Problem 1:

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.

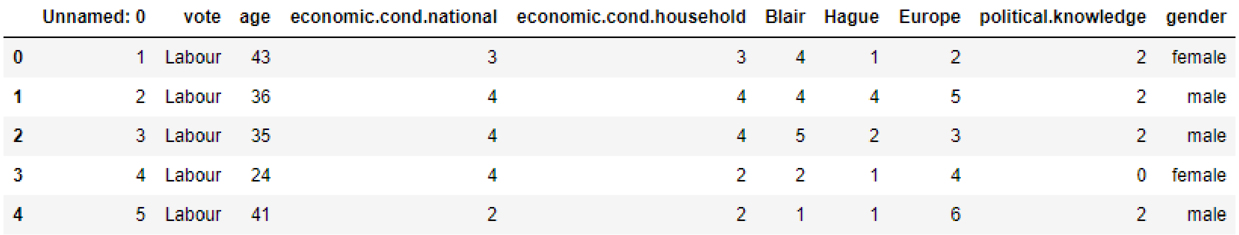
Dataset for Problem: Election_Data.xlsx

1. **Read the dataset. Do the descriptive statistics and do null value condition check. Write an inference on it. (5 Marks)**

   As we start reading data, we come to know there are 9 different attributes

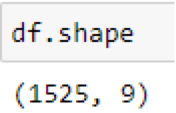
   - **vote**: Party choice: Conservative or Labour
   - **age**: in years
   - **economic.cond.national**: Assessment of current national economic conditions, 1 to 5.
   - **economic.cond.household**: Assessment of current household economic conditions, 1 to 5.
   - **Blair**: Assessment of the Labour 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.

   Lets see data, we need to remove first 'Unnamed: 0' column



	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

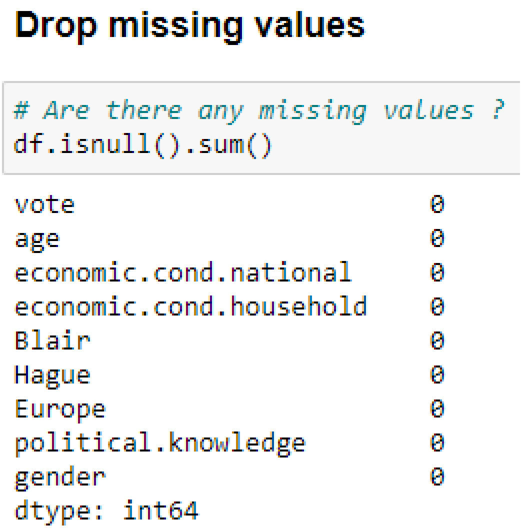
   Dataset contains total 1525 records and 9 attributes.



```
df.shape
```

```
(1525, 9)
```

Lets drop missing values if any

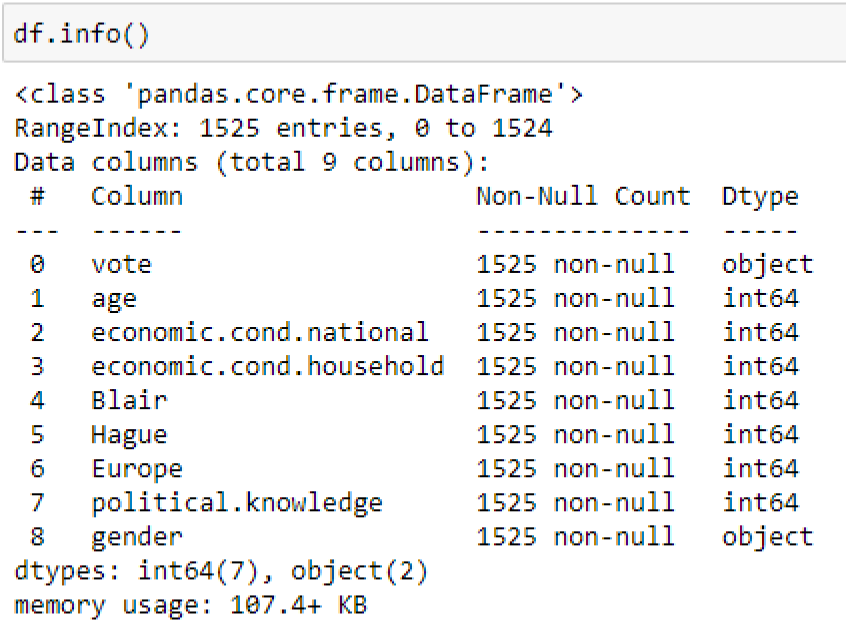


Drop missing values

```
# Are there any missing values ?
df.isnull().sum()
```

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

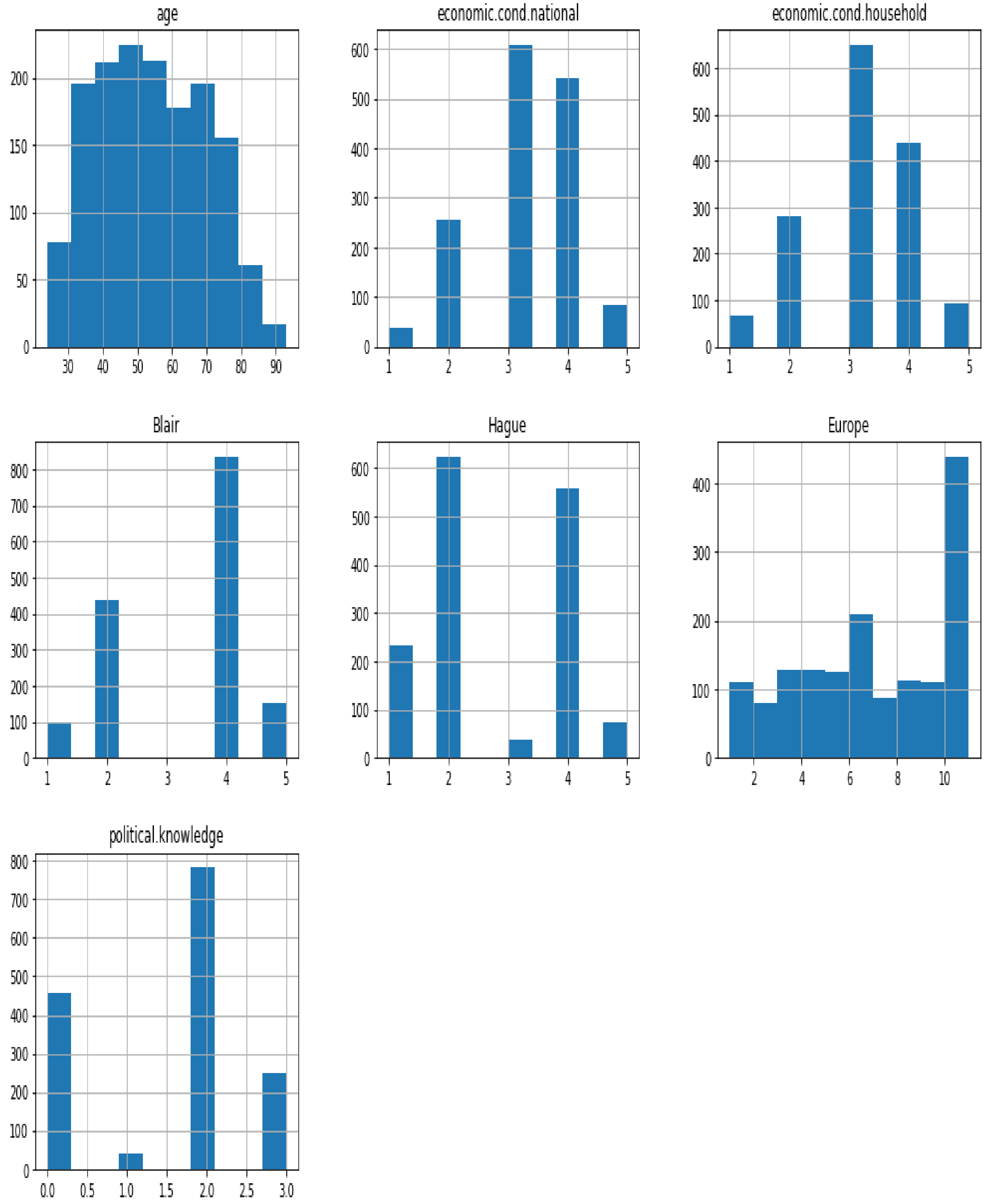
Lets see datatypes and memory usage



```
df.info()
```

```
<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
```

Lets see histogram, as per above analysis most distribution seem having more than one pick and data spread is not normal

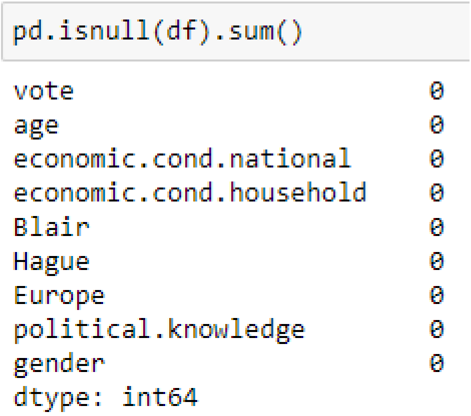


Inference

- Dataset contains 1525 records with 9 attributes
- Dataset holds two categorical and seven numerical attributes
- For attribute vote there are two different types of values
    - Conservative
    - Labour
- For attribute gender there are two different types of values
    - Male
    - Female
- Dataset contains zero missing values
- As per histogram analysis most distribution seem having more than one pick and data spread is not normal

2. Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers. (7 Marks)

Dataset contains no null value records

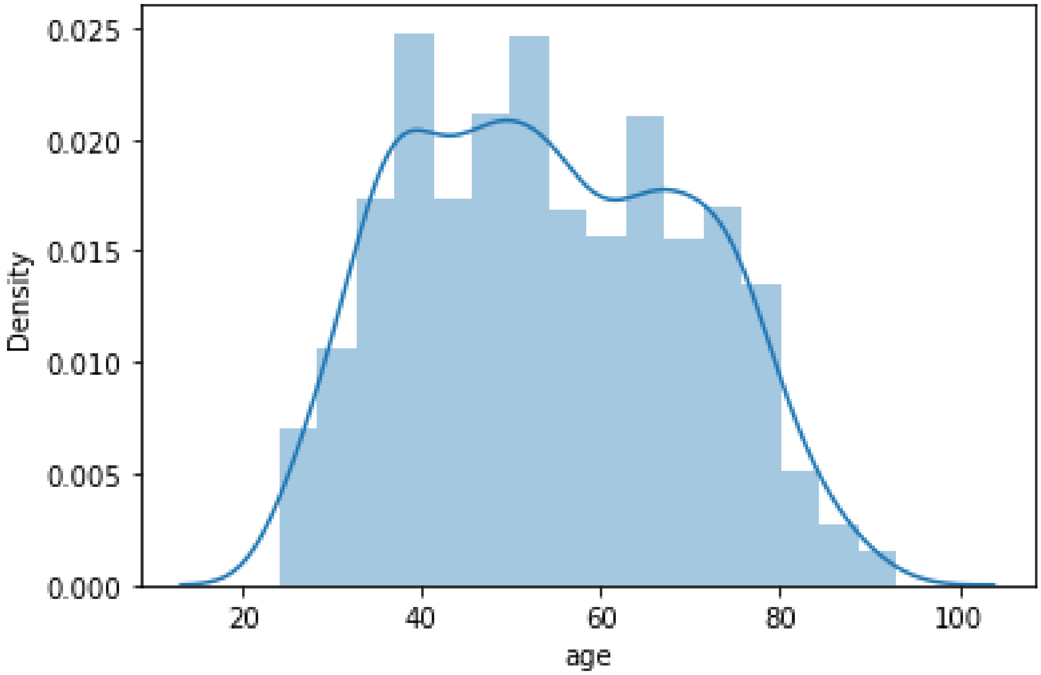


```
pd.isnull(df).sum()
```

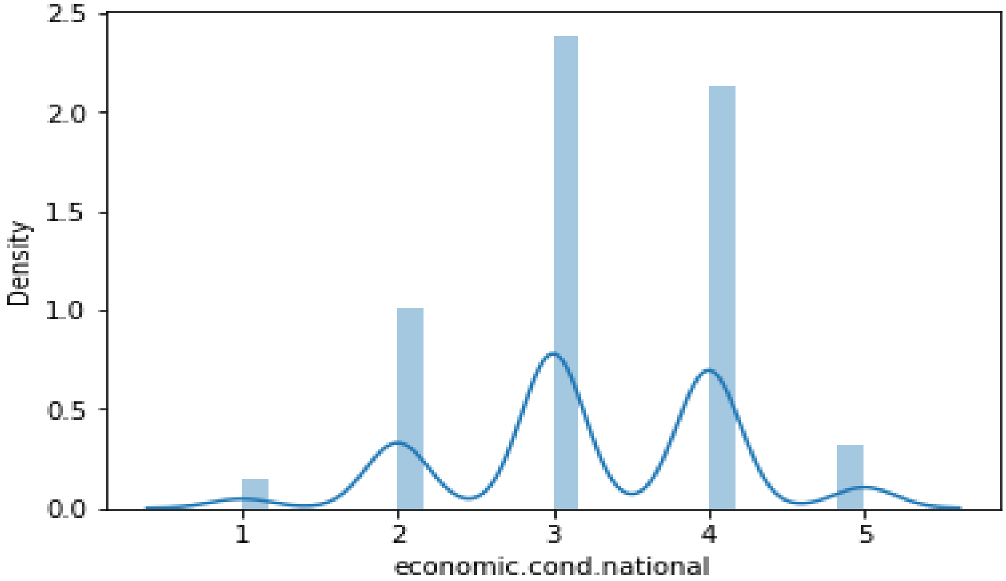
```
vote                       0
age                        0
economic.cond.national     0
economic.cond.household    0
Blair                      0
Hague                      0
Europe                     0
political.knowledge        0
gender                     0
dtype: int64
```

Lets start univariate analysis

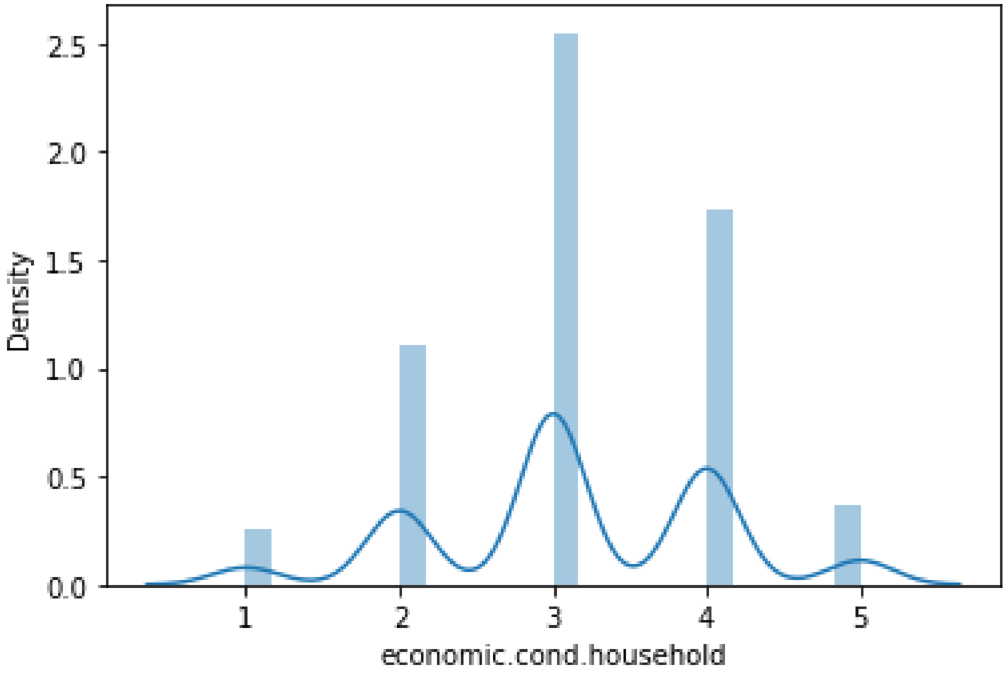
Density VS age: Distribution seem having more than one pick and trying to follow normal distribution



