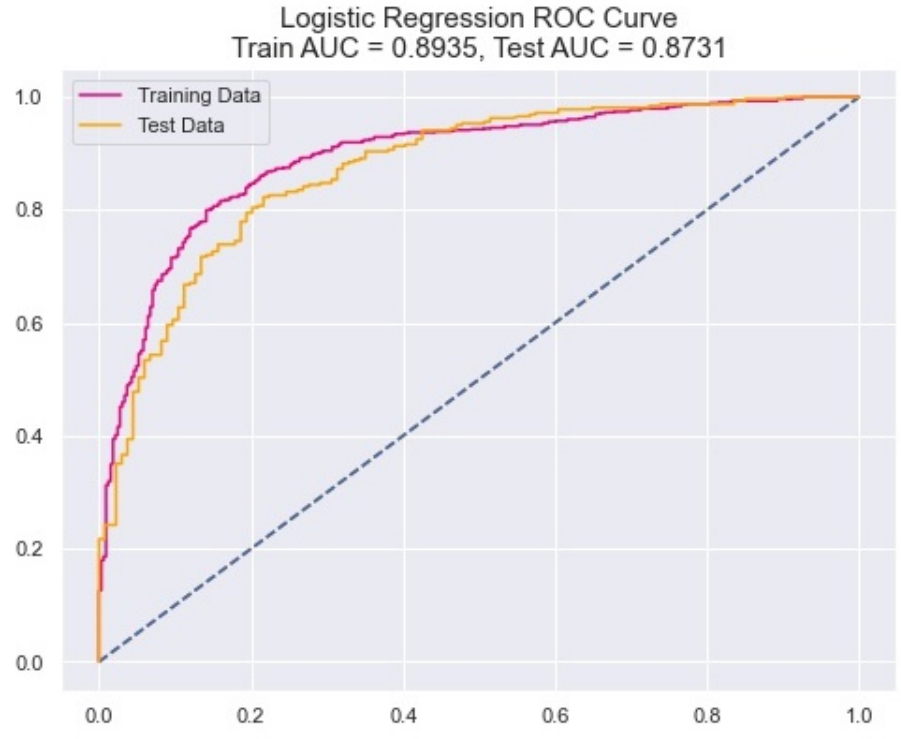
- Classification Report - Train :

	precision	recall	f1-score	support
0	0.7797	0.6840	0.7288	326
1	0.8671	0.9143	0.8901	735
accuracy	0.8435	0.8435	0.8435	0.8435
macro avg	0.8234	0.7992	0.8094	1061
weighted avg	0.8402	0.8435	0.8405	1061

- Classification Report - Test :

	precision	recall	f1-score	support
0	0.7281	0.6194	0.6694	134
1	0.8509	0.9037	0.8765	322
accuracy	0.8202	0.8202	0.8202	0.8202
macro avg	0.7895	0.7616	0.7729	456
weighted avg	0.8148	0.8202	0.8156	456

- ROC Curve :



✦ Linear Discriminant Analysis -

- Best Threshold = 0.5
- TRAIN Data:

  Accuracy: 84.64%

  AUC: 89.32%

  f1-Score for Labour: 89.08%

  f1-Score for Conservative: 74.09%

- TEST Data:

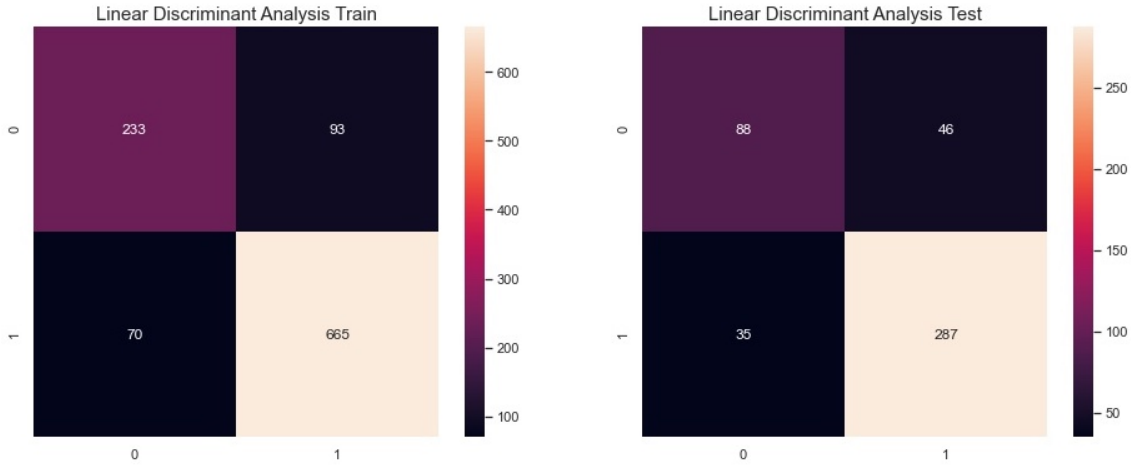
  Accuracy: 82.24%

  AUC: 87.44%

  f1-Score for Labour: 87.63%

  f1-Score for Conservative: 68.48%

- Confusion Matrix :



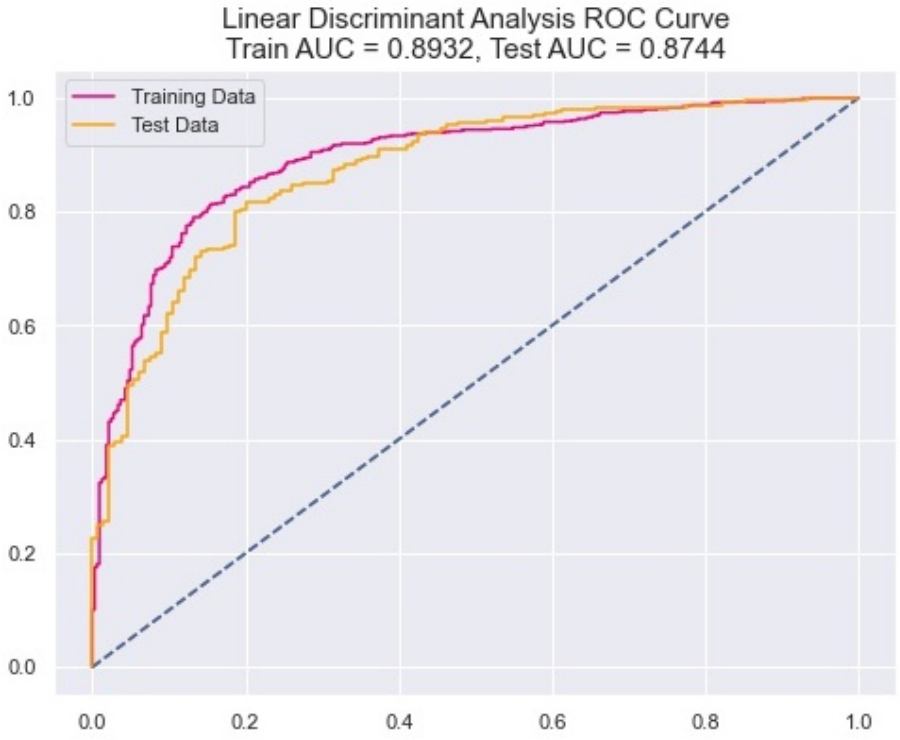
- Classification Report - Train :

	precision	recall	f1-score	support
0	0.7690	0.7147	0.7409	326
1	0.8773	0.9048	0.8908	735
accuracy	0.8464	0.8464	0.8464	0.8464
macro avg	0.8231	0.8097	0.8158	1061
weighted avg	0.8440	0.8464	0.8447	1061