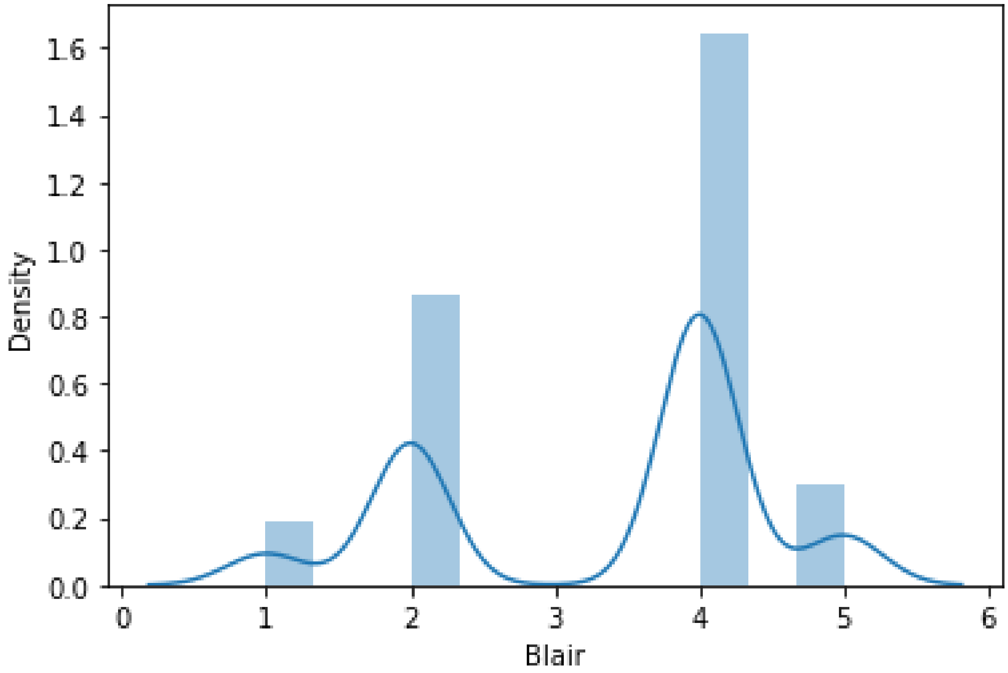
Density VS economic.cond.national: Distribution seem having more than one pick and data spread is not normal



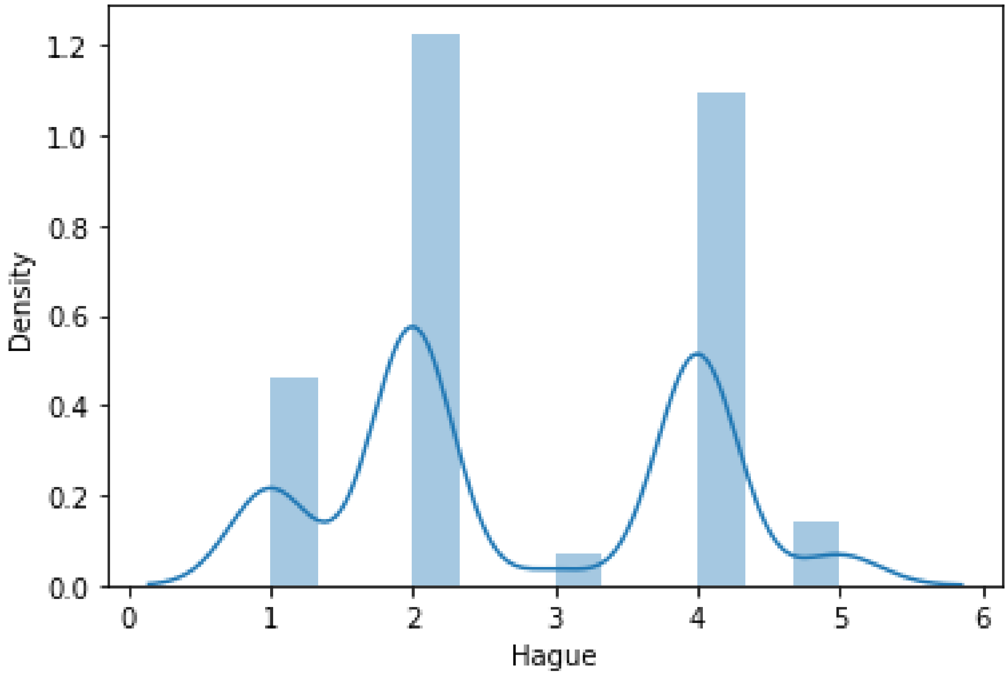
Density VS economic.cond.household: Distribution seem having more than one pick and data spread is not normal



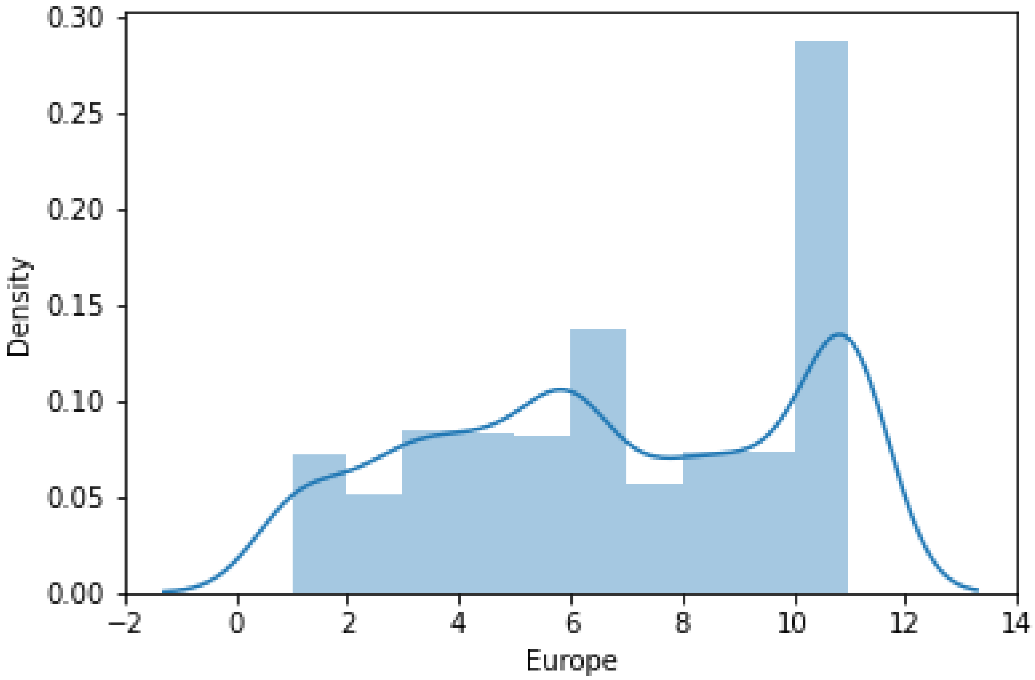
Density VS Blair: Distribution seem having more than one pick and data spread is not normal



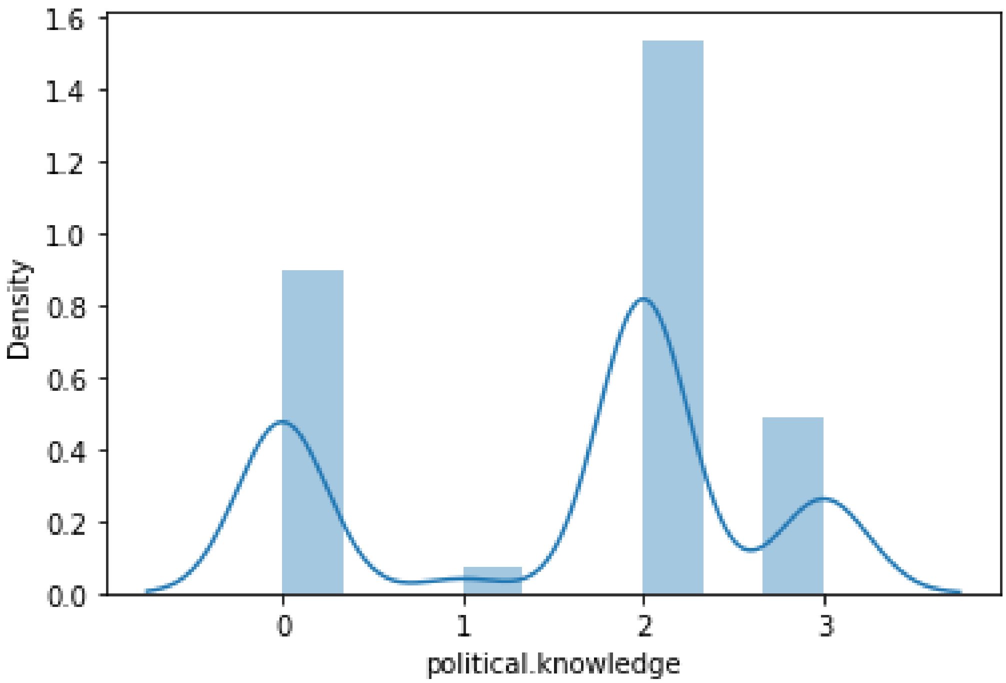
Density VS Hague: Distribution seem having more than one pick and data spread is not normal



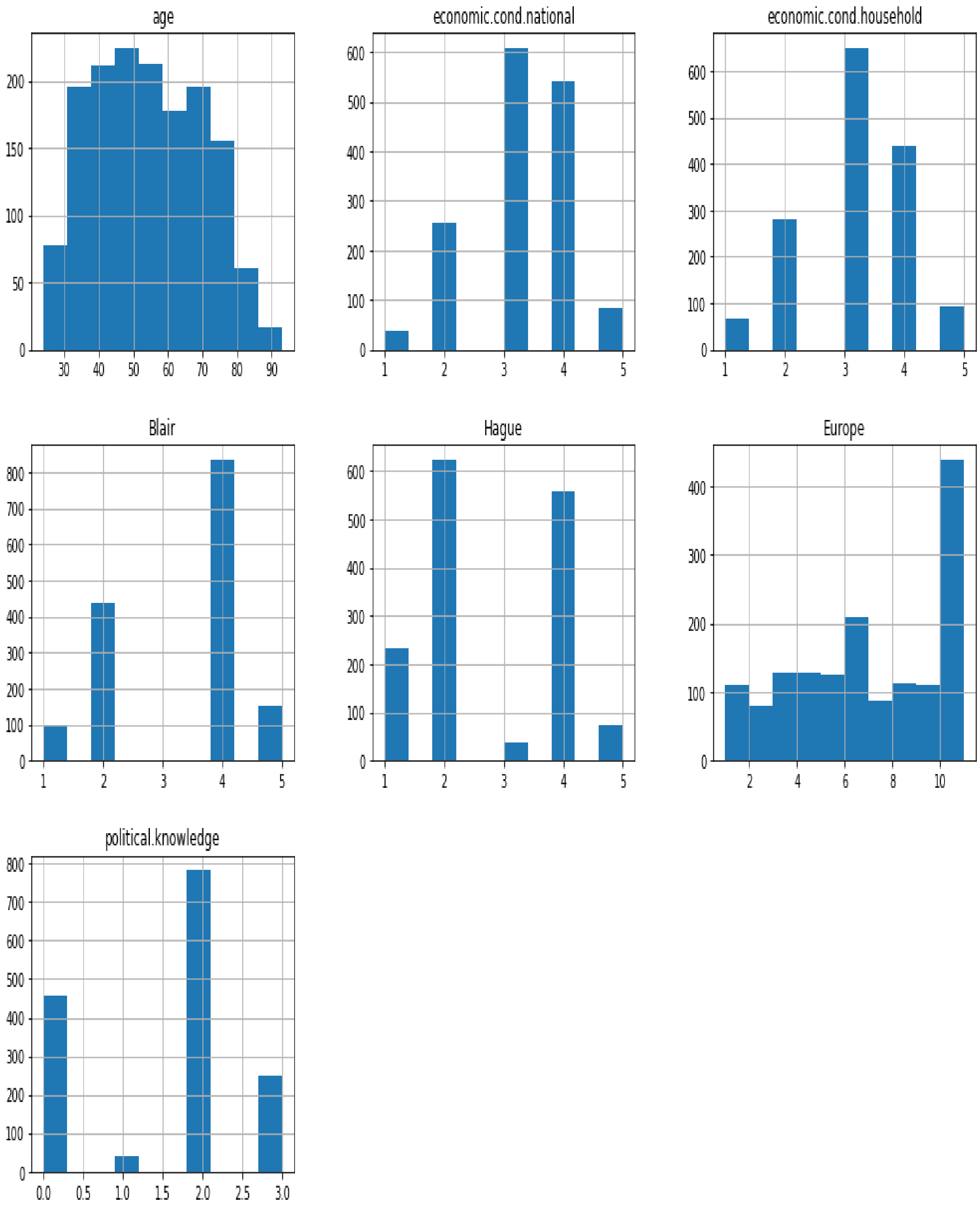
Density VS Europe: Distribution seem having more than one pick and data spread is not normal



Density VS political.knowledge: Distribution seem having more than one pick and data spread is not normal

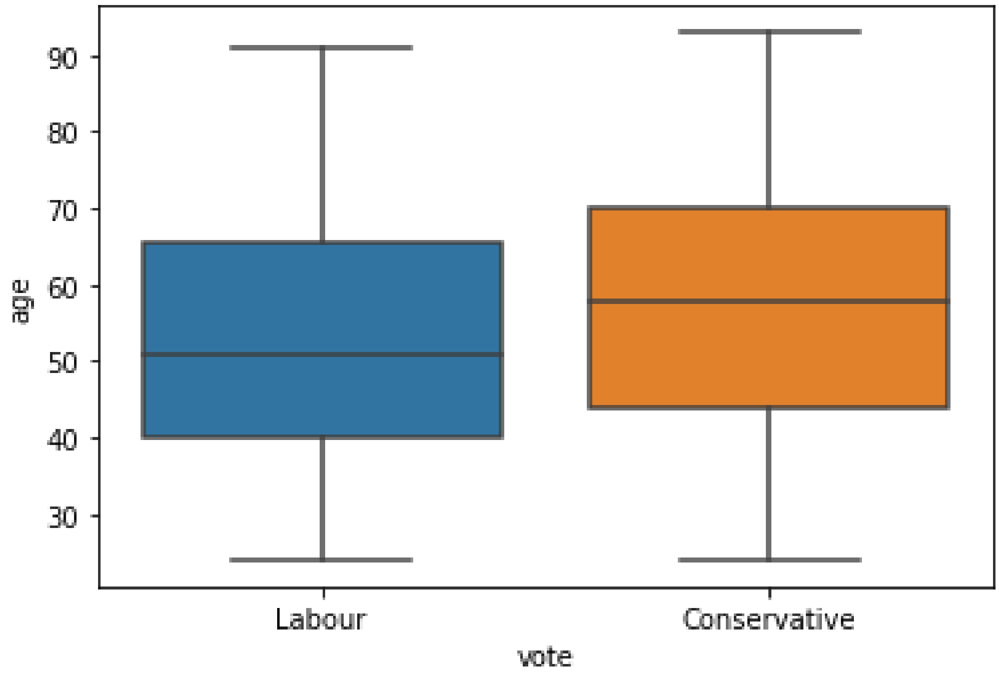


Lets see histogram, as per above analysis most distribution seem having more than one pick and data spread is not normal

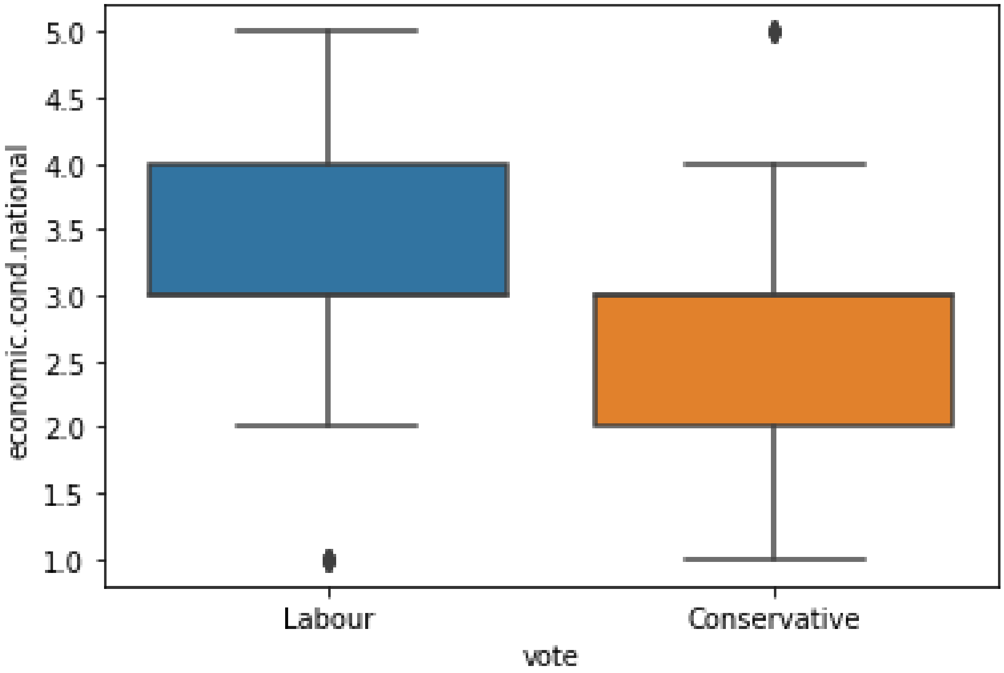


Bivariate Analysis

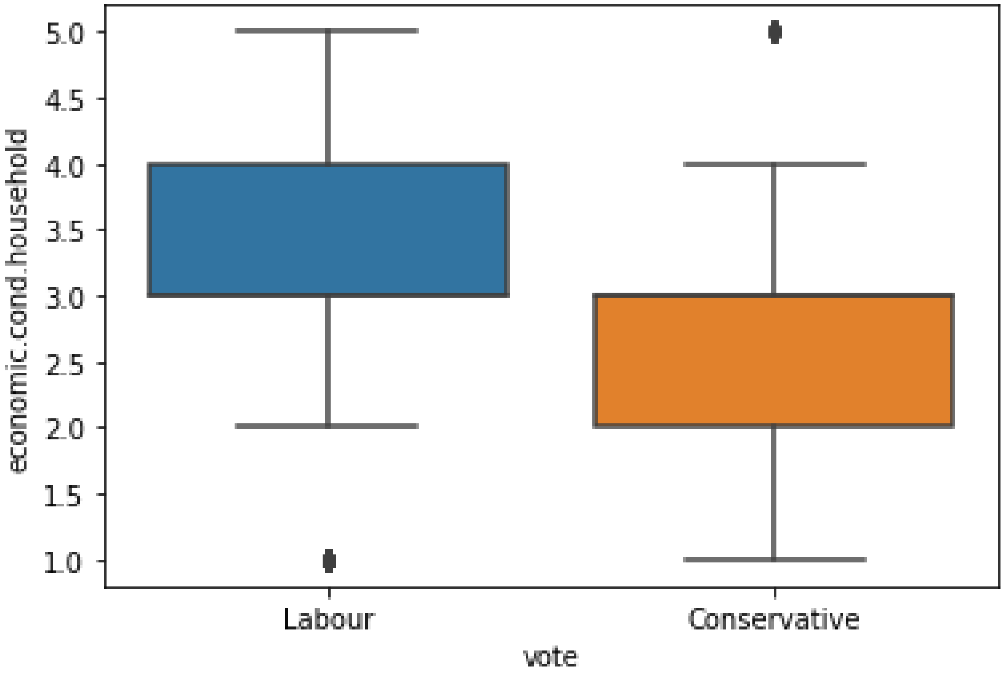
Vote VS age: new generation likes labour party more than conservative



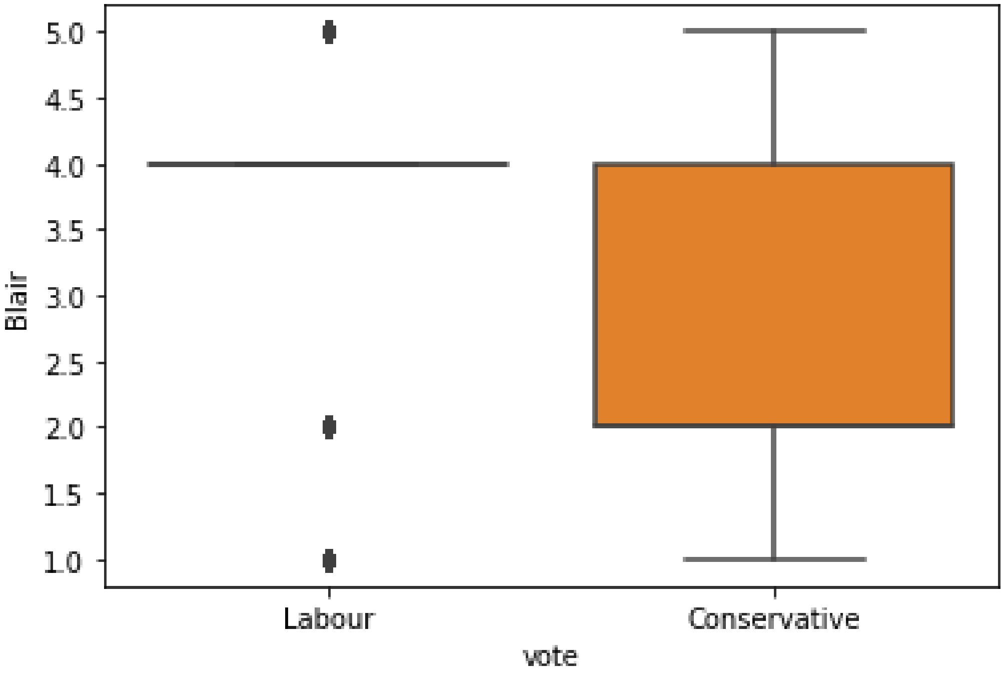
Vote VS Economic condition national: labour party hold good amount of high economic national condition followers compare to conservative party



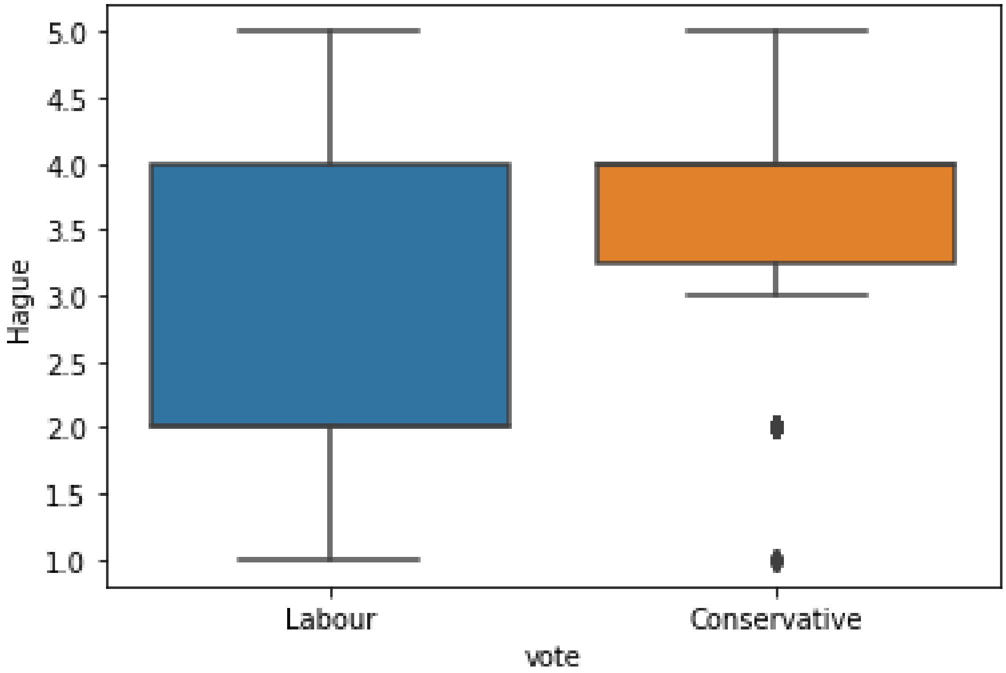
Vote VS Economic condition household: labour party hold good amount of high economic household condition followers compare to conservative party



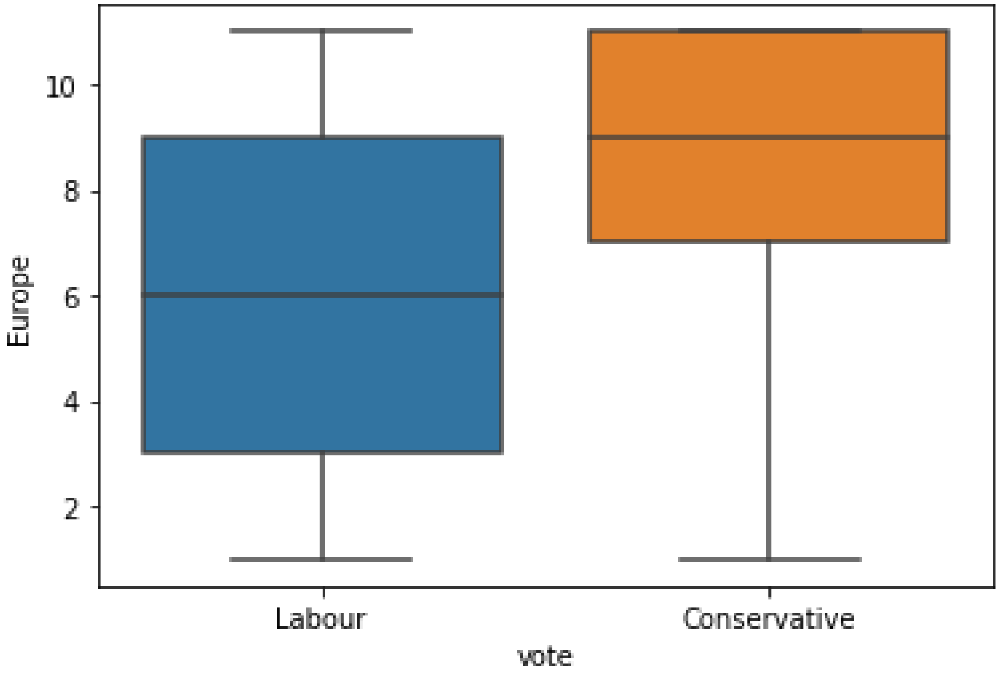
Vote VS Blair: Followers of labour party highlight blair at 4 but conservative party between 2 to 4



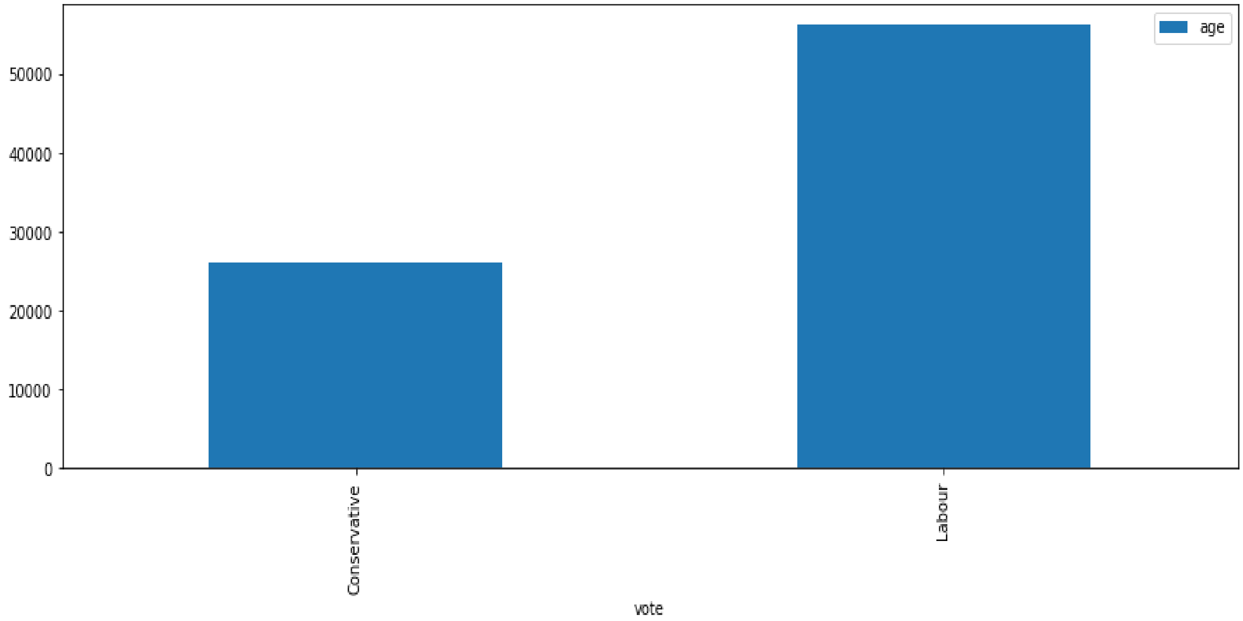
Vote VS Hague: Followers of labour party highlight hague between 2 to 4 but conservative party between 3.5 to 4



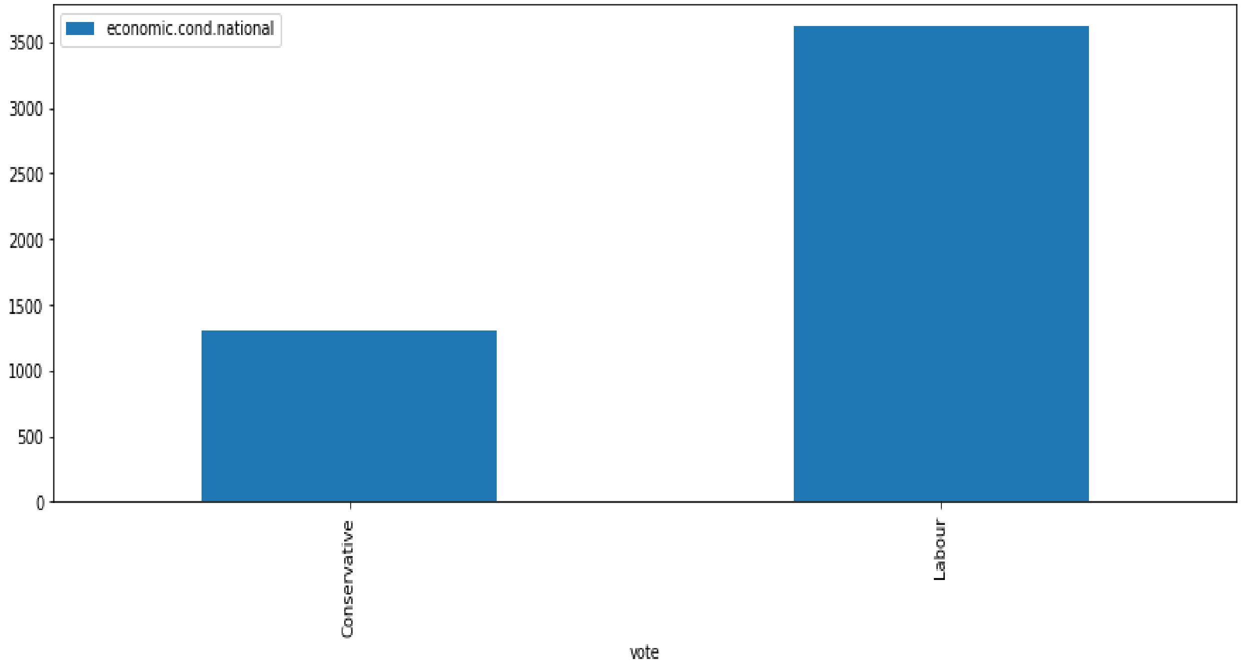
Vote VS Europe: Followers of labour party highlight high variation ‘Europe integration’ but conservative party hold high interest compare to labour



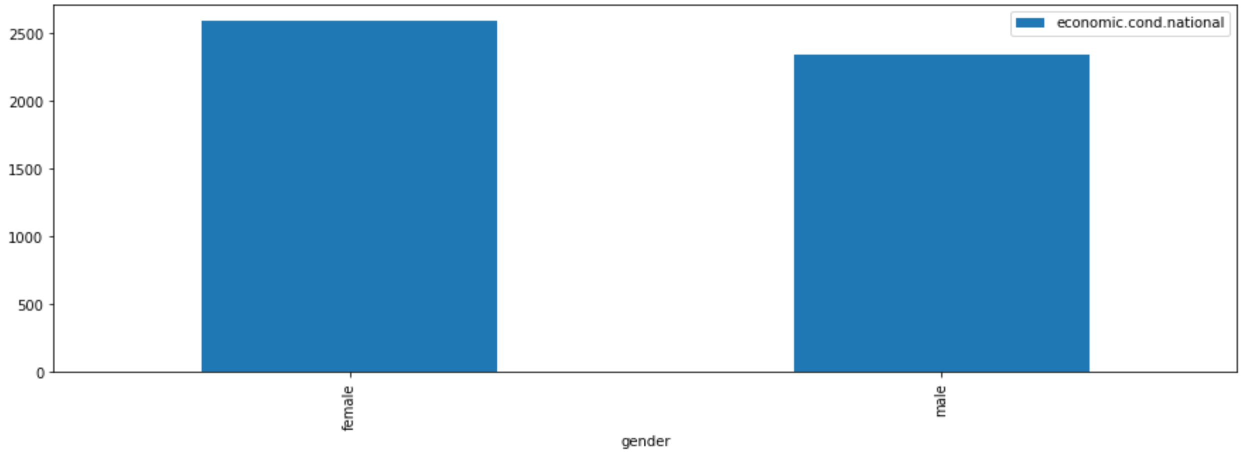
Age VS Vote: Labour party hold high age followers compare to conservative party