- Classification Report - Test :

	precision	recall	f1-score	support
0	0.7154	0.6567	0.6848	134
1	0.8619	0.8913	0.8763	322
accuracy	0.8224	0.8224	0.8224	0.8224
macro avg	0.7887	0.7740	0.7806	456
weighted avg	0.8188	0.8224	0.8201	456

- ROC Curve :



✦ K-Nearest Neighbour -

- Best Threshold = 0.5
- Optimal K = 15 (gives least Mis-Classification Error)
- TRAIN Data:

  Accuracy: 83.79%

  AUC: 91.42%

  f1-Score for Labour: 88.46%

  f1-Score for Conservative: 72.78%

- TEST Data:

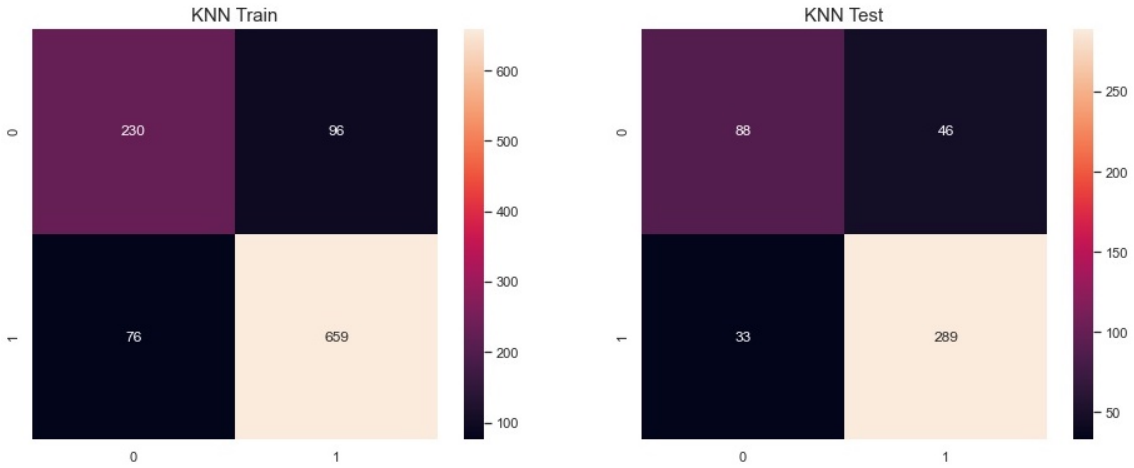
  Accuracy: 82.68%

  AUC: 87.40%

  f1-Score for Labour: 87.98%

  f1-Score for Conservative: 69.02%

- Confusion Matrix :

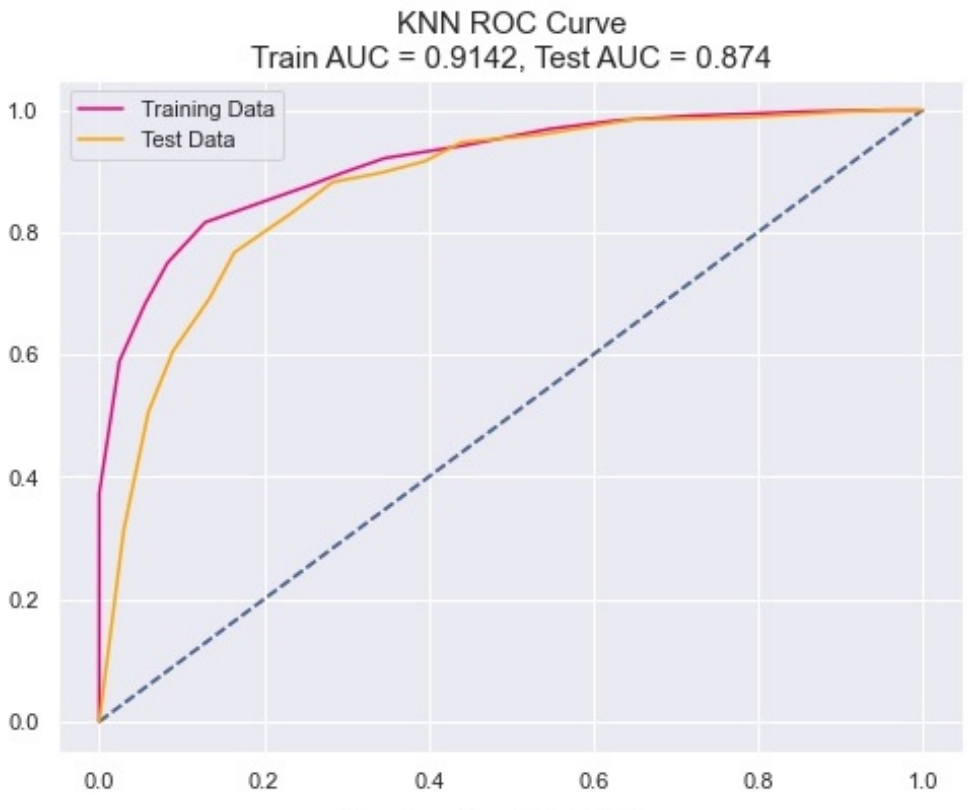


- Classification Report - Train :

	precision	recall	f1-score	support
0	0.7516	0.7055	0.7278	326
1	0.8728	0.8966	0.8846	735
accuracy	0.8379	0.8379	0.8379	0.8379
macro avg	0.8122	0.8011	0.8062	1061
weighted avg	0.8356	0.8379	0.8364	1061

- Classification Report - Test :

	precision	recall	f1-score	support
0	0.7273	0.6567	0.6902	134
1	0.8627	0.8975	0.8798	322
accuracy	0.8268	0.8268	0.8268	0.8268
macro avg	0.7950	0.7771	0.7850	456
weighted avg	0.8229	0.8268	0.8241	456



✦ Adaptive Boost -

- Best Threshold = 0.5
- TRAIN Data:

  Accuracy: 85.67%

  AUC: 91.55%

  f1-Score for Labour: 89.80%

  f1-Score for Conservative: 75.95%

- TEST Data:

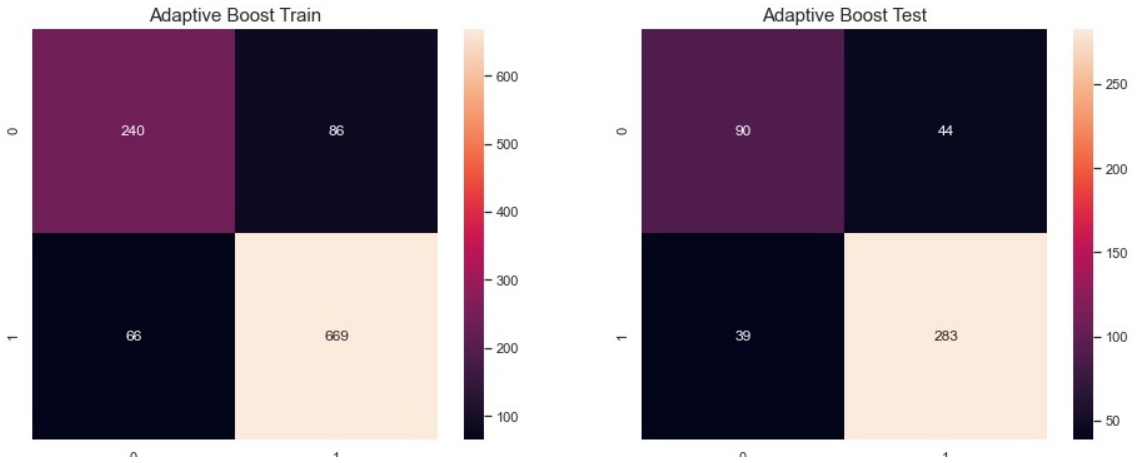
  Accuracy: 81.80%

  AUC: 84.35%

  f1-Score for Labour: 87.21%

  f1-Score for Conservative: 68.44%

- Confusion Matrix :



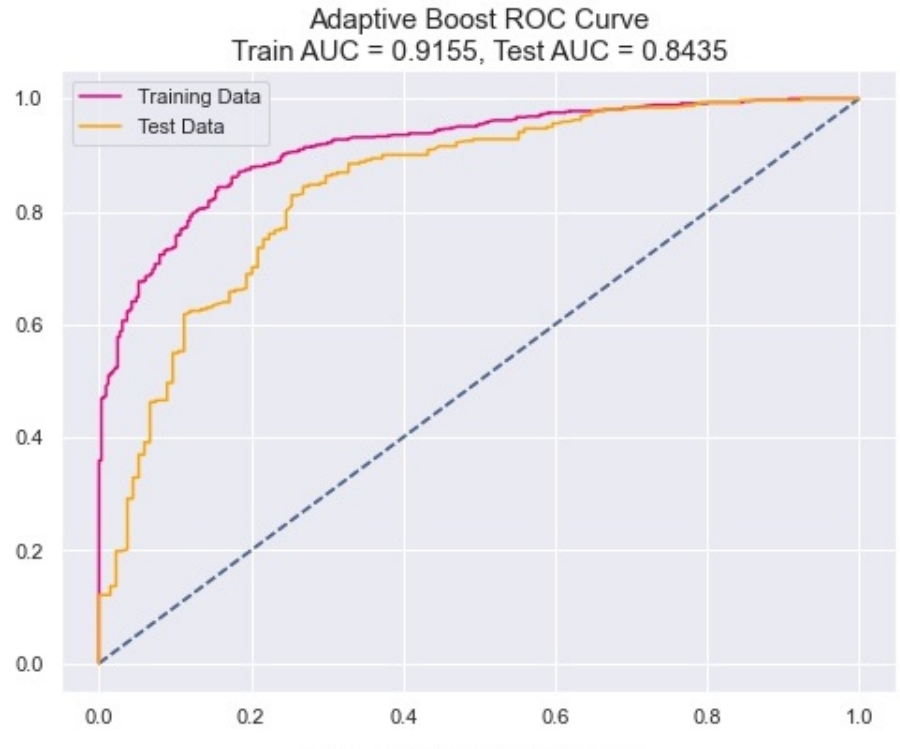
- Classification Report - Train :

	precision	recall	f1-score	support
0	0.7843	0.7362	0.7595	326
1	0.8861	0.9102	0.8980	735
accuracy	0.8567	0.8567	0.8567	0.8567
macro avg	0.8352	0.8232	0.8287	1061
weighted avg	0.8548	0.8567	0.8554	1061

- Classification Report - Test :

	precision	recall	f1-score	support
0	0.6977	0.6716	0.6844	134
1	0.8654	0.8789	0.8721	322
accuracy	0.8180	0.8180	0.8180	0.8180
macro avg	0.7816	0.7753	0.7783	456
weighted avg	0.8161	0.8180	0.8170	456

- ROC Curve :



✦ <u>Gradient Boost -</u>

- Best Threshold = 0.5
- TRAIN Data:

  Accuracy: 89.07%

  AUC: 95.34%

  f1-Score for Labour: 92.23%

  f1-Score for Conservative: 81.59%

- TEST Data:

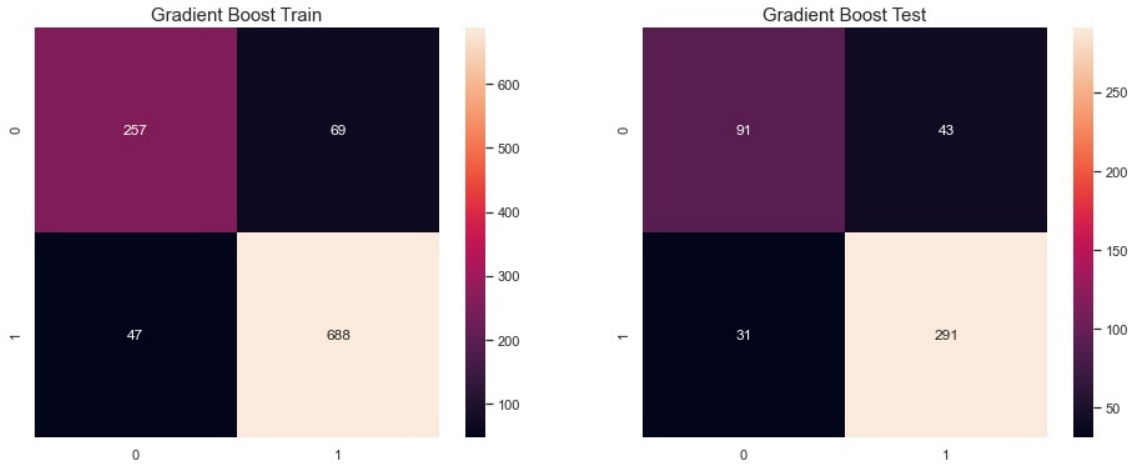
  Accuracy: 83.77%

  AUC: 88.05%

  f1-Score for Labour: 88.72%

  f1-Score for Conservative: 71.09%

- Confusion Matrix :