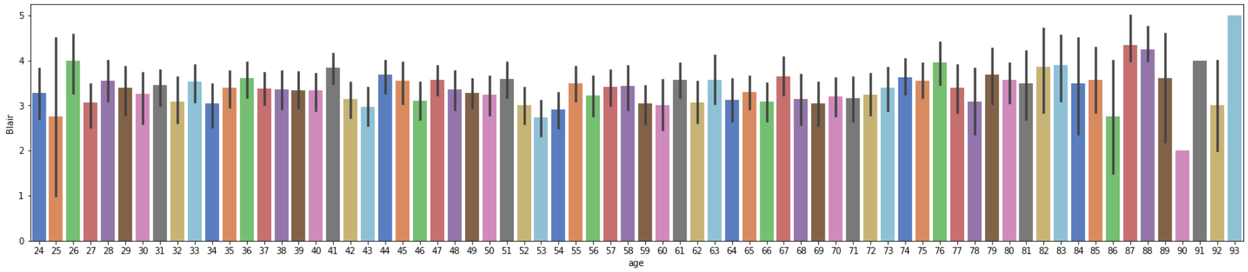
Economical condition national VS vote: Followers of labour party seem holding high economical national condition compare to conservative



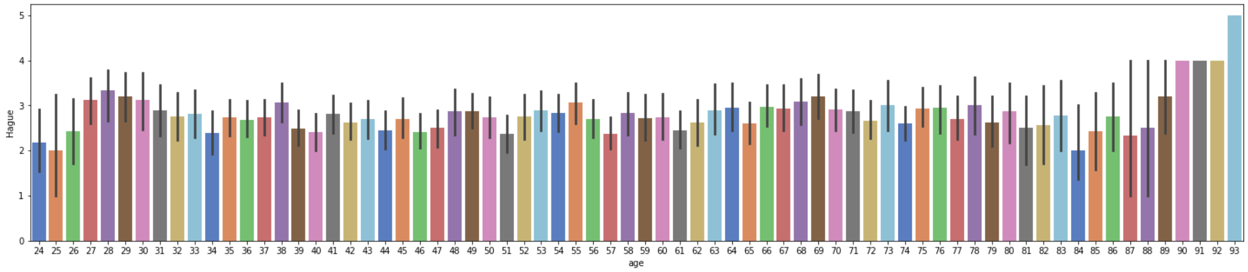
Economical condition household VS gender: Female followers seem holding high economical household condition compare to male followers



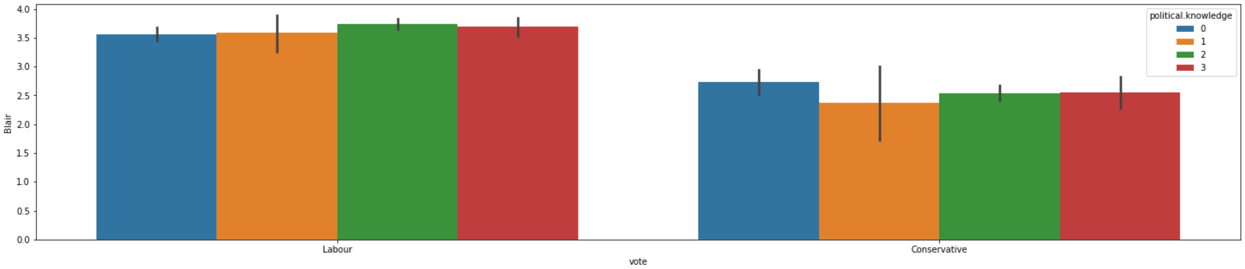
Age VS Blair: Blair seem more popular compare to Hauge in all ages



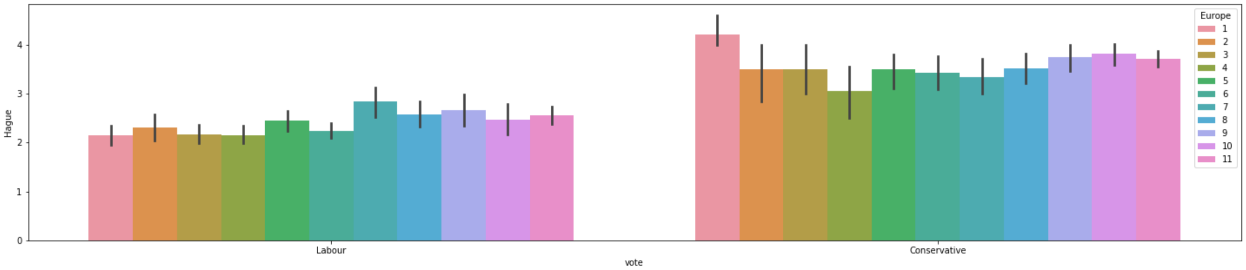
Age VS Blair: Hauge seem less popular compare to Blair in all ages



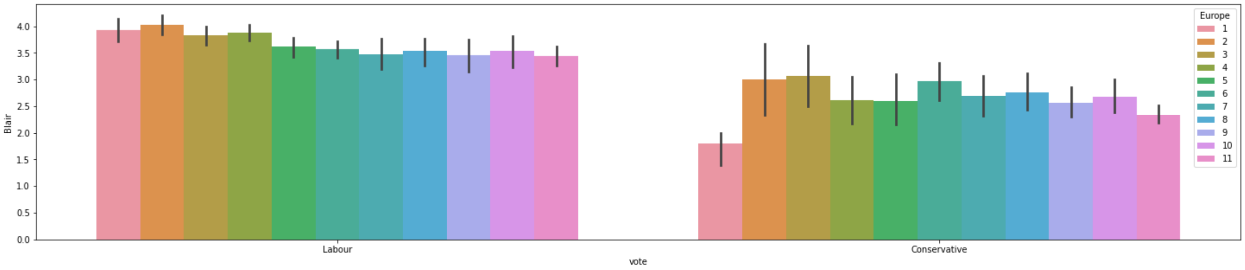
Vote VS Political knowledge: labour party voters seem having high knowledge compare to conservative party



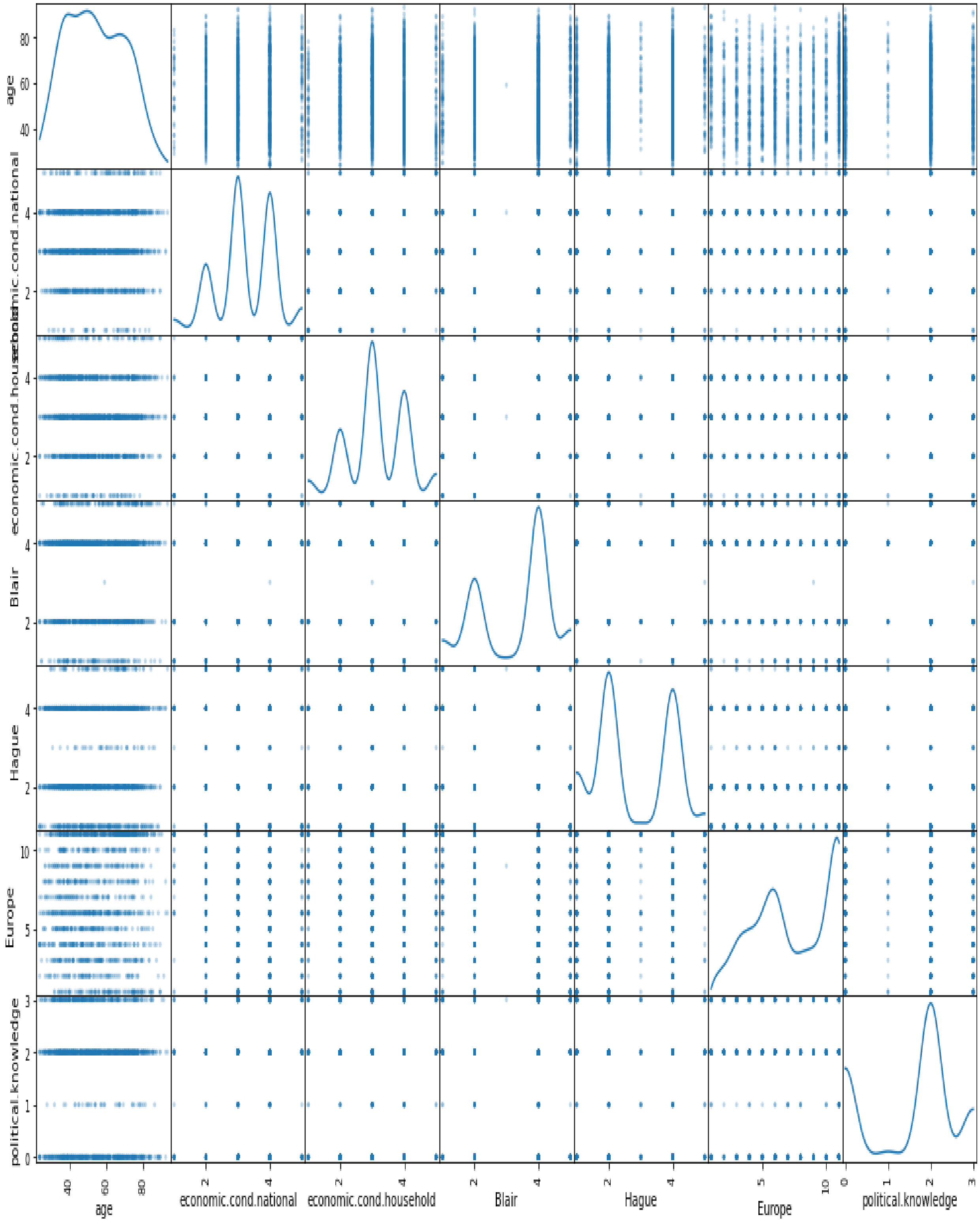
Vote VS Hauge (Hue: Europe integration) It seem conservative party followers seem to be hauge followers interested more in ‘europe integration’ compare to labour party’s hauge followers



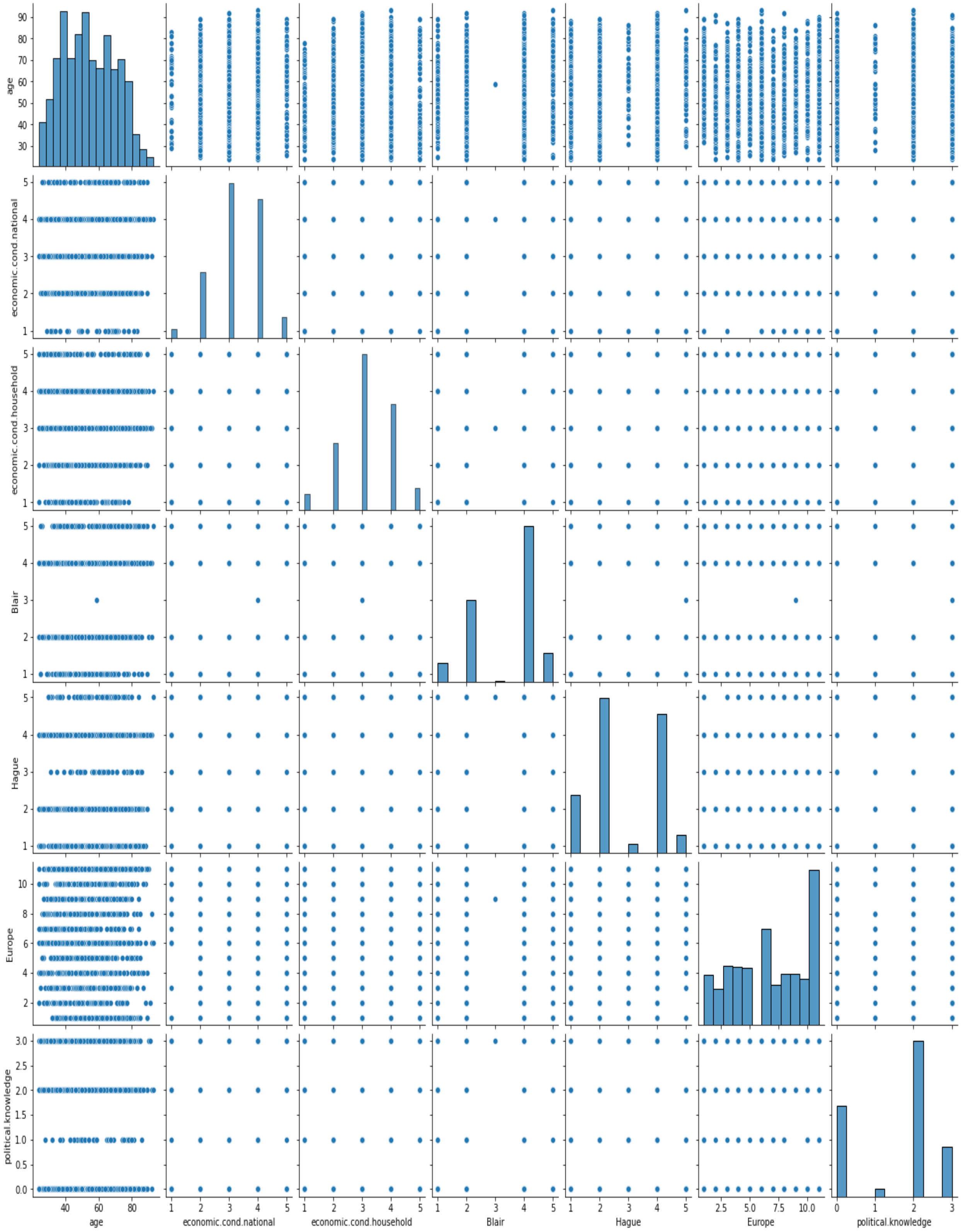
Vote VS Blair (Hue: Europe integration) It seem labour party followers seem to be blair followers interested more in ‘europe integration’ compare to conservative party’s blair followers



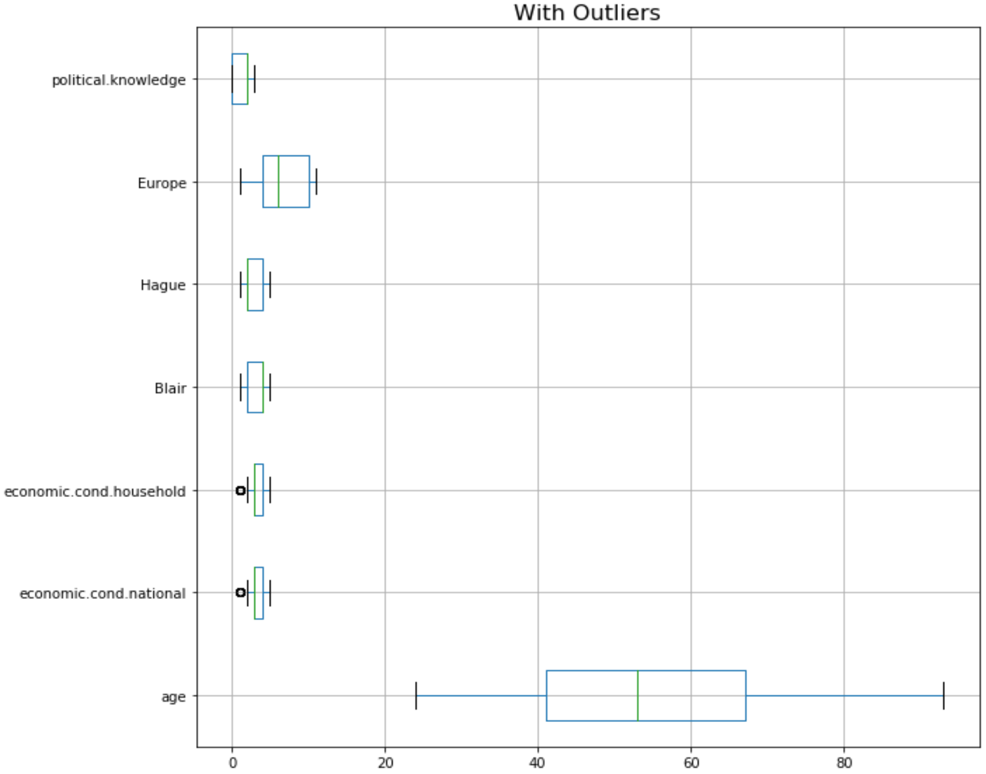
Lets see scatter plot

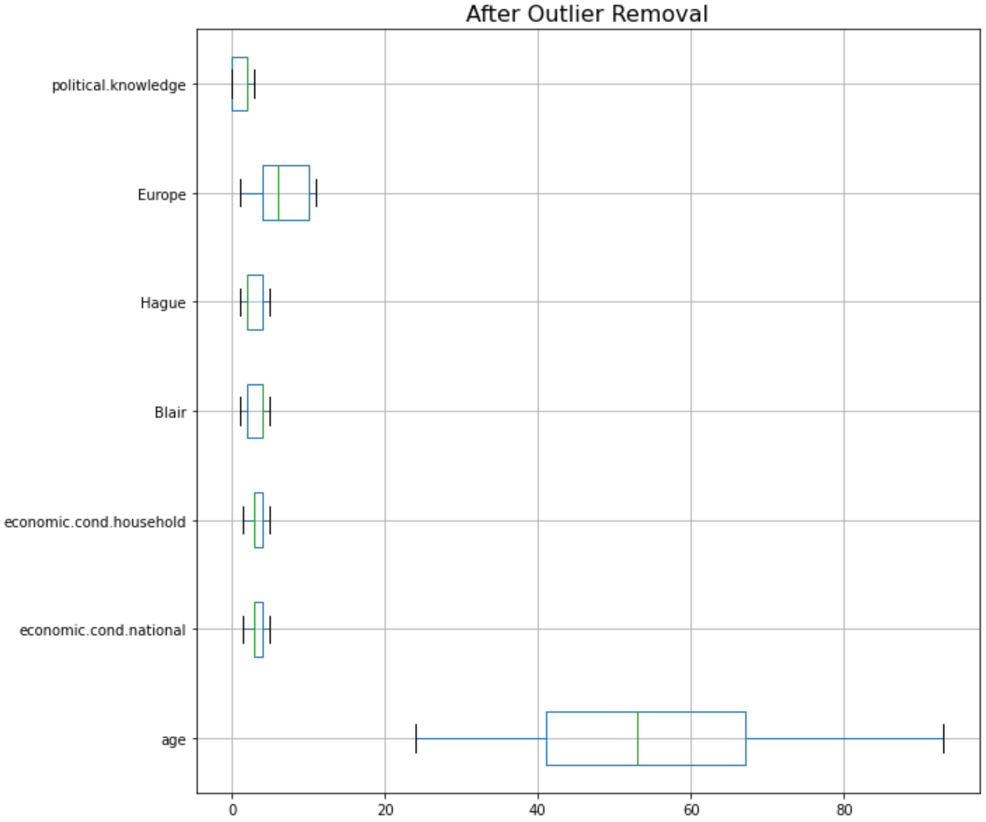


Lets see pair plot

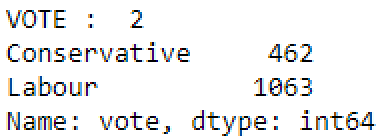


Lets see outliers, before treatment

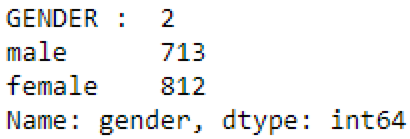




Lets see unique values for categorical variables

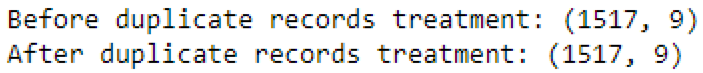


```
VOTE :  2
Conservative     462
Labour          1063
Name: vote, dtype: int64
```



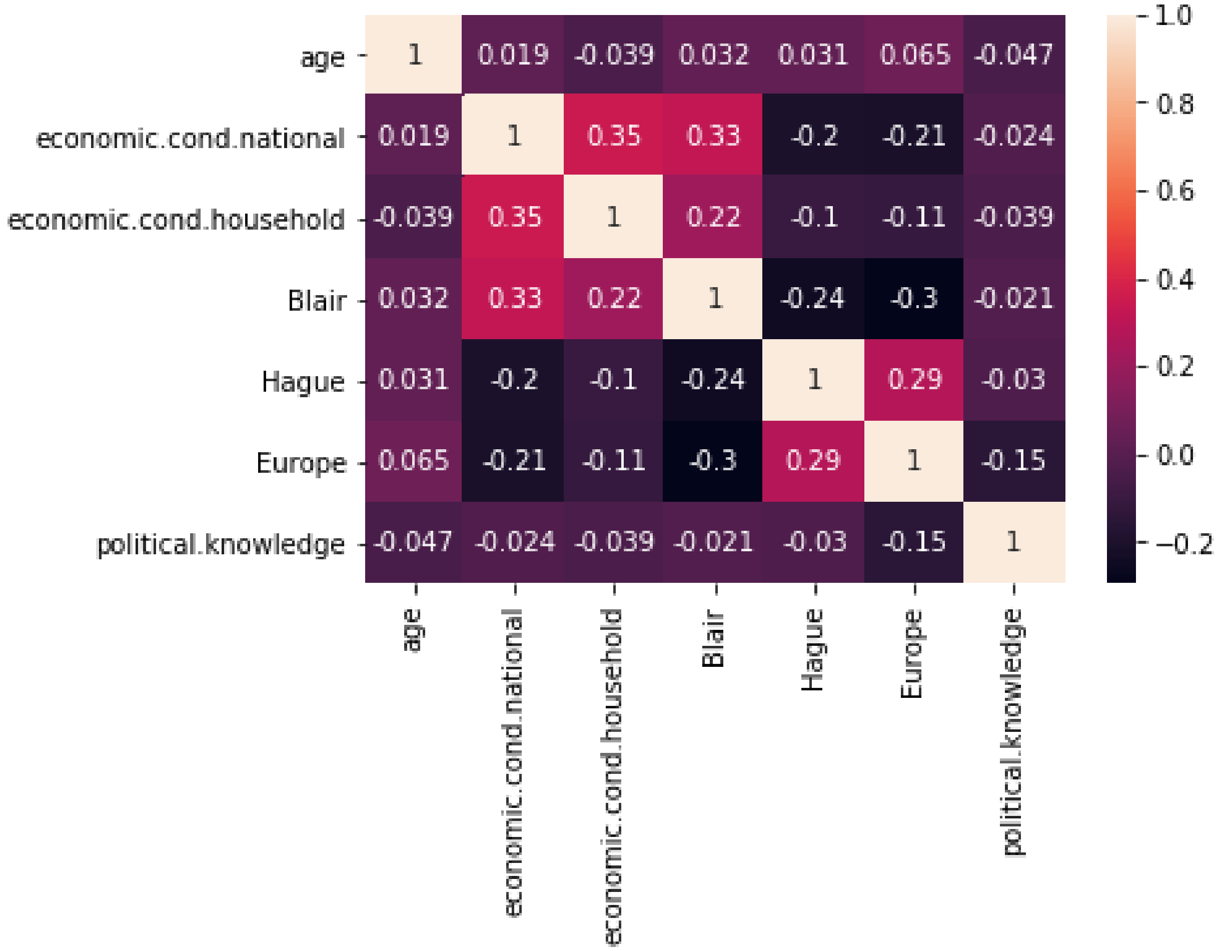
```
GENDER :  2
male      713
female    812
Name: gender, dtype: int64
```

Lets check for duplicate records. We have total 8 duplicate records

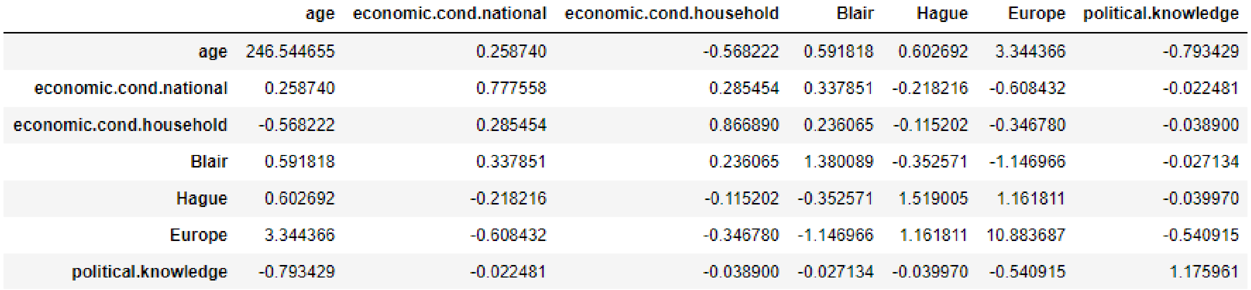


```
Before duplicate records treatment: (1517, 9)
After duplicate records treatment: (1517, 9)
```

Lets check correlation plot



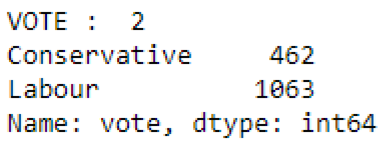
Lets see covariance plot



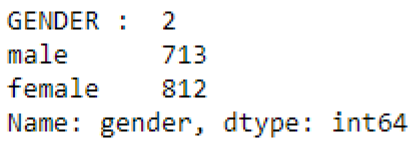
	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
age	246.544655	0.258740	-0.568222	0.591818	0.602692	3.344366	-0.793429
economic.cond.national	0.258740	0.777558	0.285454	0.337851	-0.218216	-0.608432	-0.022481
economic.cond.household	-0.568222	0.285454	0.866890	0.236065	-0.115202	-0.346780	-0.038900
Blair	0.591818	0.337851	0.236065	1.380089	-0.352571	-1.146966	-0.027134
Hague	0.602692	-0.218216	-0.115202	-0.352571	1.519005	1.161811	-0.039970
Europe	3.344366	-0.608432	-0.346780	-1.146966	1.161811	10.883687	-0.540915
political.knowledge	-0.793429	-0.022481	-0.038900	-0.027134	-0.039970	-0.540915	1.175961

3. **Encode the data (having string values) for Modelling. Is Scaling necessary here or not?(3 pts), Data Split: Split the data into train and test (70:30) (2 pts).**

We can see dataset attribute vote & gender contains variations which need to be converted from categorical to numerical



```
VOTE :  2
Conservative     462
Labour          1063
Name: vote, dtype: int64
```

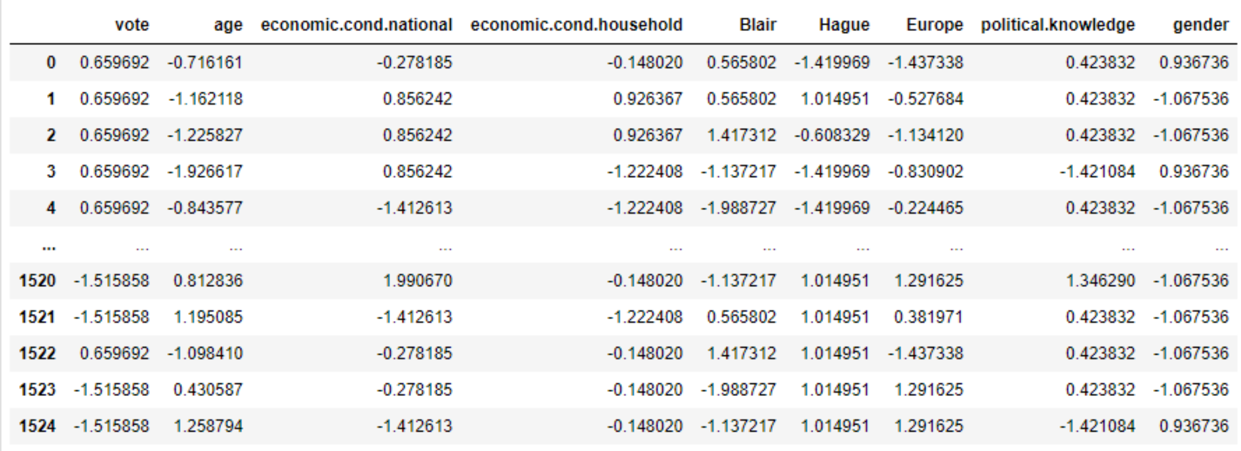


```
GENDER :  2
male      713
female    812
Name: gender, dtype: int64
```

Vote contains two values ‘conservative’ and ‘labour’ only
Gender contains two values ‘male’ and ‘female’ only

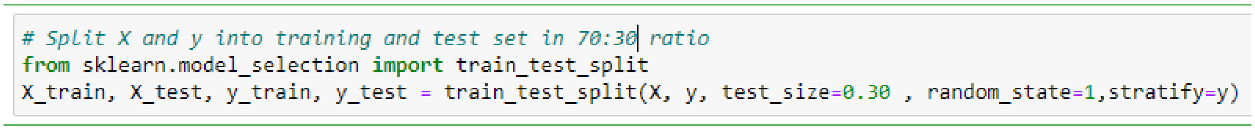
We use manual encoding technique.

We need to do scaling because there are different level available in dataset which need to standardize. We use standard-scaler



	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	0.659692	-0.716161	-0.278185	-0.148020	0.565802	-1.419969	-1.437338	0.423832	0.936736
1	0.659692	-1.162118	0.856242	0.926367	0.565802	1.014951	-0.527684	0.423832	-1.067536
2	0.659692	-1.225827	0.856242	0.926367	1.417312	-0.608329	-1.134120	0.423832	-1.067536
3	0.659692	-1.926617	0.856242	-1.222408	-1.137217	-1.419969	-0.830902	-1.421084	0.936736
4	0.659692	-0.843577	-1.412613	-1.222408	-1.988727	-1.419969	-0.224465	0.423832	-1.067536
...	...	...	...	...	...	...	...	...	...
1520	-1.515858	0.812836	1.990670	-0.148020	-1.137217	1.014951	1.291625	1.346290	-1.067536
1521	-1.515858	1.195085	-1.412613	-1.222408	0.565802	1.014951	0.381971	0.423832	-1.067536
1522	0.659692	-1.098410	-0.278185	-0.148020	1.417312	1.014951	-1.437338	0.423832	-1.067536
1523	-1.515858	0.430587	-0.278185	-0.148020	-1.988727	1.014951	1.291625	0.423832	-1.067536
1524	-1.515858	1.258794	-1.412613	-0.148020	-1.137217	1.014951	1.291625	-1.421084	0.936736

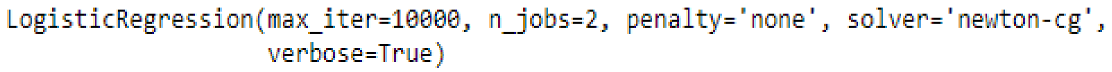
Lets Split the data into train and test (70:30)



```
# Split X and y into training and test set in 70:30 ratio
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30 , random_state=1,stratify=y)
```

4. Apply Logistic Regression and LDA (Linear Discriminant Analysis) (3 pts). Interpret the inferences of both model s (2 pts)

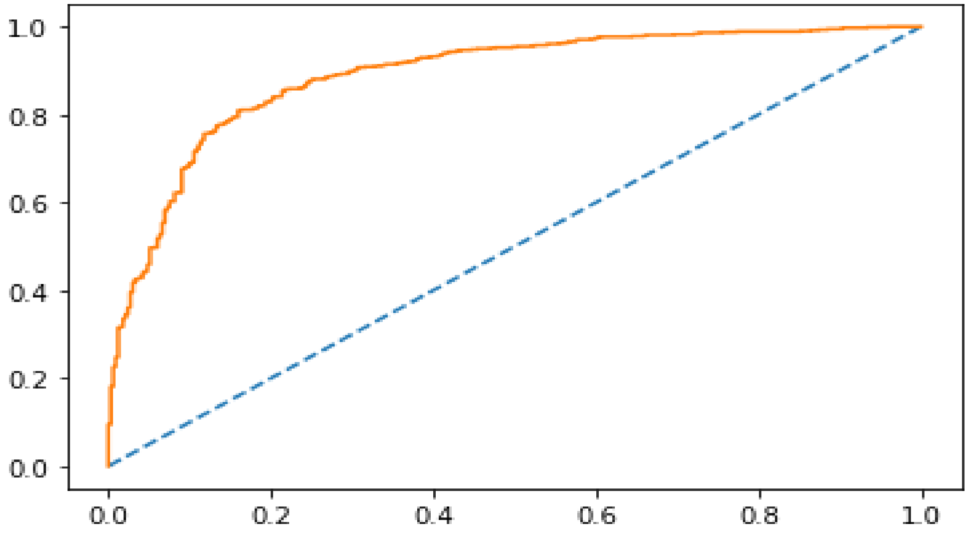
- Logistic regression model



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

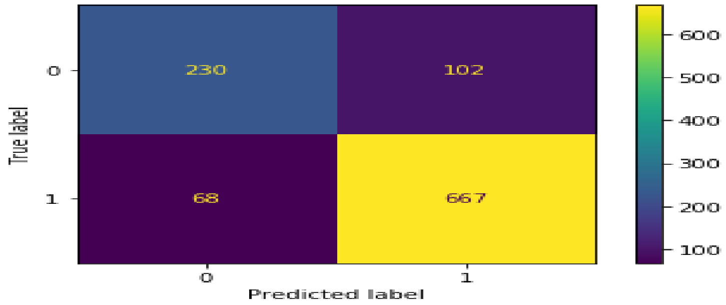
Logistic regression model accuracy (Train data): 0.8406747891283973

AUC/ROC Curve: AUC: 0.889

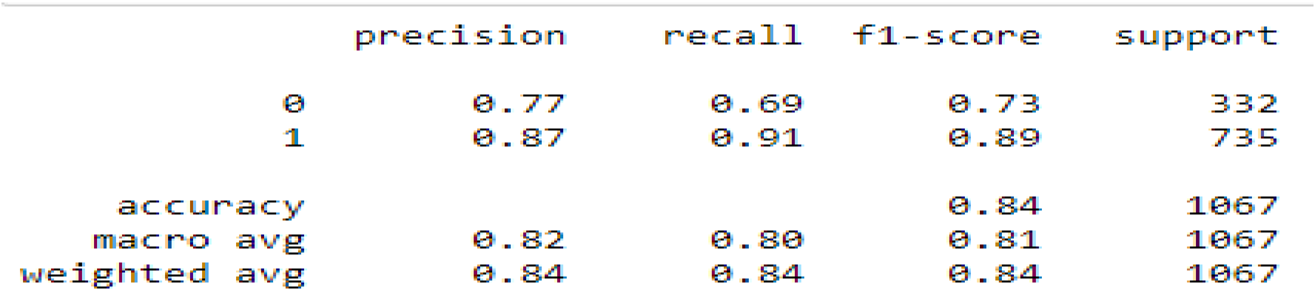


.