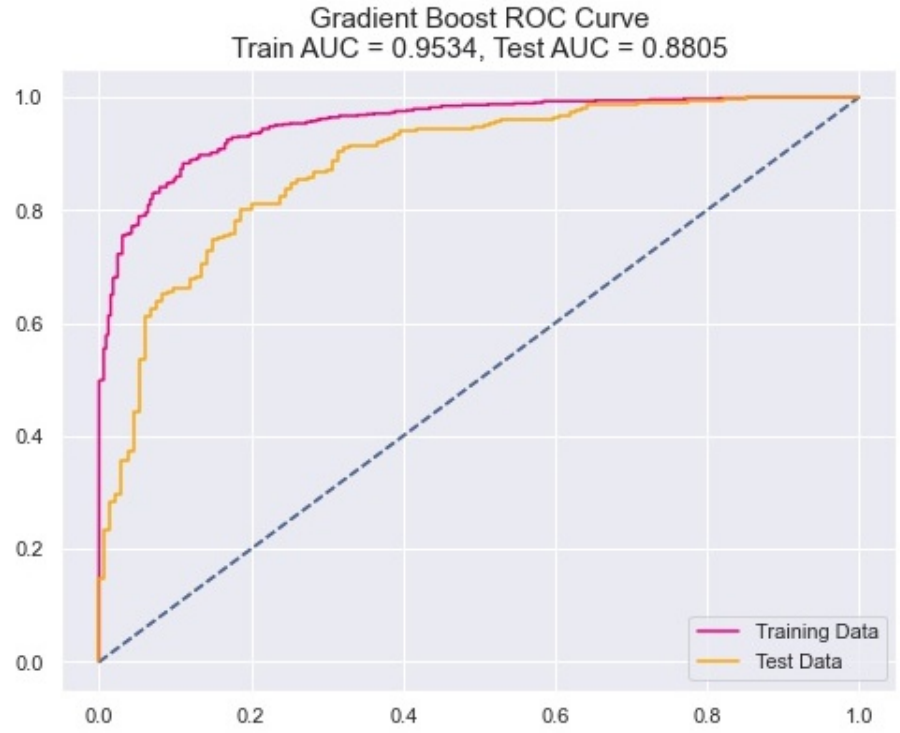
- Classification Report - Train :

	precision	recall	f1-score	support
0	0.8454	0.7883	0.8159	326
1	0.9089	0.9361	0.9223	735
accuracy	0.8907	0.8907	0.8907	0.8907
macro avg	0.8771	0.8622	0.8691	1061
weighted avg	0.8894	0.8907	0.8896	1061

- Classification Report - Test :

	precision	recall	f1-score	support
0	0.7459	0.6791	0.7109	134
1	0.8713	0.9037	0.8872	322
accuracy	0.8377	0.8377	0.8377	0.8377
macro avg	0.8086	0.7914	0.7991	456
weighted avg	0.8344	0.8377	0.8354	456

- ROC Curve :



✦ Naive Bayes -

- Best Threshold = 0.4
- TRAIN Data:

  Accuracy: 83.69%

  AUC: 89.01%

  f1-Score for Labour: 88.52%

  f1-Score for Conservative: 71.87%

- TEST Data:

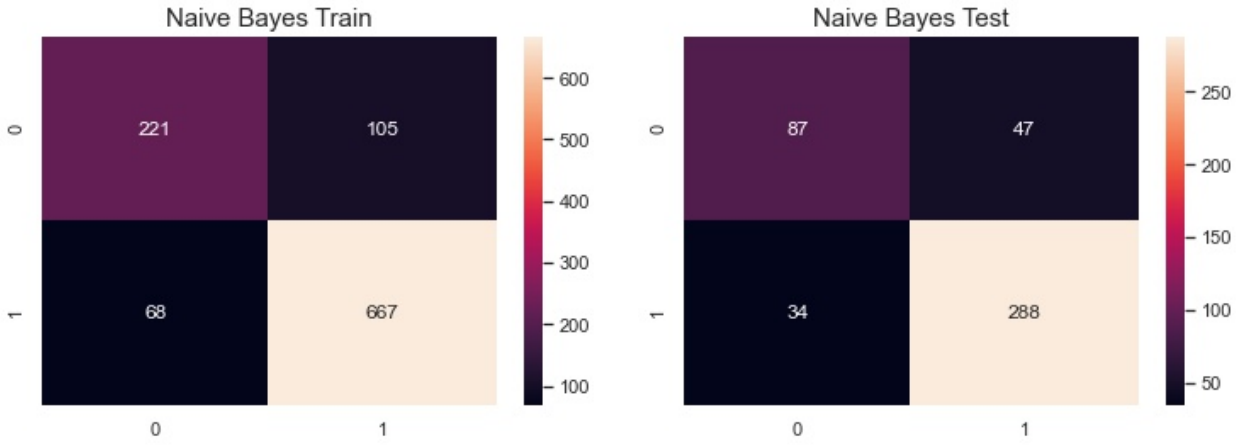
  Accuracy: 82.24%

  AUC: 89.01%

  f1-Score for Labour: 87.67%

  f1-Score for Conservative: 68.24%

- Confusion Matrix :



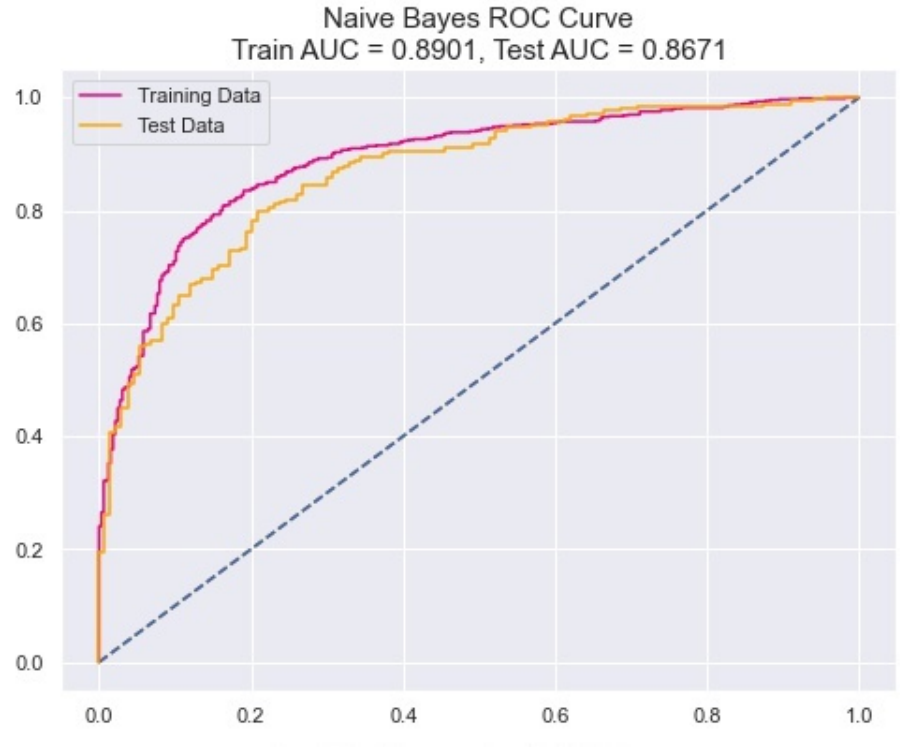
- Classification Report - Train :

	precision	recall	f1-score	support
0	0.7647	0.6779	0.7187	326
1	0.8640	0.9075	0.8852	735
accuracy	0.8369	0.8369	0.8369	0.8369
macro avg	0.8143	0.7927	0.8020	1061
weighted avg	0.8335	0.8369	0.8340	1061

- Classification Report - Test :

	precision	recall	f1-score	support
0	0.7190	0.6493	0.6824	134
1	0.8597	0.8944	0.8767	322
accuracy	0.8224	0.8224	0.8224	0.8224
macro avg	0.7894	0.7718	0.7795	456
weighted avg	0.8184	0.8224	0.8196	456