Confusion matrix (Train data)

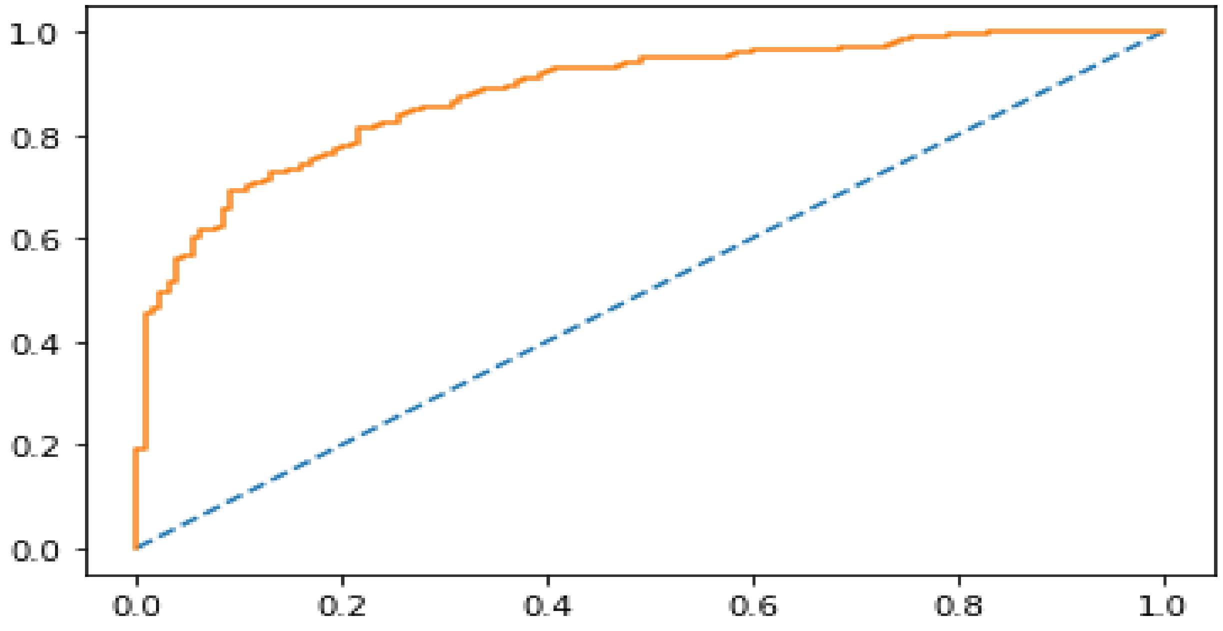


Classification report

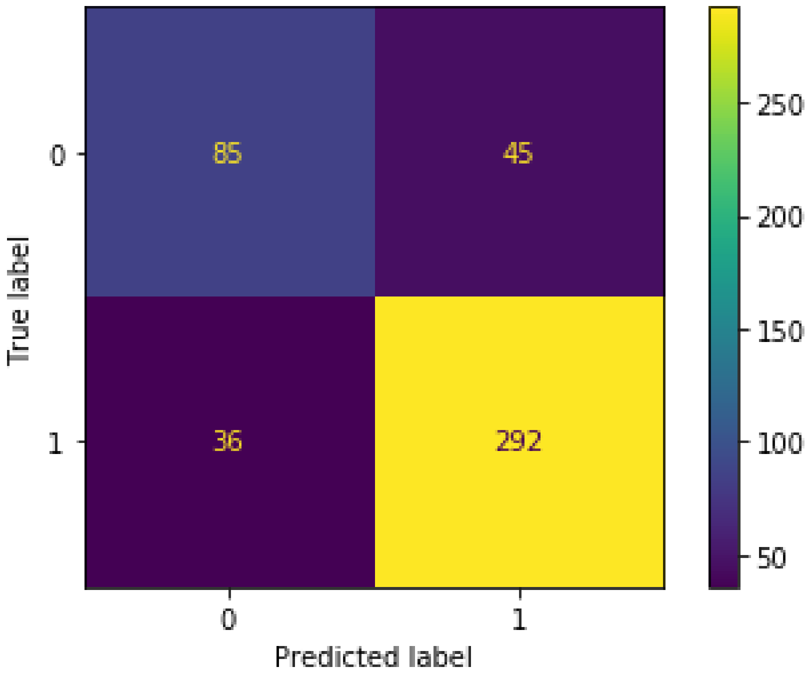


```
              precision    recall  f1-score   support

           0       0.77      0.69      0.73       332
           1       0.87      0.91      0.89       735

    accuracy                           0.84      1067
   macro avg       0.82      0.80      0.81      1067
weighted avg       0.84      0.84      0.84      1067
```

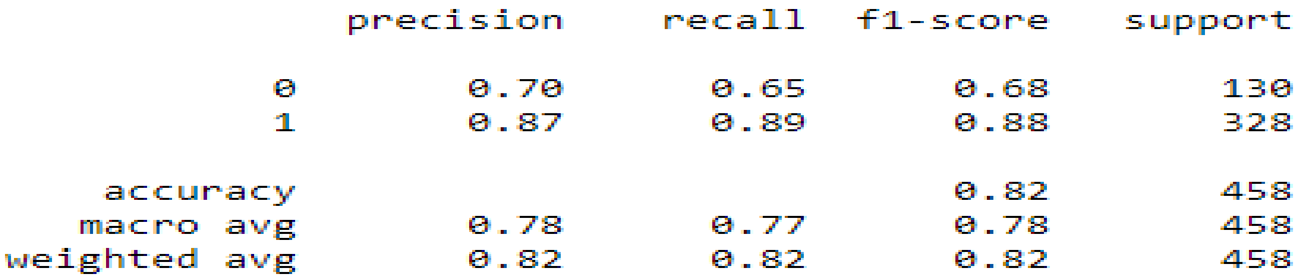
```
Logistic regression model accuracy (Test data): 0.8231441048034934

AUC/ROC Curve AUC: 0.889
```



Confusion matrix (Test Data)

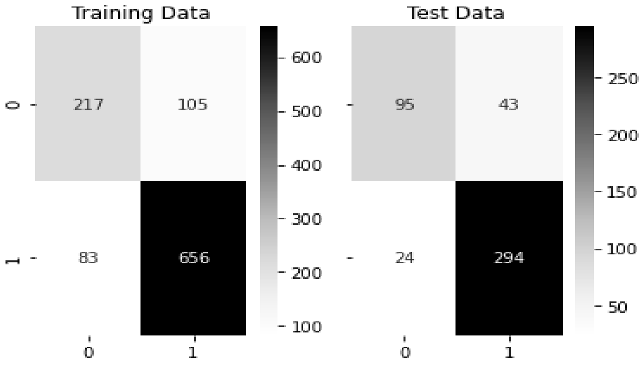


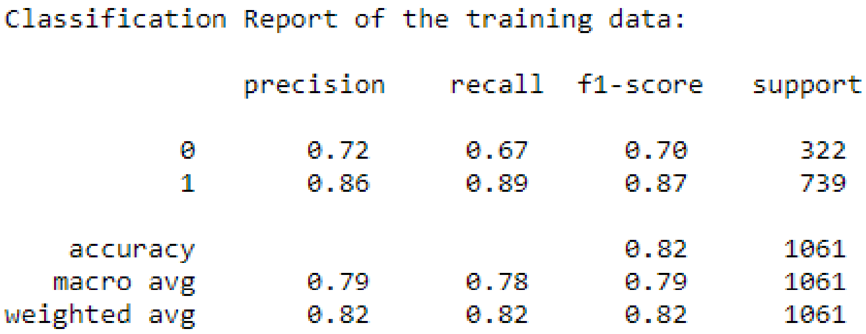
Classification report



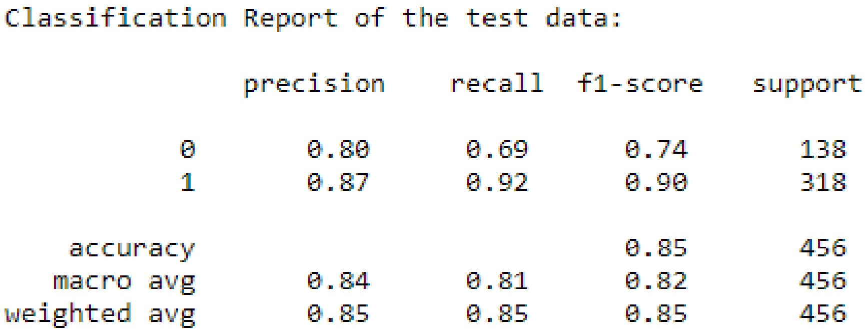
```
              precision    recall  f1-score   support

           0       0.70      0.65      0.68       130
           1       0.87      0.89      0.88       328

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

- LDA Linear Discriminant Analysis

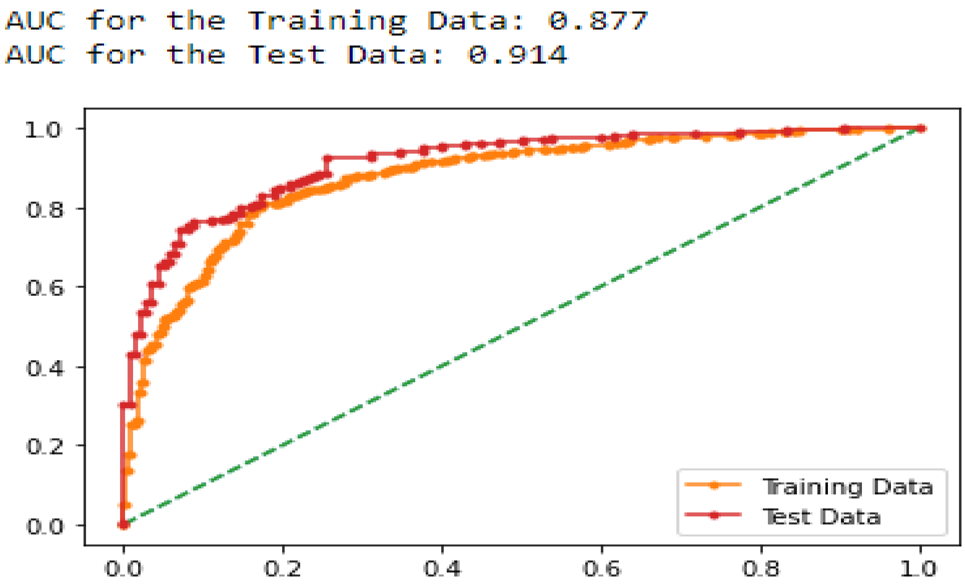




```
Classification Report of the training data:

              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
```



```
Classification Report of the test data:

              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
```

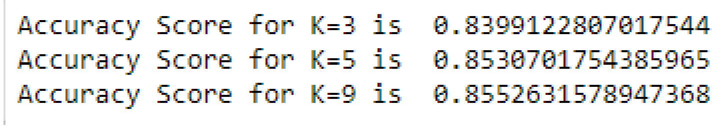


```
AUC for the Training Data: 0.877
AUC for the Test Data: 0.914
```

- Inferences
  - Logistic regression seem to be working good on both train data 84% and test data 82% accuracy
  - Logistic regression seem to be gaining accuracy but losing recall while moving from train data to test data
  - LDA seem to be working good on both train data 87% and test data 91% accuracy
  - LDA seem to be gaining accuracy but losing recall while moving from train data to test data

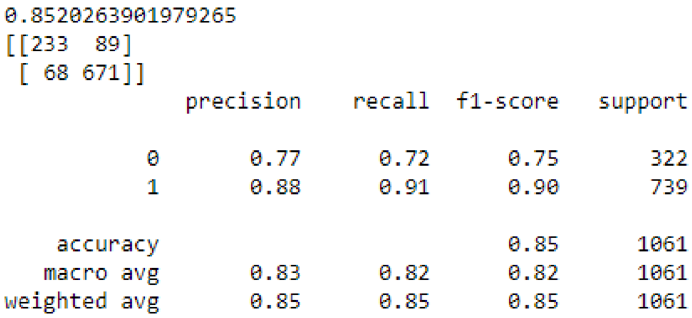
5. **Apply KNN Model and Naïve Bayes Model(5 pts). Interpret the inferences of each model (2 pts)**

- **KNN**



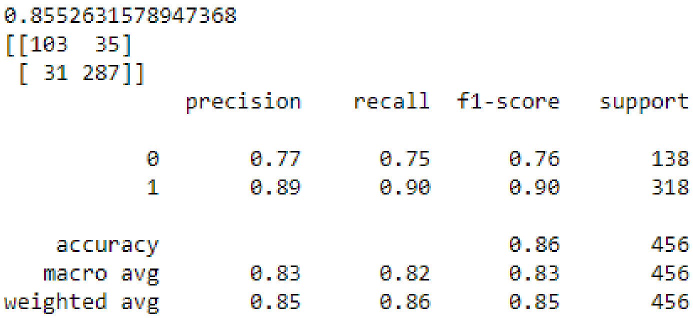
```
Accuracy Score for K=3 is  0.8399122807017544
Accuracy Score for K=5 is  0.8530701754385965
Accuracy Score for K=9 is  0.8552631578947368
```

Model score train data



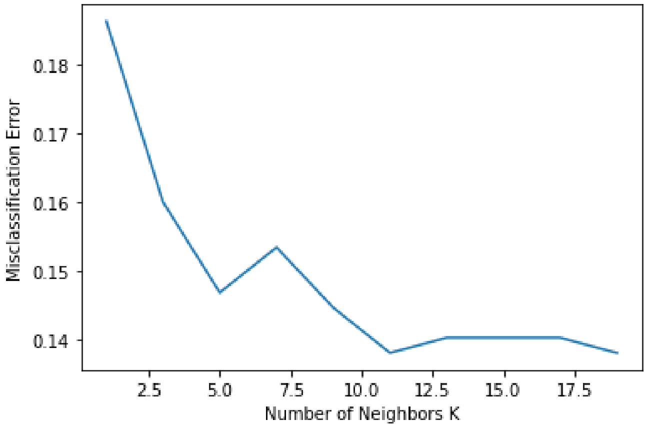
```
0.8520263901979265
[[233  89]
 [ 68 671]]
              precision    recall  f1-score   support