- ROC Curve :



✦ <u>Bagging - Random Forest</u> -

- Best Threshold = 0.6
- TRAIN Data:

  Accuracy: 84.07%

  AUC: 91.26%

  f1-Score for Labour: 88.22%

  f1-Score for Conservative: 75.40%

- TEST Data:

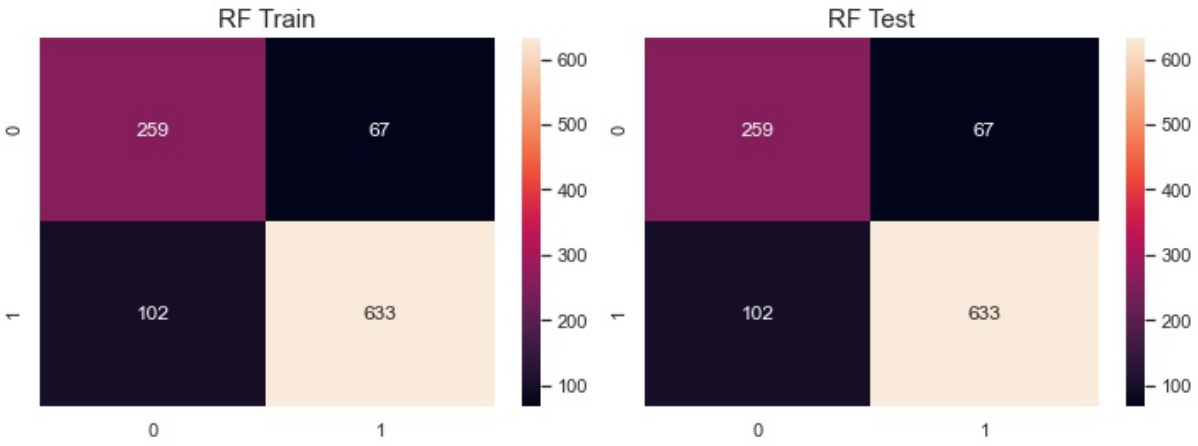
  Accuracy: 82.89%

  AUC: 91.26%

  f1-Score for Labour: 87.62%

  f1-Score for Conservative: 72.34%

- Confusion Matrix :



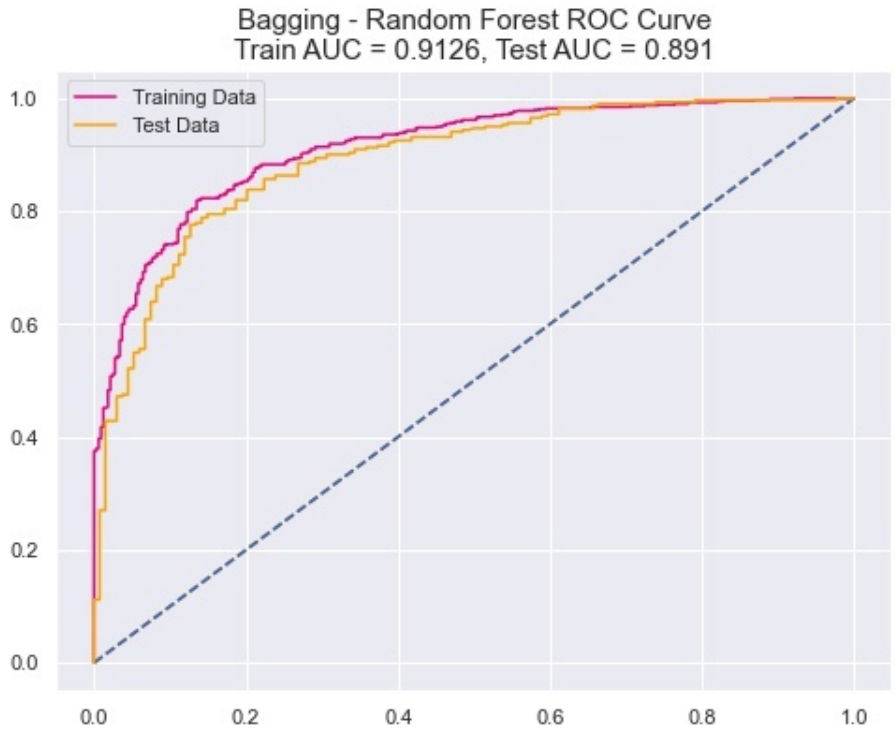
- Classification Report - Train :

	precision	recall	f1-score	support
0	0.7175	0.7945	0.7540	326
1	0.9043	0.8612	0.8822	735
accuracy	0.8407	0.8407	0.8407	0.8407
macro avg	0.8109	0.8279	0.8181	1061
weighted avg	0.8469	0.8407	0.8428	1061

- Classification Report - Test :

	precision	recall	f1-score	support
0	0.6892	0.7612	0.7234	134
1	0.8961	0.8571	0.8762	322
accuracy	0.8289	0.8289	0.8289	0.8289
macro avg	0.7926	0.8092	0.7998	456
weighted avg	0.8353	0.8289	0.8313	456

- ROC Curve :



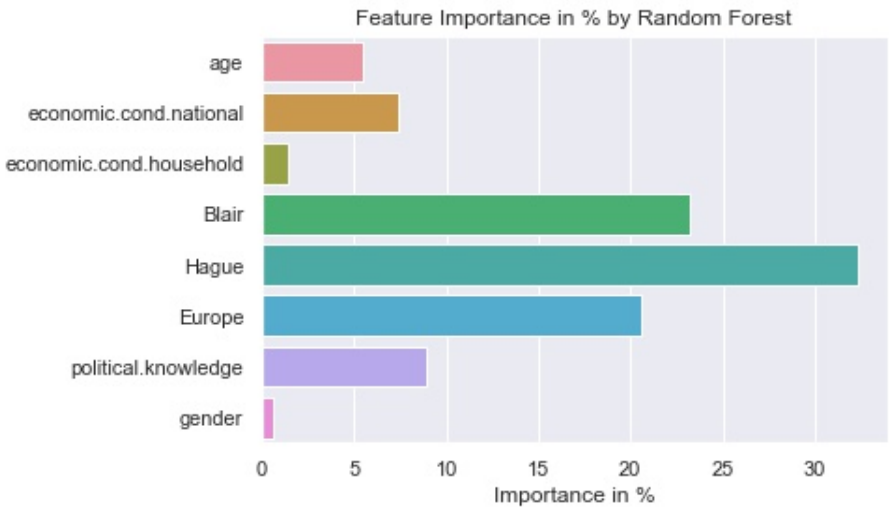
✦ ALL MODEL PERFORMANCE -

	Best Threshold	Test Score	Test AUC	Train Score	Train AUC	Test f1 Score Labour	Test f1 Score Conservative
Gradient Boosting	0.5	0.8377	0.8805	0.8907	0.9534	0.8872	0.7109
Bagging (Random Forest Regularised)	**0.6**	**0.8289**	**0.9126**	**0.8407**	**0.9126**	**0.8762**	**0.7234**
KNN for K=15	0.5	0.8268	0.8740	0.8379	0.9142	0.8798	0.6902
Naive Bayes	0.4	0.8224	0.8901	0.8369	0.8901	0.8767	0.6824
Linear Discriminant Analysis	0.5	0.8224	0.8744	0.8464	0.8932	0.8763	0.6848
Logistic Regression	0.5	0.8202	0.8731	0.8435	0.8935	0.8765	0.6694
Adaptive Boosting	0.5	0.8180	0.8435	0.8567	0.9155	0.8721	0.6844

- In this Problem Statement, we are building Exit Polls and predicting which party would the voters vote.
- Here we don't have any Class of Interest - Both Classes are equally important
- Also, data is 70:30 balanced, which is sufficient to not create bias in Classification Models
- Hence, we **focus on Accuracy Scores and ROC AUC Scores** to select the best model for production
- **Top 3 models** sorted by Test Scores and then Test AUC are -
    - **Gradient Boosting (GB)**
    - **Bagging - Random Forest (RF)**
    - **KNN for K=15**
- Though GB has slightly better Accuracy Score than RF but RF has a better AUC Score. So they both are neck to neck in comparison
- KNN is a lazy learner algorithm and gets computationally intensive, hence we would ignore this
- But we know that, **Random Forest is easier to tune than Gradient Boost**
- Here, in this Problem Statement, **interpretability and feature importance** would greatly help in designing and predicting Exit Polls
- **Random Forest** outputs **Feature Importance** very well. It also gives weightage of each Feature individually
- Hence, for this reason, **we choose Regularised Random Forest with Threshold = 0.6 as the best optimised model. We'll deploy Random Forest for production**

[Q 1.8] Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective.

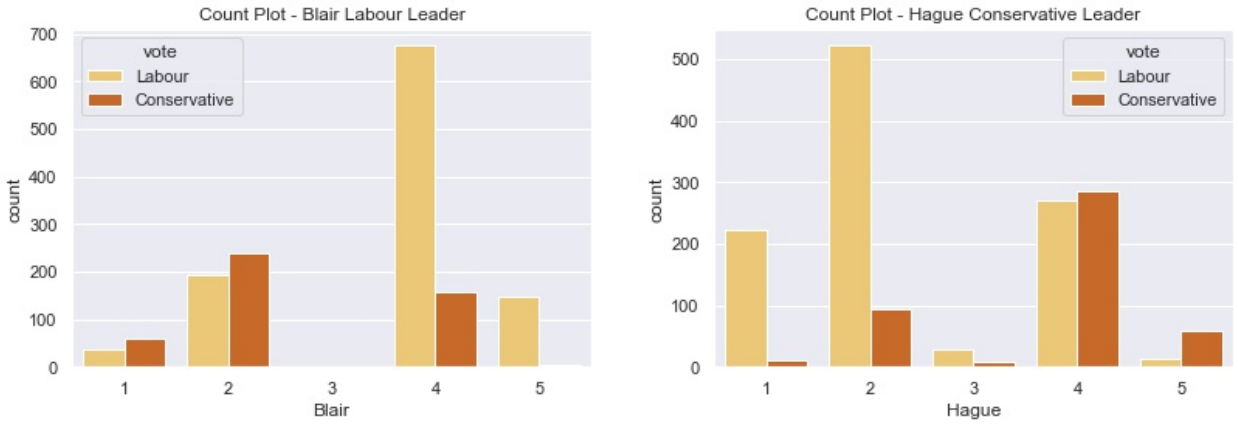
- We have chosen regularised Random Forest for production
- RF gives us the following importance of each Feature -



- As expected, Hague and Blair are the most important - wherein high approval rating for Blair will result in vote for Labour and similarly Hague to Conservative
- But if we note the following graphs, it is seen

  Higher Blair rating = almost surely vote for Labour

  But, it is not necessarily the same for Hague



- The **next 3 important features** for Voter Classification are ratings for -

  - Euro-scepticism

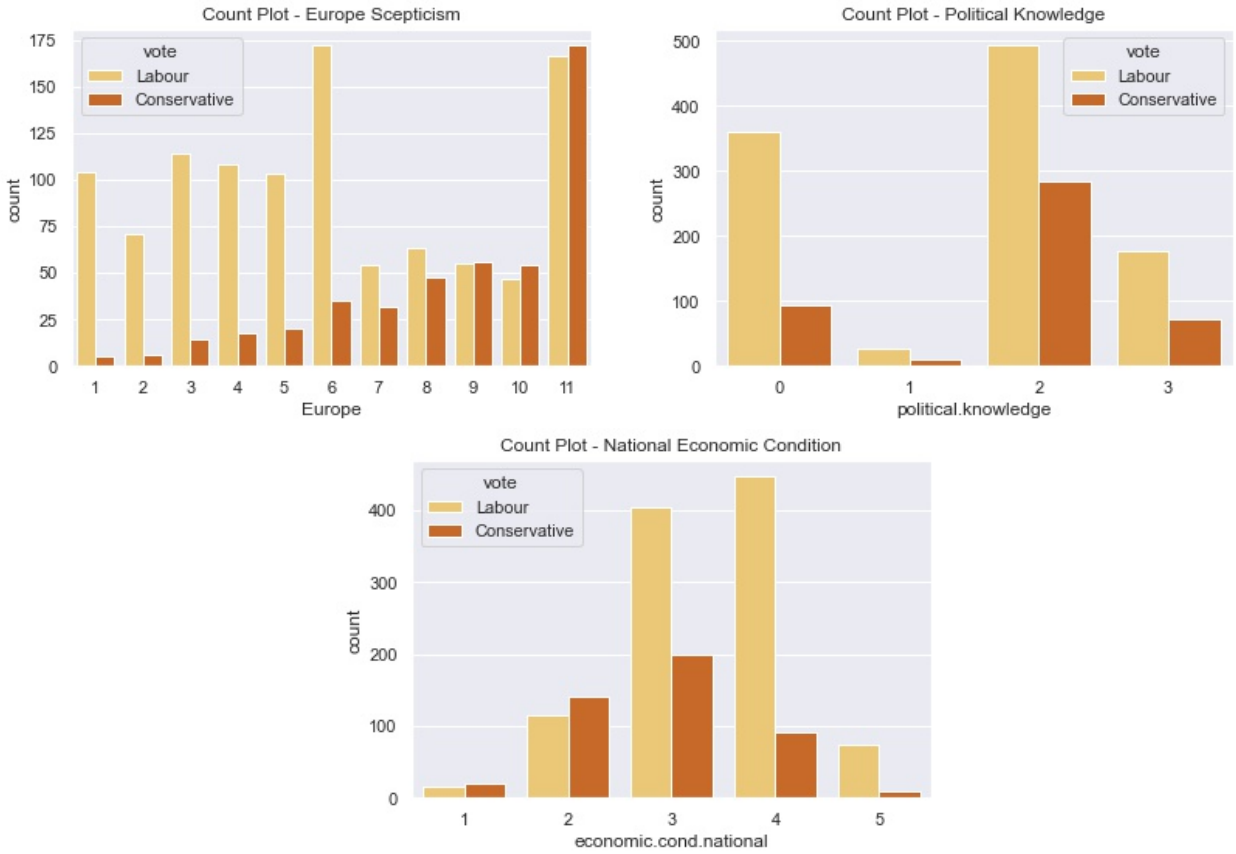
  - Political Knowledge and

  - National Economic Condition

- **Euro-sceptic** ratings of 6 and below = almost certain vote for Labour

  But, ratings of >6, both parties are 50-50 (refer figure on next page)