           0       0.77      0.72      0.75       322
           1       0.88      0.91      0.90       739

    accuracy                           0.85      1061
   macro avg       0.83      0.82      0.82      1061
weighted avg       0.85      0.85      0.85      1061
```

Model score test data



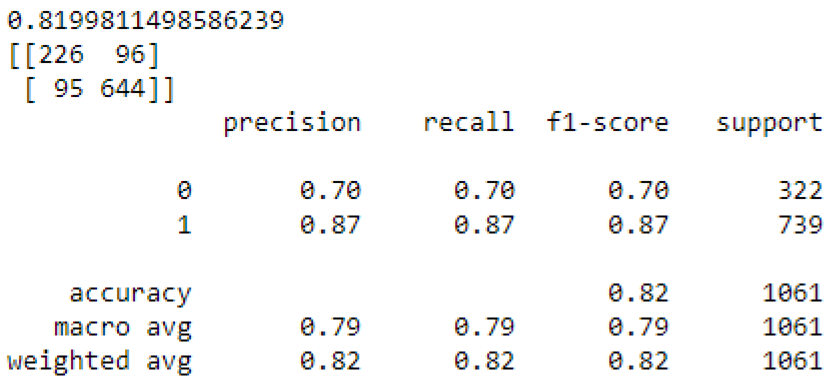
```
0.8552631578947368
[[103  35]
 [ 31 287]]
              precision    recall  f1-score   support

           0       0.77      0.75      0.76       138
           1       0.89      0.90      0.90       318

    accuracy                           0.86       456
   macro avg       0.83      0.82      0.83       456
weighted avg       0.85      0.86      0.85       456
```

Inference: Knn seem be working good for k=5 with least misclassification error



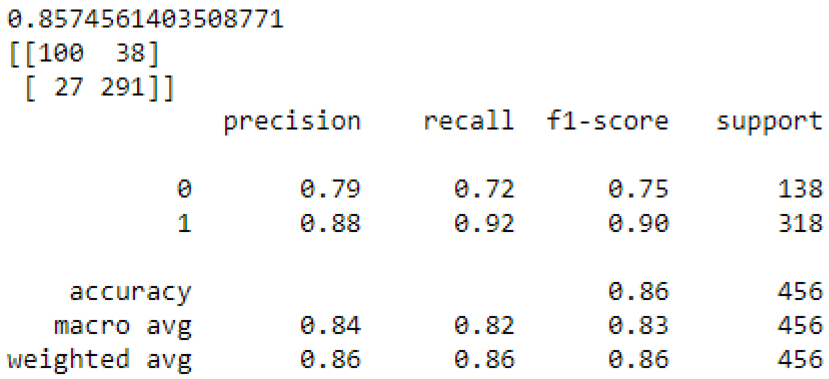
- Naïve bayes

Performance matrix on train data



```
0.8199811498586239
[[226  96]
 [ 95 644]]
              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
```

Performance matrix on test data

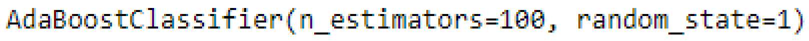


```
0.8574561403508771
[[100  38]
 [ 27 291]]
              precision    recall  f1-score   support

           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
```

- Inferences
    - knn seem to be working good for k=5 at 85% compare to k=3 at 83%, k=9 at 85%
    - knn seem to be working good on both train data 83%(k=3) and test data 82%(k=3) accuracy
    - knn show high precision and high recall while k=3 at 83%
    - knn seem be working good for k=5 with least misclassification error
    - Naïve Bayes seem to be working good on both train data 81% and test data 82%
    - Naïve Bayes show high precision and high recall while train and test

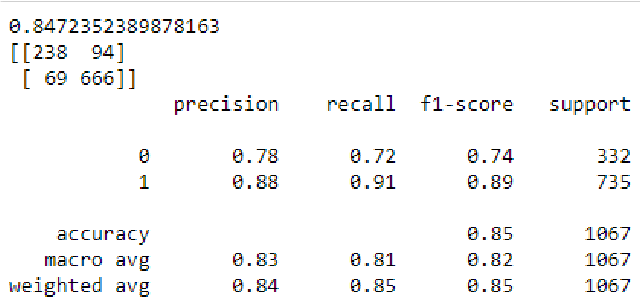
6. **Model Tuning (2 pts) , Bagging (2.5 pts) and Boosting (2.5 pts).**

- **Adaptive boosting**



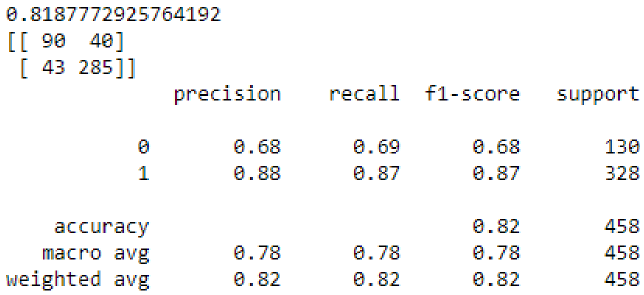
```
AdaBoostClassifier(n_estimators=100, random_state=1)
```

Performance matrix on train data



```
0.8472352389878163
[[238  94]
 [ 69 666]]
              precision    recall  f1-score   support

           0       0.78      0.72      0.74       332
           1       0.88      0.91      0.89       735

    accuracy                           0.85      1067
   macro avg       0.83      0.81      0.82      1067
weighted avg       0.84      0.85      0.85      1067
```

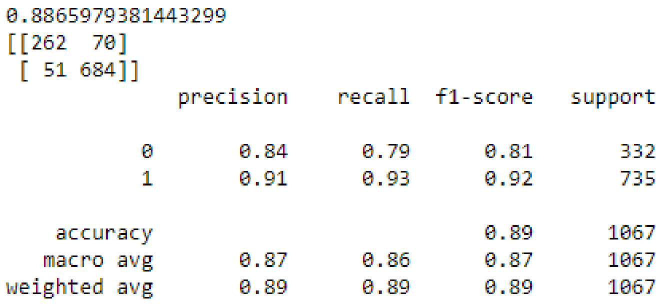
Performance matrix on test data



```
0.8187772925764192
[[ 90  40]
 [ 43 285]]
              precision    recall  f1-score   support

           0       0.68      0.69      0.68       130
           1       0.88      0.87      0.87       328

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

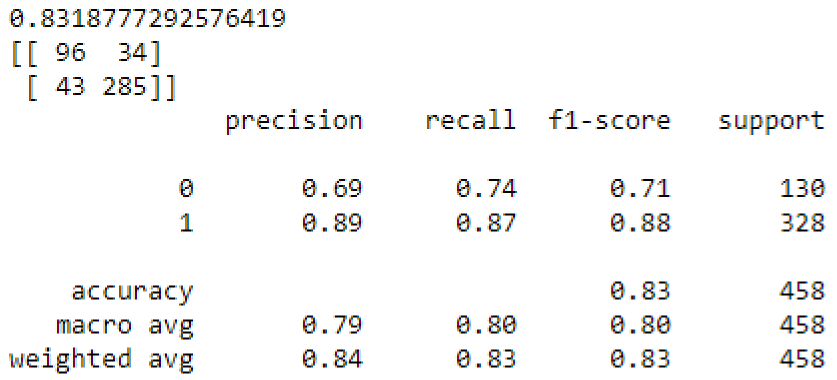
- Gradient boosting

Performance matrix on train data



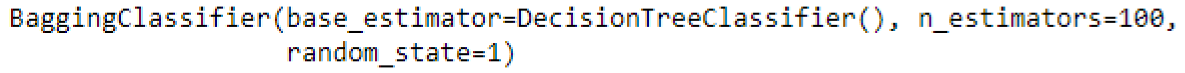
```
0.8865979381443299
[[262  70]
 [ 51 684]]
              precision    recall  f1-score   support

           0       0.84      0.79      0.81       332
           1       0.91      0.93      0.92       735

    accuracy                           0.89      1067
   macro avg       0.87      0.86      0.87      1067
weighted avg       0.89      0.89      0.89      1067
```

Performance matrix on test data



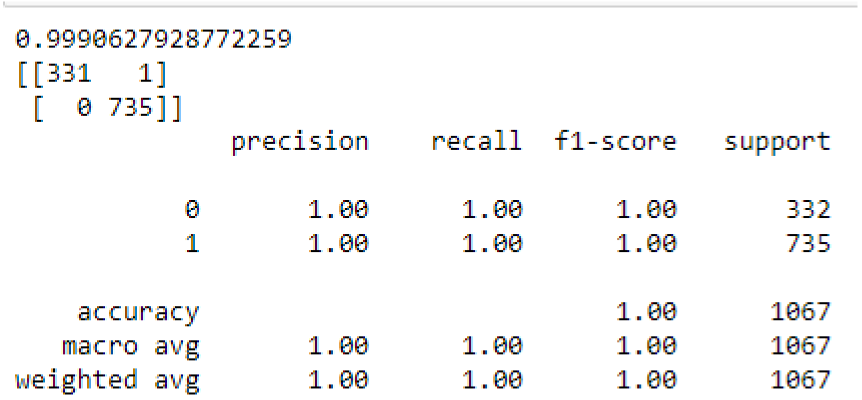
```
0.8318777292576419
[[ 96  34]
 [ 43 285]]
              precision    recall  f1-score   support

           0       0.69      0.74      0.71       130
           1       0.89      0.87      0.88       328

    accuracy                           0.83       458
   macro avg       0.79      0.80      0.80       458
weighted avg       0.84      0.83      0.83       458
```

- Bagging



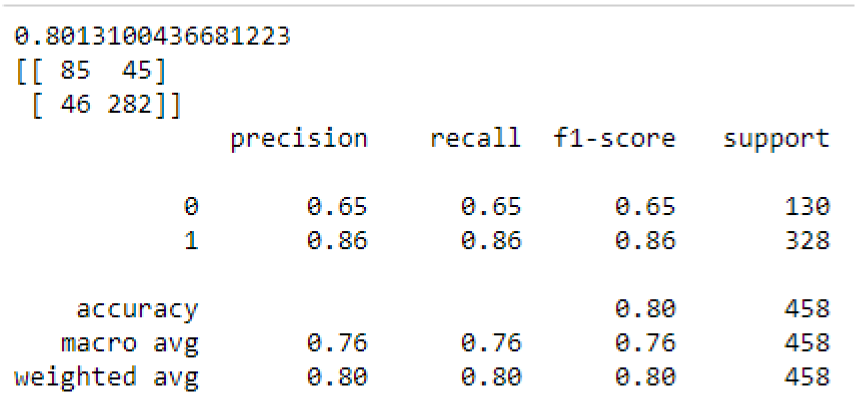
```
BaggingClassifier(base_estimator=DecisionTreeClassifier(), n_estimators=100,
                  random_state=1)
```

Performance matrix with train data



```
0.9990627928772259
[[331   1]
 [  0 735]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       332
           1       1.00      1.00      1.00       735

    accuracy                           1.00      1067
   macro avg       1.00      1.00      1.00      1067
weighted avg       1.00      1.00      1.00      1067
```

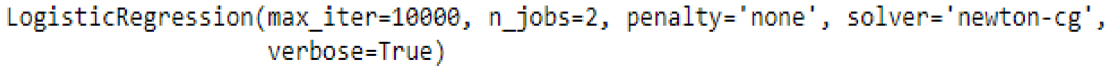
Performance matrix with test data



```
0.8013100436681223
[[ 85  45]
 [ 46 282]]
              precision    recall  f1-score   support

           0       0.65      0.65      0.65       130
           1       0.86      0.86      0.86       328

    accuracy                           0.80       458
   macro avg       0.76      0.76      0.76       458
weighted avg       0.80      0.80      0.80       458
```

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)**

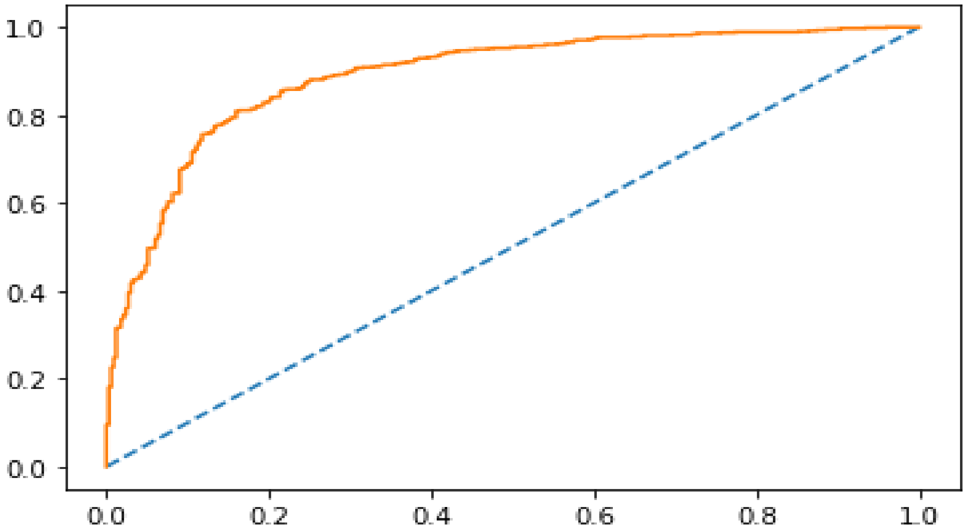
- Logistic regression model



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

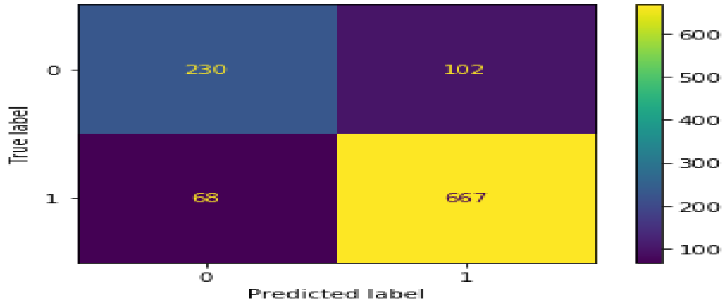
Logistic regression model accuracy (Train data): 0.8406747891283973

```
AUC/ROC Curve: AUC: 0.889
```