- Voters with higher ratings for **National Economic Condition** tend to vote for Labour more



- Voters with zero Political Knowledge are seen to prefer Labour party more than Conservatives.
- **Labour party voters can be bucketed as having -**
  - **High Blair ratings**
  - **Low Political Knowledge**
  - **Low Euro-sceptic sentiments**
  - **Think highly of present Economic National Condition**
- **Conservative party voters can be bucketed as having -**
  - **Perfect 5 Hague ratings**
  - **High Political Knowledge**
  - **High Euro-sceptic sentiments**
- CNBE can predict Exit Polls by considering the above characteristics of the polled voters

PROBLEM 2 - US Presidents' Speech

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

SYNOPSIS

- We perform Text Analytics on *Natural Language Tool Kit's* (nltk) Corpus of Inaugural Speeches by US Presidents
- This Inaugural Corpus has 58 speeches
- We use the following 3 speeches for this Problem Statement -
  - President Franklin D. Roosevelt in 1941
  - President John F. Kennedy in 1961
  - President Richard Nixon in 1973
- First few lines of Roosevelt's 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 institutions from disruption from without.\n\nTo us there has come a time, in the midst of swift happenings, to pause for a moment and take stock -- to recall what our place in history has been, and to rediscover what we are and what we may be.....'
- First few lines of Kennedy's 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 as 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……'

- First few lines of Nixon's Speech -

  'Mr. Vice President, Mr. Speaker, Mr. Chief Justice, Senator Cook, Mrs. Eisenhower, and my fellow citizens of this great and good country we share together:\n\nWhen we met here four years ago, America was bleak in spirit, depressed by the prospect of seemingly 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 about to enter will not be what other postwar periods have so often been: a time of retreat and isolation that leads to stagnation at home and invites new danger abroad…..'

[Q 2.1] Find the number of characters, words and sentences for the mentioned documents. – 3 Marks

- In non-cleaned raw speech of President Roosevelt -
  - Num of sentences in Raw Roosevelt Speech = 68
  - Num of words in Raw Roosevelt Speech = 1526
  - Num of characters in Raw Roosevelt Speech = 7466

- In non-cleaned raw speech of President Kennedy -
  - Num of sentences in Raw Kennedy Speech = 52
  - Num of words in Raw Kennedy Speech = 1543
  - Num of characters in Raw Kennedy Speech = 7540

- In non-cleaned raw speech of President Nixon -
  - Num of sentences in Raw Nixon Speech = 68
  - Num of words in Raw Nixon Speech = 2006
  - Num of characters in Raw Nixon Speech = 9874

- Hence, Total figures of all 3 non-cleaned raw speeches -
  - Total Num of sentences = 188

- Total Num of words = 3685
- Total Num of characters = 24880

[Q 2.2] Remove all the stop-words from all the three speeches. – 3 Marks

- Stopwords are those words which do not add any value or meaning to understanding of any text like - 'I', 'me', 'myself', 'have', 'you', 'your', etc
- We use list of English language Stopwords from NLTK to remove them from our speech. List of some NLTK Stopwords given below -

```
['ourselves', 'hers', 'between', 'yourself', 'but', 'again',
'there', 'about', 'once', 'during', 'out', 'very', 'having',
'with', 'they', 'own', 'an', 'be', 'some', 'for', 'do', 'its',
'yours', 'such', 'into', 'of', 'most', 'itself', 'other', 'off',
'is', 's', 'am', 'or', 'who', 'as', 'from', 'him', 'each', 'the',
'themselves', 'until', 'below', 'are', 'we', 'these',…]
```

- We can also add our Stopwords, specific to our content. Here, we are analysing speeches of USA Presidents, hence words like - 'usa', 'america', 'american', 'President', Vice President', etc may occur a large number of times without conveying any real meaning.

  Hence, we can add these to our list of Stopwords and remove from our Bag of Words for analysis

- Further, we do more Pre-processing -

  - Convert all words to lowercase

  - Stemming - is the process of reducing extensions in words to their root forms even if the stem itself is not a valid word in the Language

[Q 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)

- After removing all the standard Stopwords, top counts of each speech is given below

- **Roosevelt Speech**
  - Top 10 -

    ```
    {'--': 25, 'nation': 12, 'know': 10, 'spirit': 9, 'life': 9,
    'democracy': 9, 'us': 8, 'people': 7, 'america': 7, 'years':
    6, ...}
    ```

  - But we see that few words like - '—', 'know', 'us' add no value, hence, lets add them to Stopwords and remove them
  - **Final Top 3 of Roosevelt Speech -**

    ```
    'nation': 12,
    'know': 10,
    'spirit': 9
    ```

- **Kennedy Speech**
  - Top 10 -

    ```
    {'--': 25, 'let': 16, 'us': 12, 'world': 8, 'sides': 8,
    'new': 7, 'pledge': 7, 'citizens': 5, 'power': 5, 'shall': 5,
    ...}
    ```

  - But we see that few words like - '—', 'let', 'us', 'shall' add no value, hence, lets add them to Stopwords and remove them
  - **Final Top 3 of Kennedy Speech -**

    ```
    'world': 8,
    'sides': 8,
    'new': 7
    ```

- **Nixon Speech**
  - Top 10 -

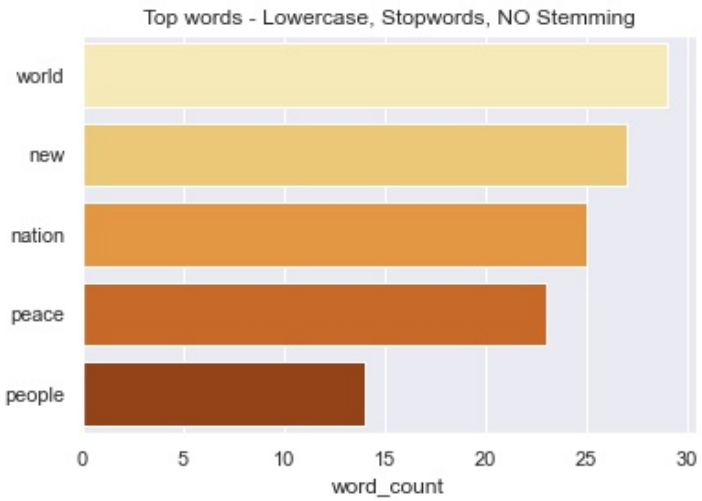
    ```
    {'us': 26, 'let': 22, 'america': 21, 'peace': 19, 'world':
    18, '--': 17, 'new': 15, 'nation': 11, 'responsibility': 11,
    'government': 10, ...}
    ```

  - But we see that few words like - '—', 'let', 'us', 'america' add no value, hence, lets add them to Stopwords and remove them
  - **Final Top 3 of Nixon Speech -**

    ```
    'peace': 19,
    'world': 18,
    'new': 15
    ```

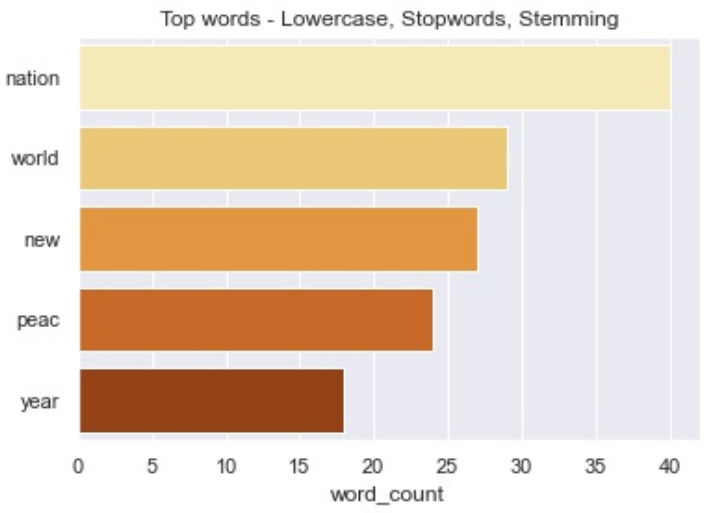
- Please note that ‘us’ can be a collective pronoun or acronym for ‘United States’ - both ways, it is not of value as this is a speech by American Presidents. It is but expected that they’ll talk about America.
- Hence, we also remove ‘america’
- **All 3 speeches - Removing Stopwords - Top 3**

```
'world': 29,
'new': 27,
'nation': 25
```



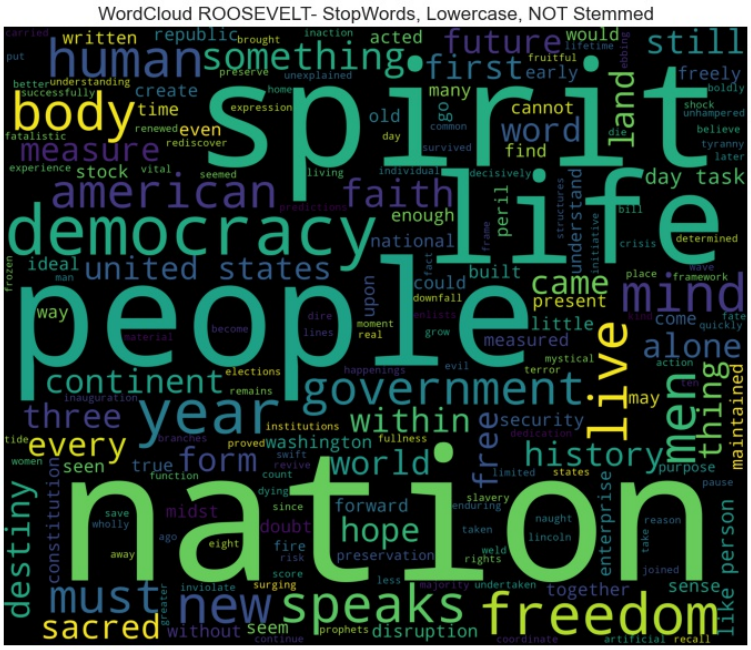
- **All 3 speeches - Removing Stopwords - After Stemming - Top 3**

```
‘nation’ : 40
‘world’ : 29
‘new’ : 27
```

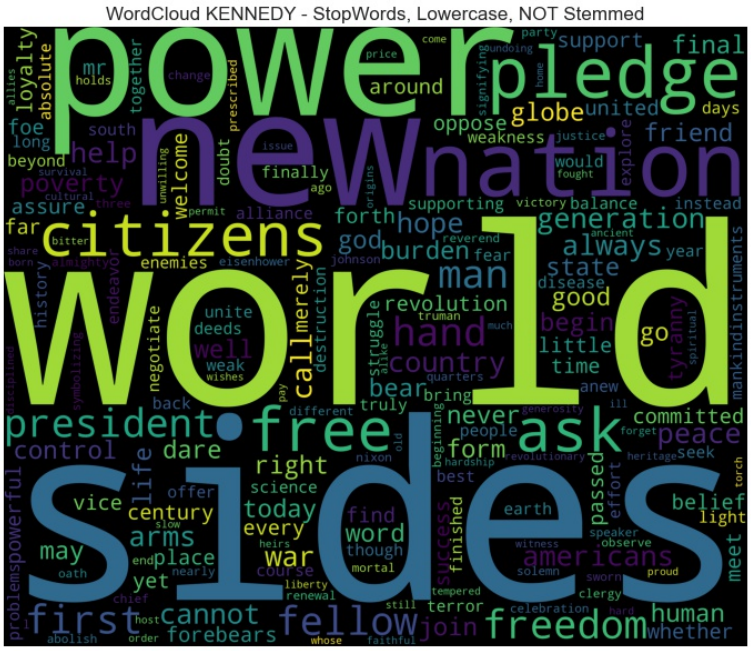


[Q 2.4] Plot the word cloud of each of the three speeches. (after removing the stopwords)

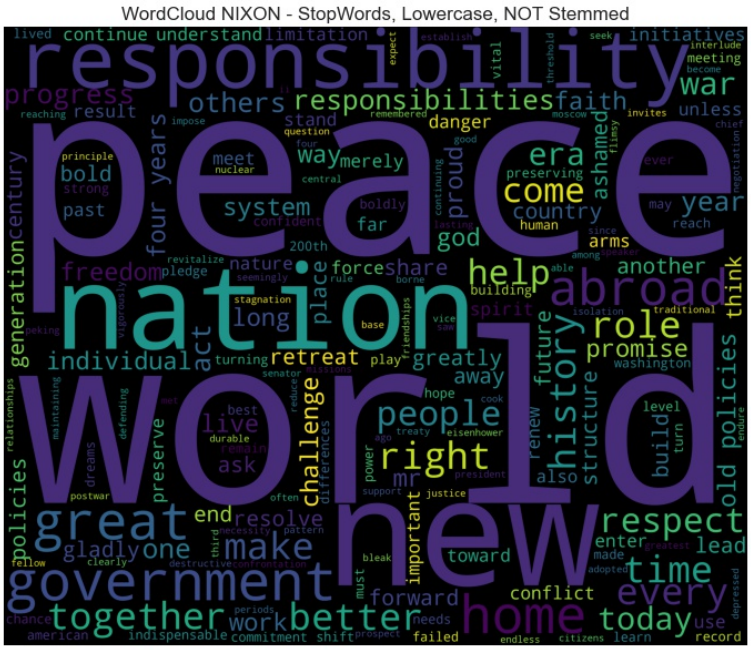
- **Roosevelt** Speech- Removing Stopwords - **Not Stemmed - Word-cloud**



- **Kennedy** Speech- Removing Stopwords - **Not Stemmed - Word-cloud**



- **Nixon** Speech- Removing Stopwords - **Not Stemmed - Word-cloud**



- **All 3** Speeches - Removing Stopwords - **Not Stemmed - Word-cloud**

————————- END OF PROJECT —————————-