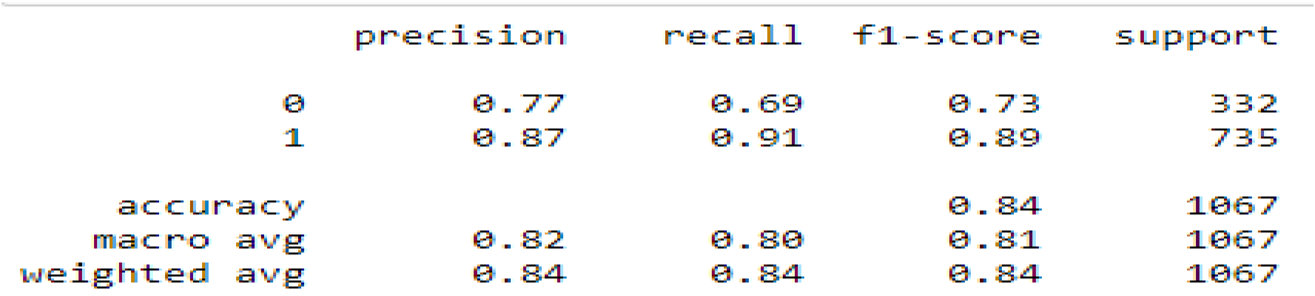


.

```
Confusion matrix (Train data)
```

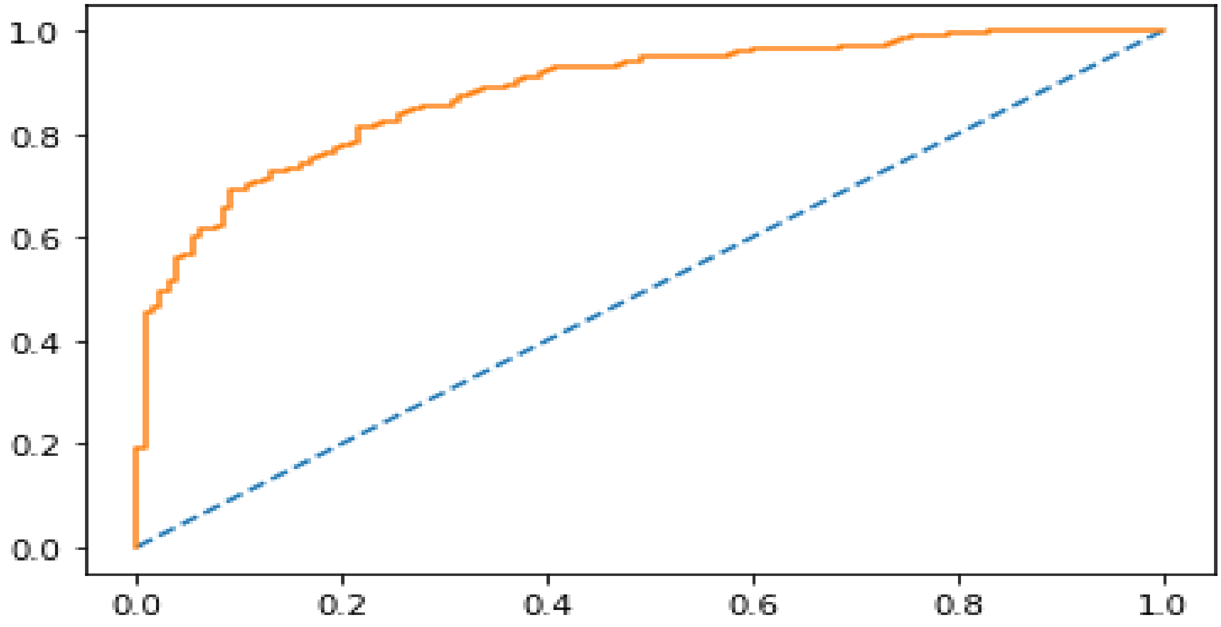


```
Classification report
```

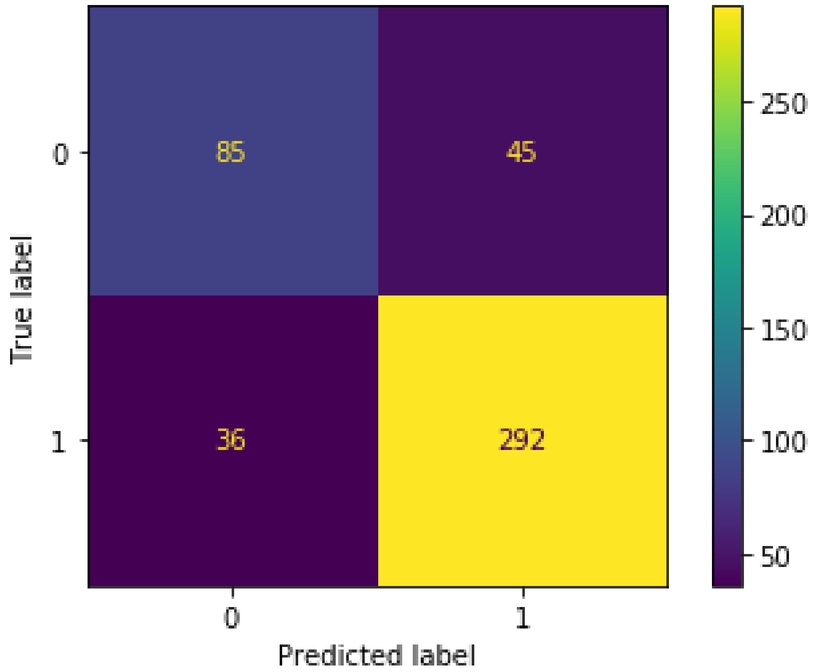


```
              precision    recall  f1-score   support

           0       0.77      0.69      0.73       332
           1       0.87      0.91      0.89       735

    accuracy                           0.84      1067
   macro avg       0.82      0.80      0.81      1067
weighted avg       0.84      0.84      0.84      1067
```

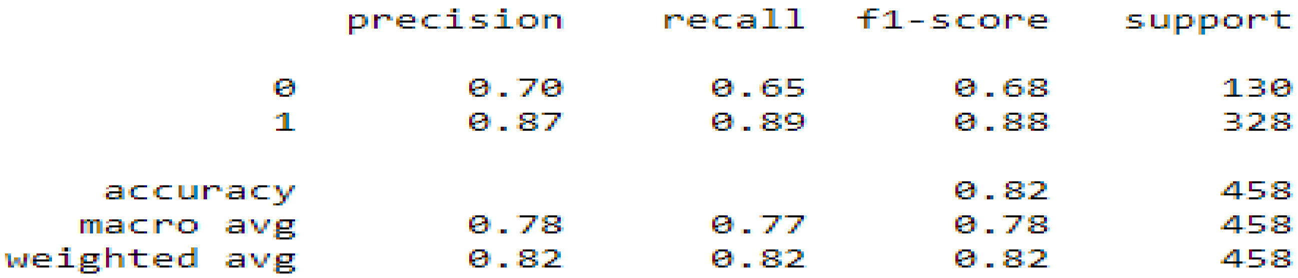
```
Logistic regression model accuracy (Test data): 0.8231441048034934

AUC/ROC Curve AUC: 0.889
```



Confusion matrix (Test Data)

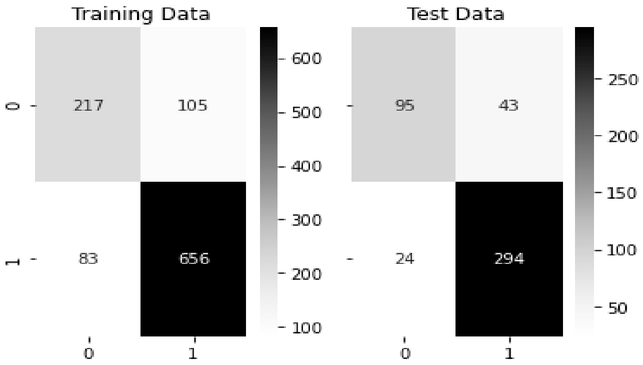


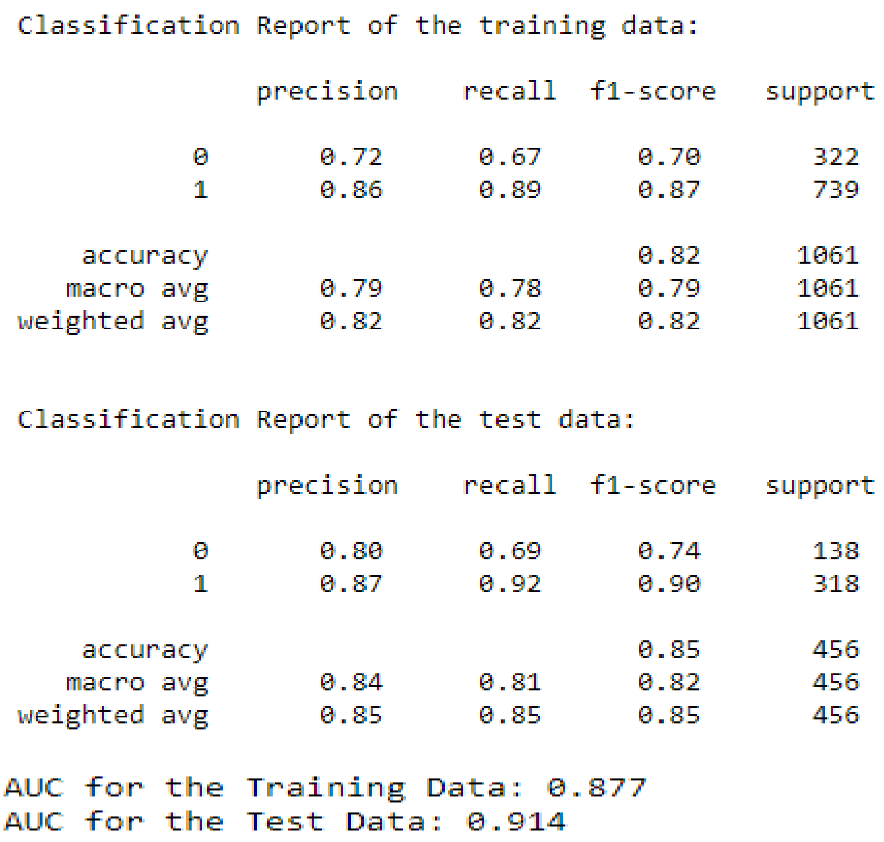
Classification report



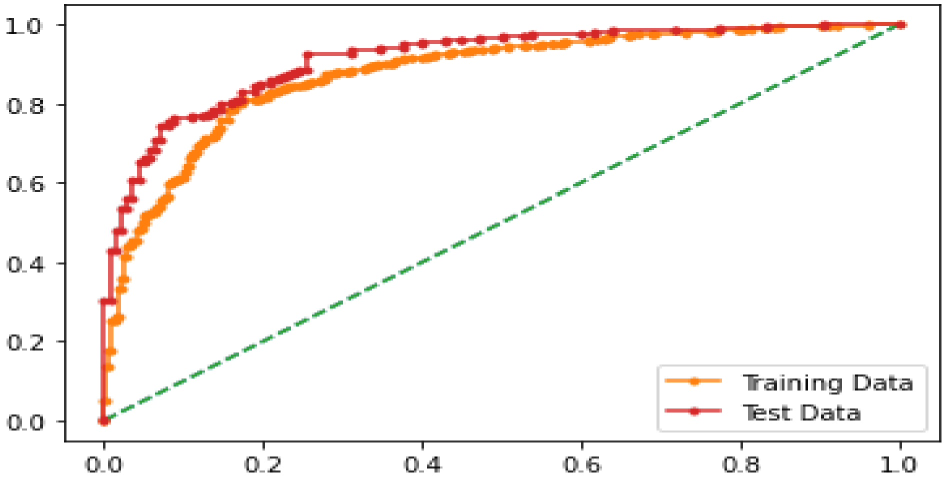
```
              precision    recall  f1-score   support

           0       0.70      0.65      0.68       130
           1       0.87      0.89      0.88       328

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

- LDA Linear Discriminant Analysis

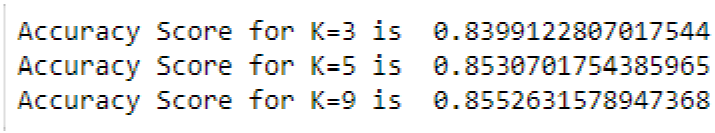




```
Classification Report of the training data:

              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


Classification Report of the test data:

              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

AUC for the Training Data: 0.877
AUC for the Test Data: 0.914
```

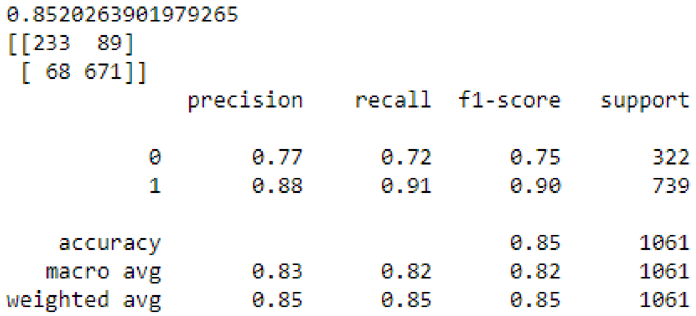


- **KNN**



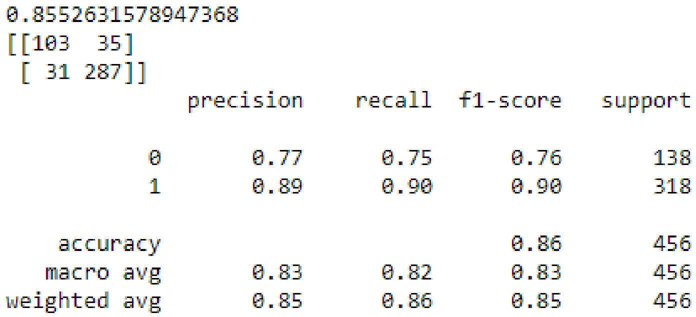
```
Accuracy Score for K=3 is  0.8399122807017544
Accuracy Score for K=5 is  0.8530701754385965
Accuracy Score for K=9 is  0.8552631578947368
```

Model score train data



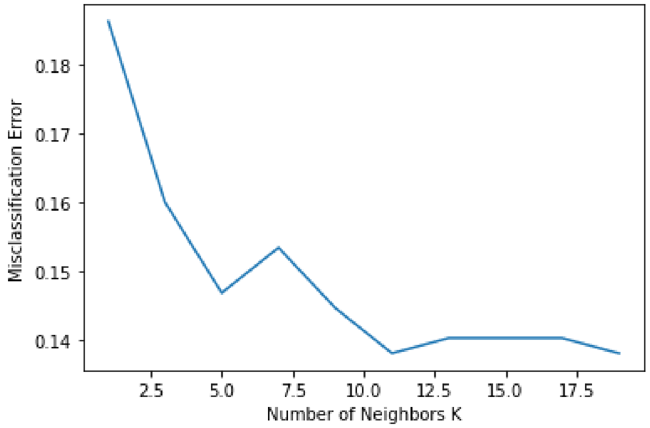
```
0.8520263901979265
[[233  89]
 [ 68 671]]
              precision    recall  f1-score   support

           0       0.77      0.72      0.75       322
           1       0.88      0.91      0.90       739

    accuracy                           0.85      1061
   macro avg       0.83      0.82      0.82      1061
weighted avg       0.85      0.85      0.85      1061
```

Model score test data



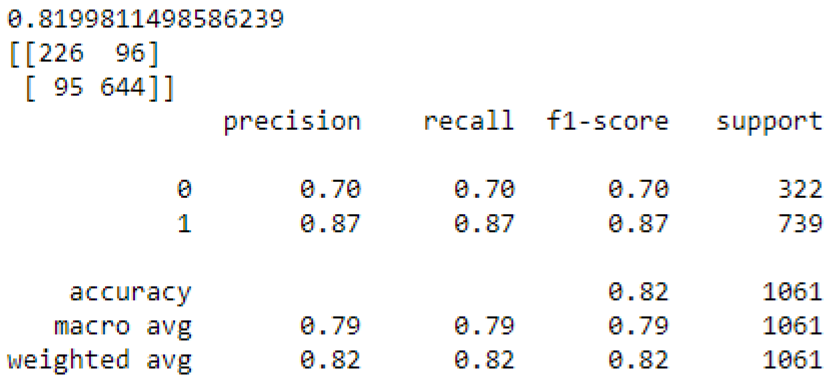
```
0.8552631578947368
[[103  35]
 [ 31 287]]
              precision    recall  f1-score   support

           0       0.77      0.75      0.76       138
           1       0.89      0.90      0.90       318

    accuracy                           0.86       456
   macro avg       0.83      0.82      0.83       456
weighted avg       0.85      0.86      0.85       456
```

Inference: Knn seem be working good for k=5 with least misclassification error



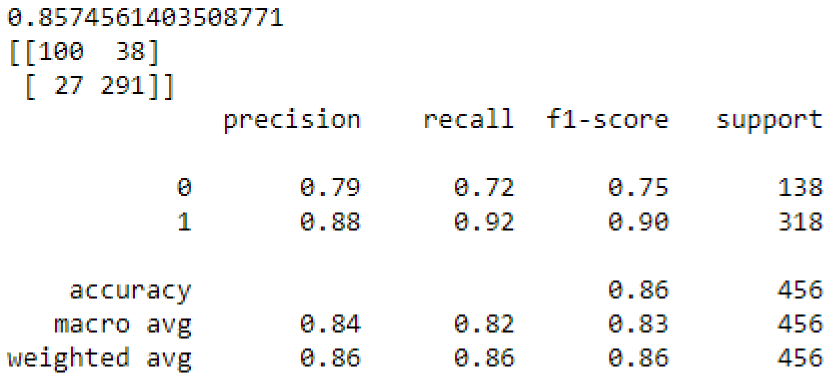
- Naïve bayes

Performance matrix on train data



```
0.8199811498586239
[[226  96]
 [ 95 644]]
              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
```

Performance matrix on test data



```
0.8574561403508771
[[100  38]
 [ 27 291]]
              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
```

Comparison of all four models performance matrix

	Logit			**LDA**			**kNN**			**Naïve Bayes**		
Train data score	84%			87%			83%			81%		
Test data score	82%			82%			85%			82%		
Performance Matrix (Train Data)		A	P		A	P		A	P		A	P
	0	.77	.69	0	.77	.69	0	.77	.72	0	.70	.70
	1	.87	.91	1	.87	.91	1	.88	.91	1	.87	.87
Performance Matrix (Test Data)		A	P		A	P		A	P		A	P
	0	.70	.65	0	.70	.65	0	.77	.75	0	.79	.72
	1	.87	.89	1	.87	.89	1	.89	.90	1	.88	.92

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.**

Feedback

Labour party specific

- Having good economical national condition indicate, chance of being labour party follower is high
- Having good economical household condition indicate, chance of being labour party follower is high
- Labour party follower seem to be less interested in Europe integration comparative to conservative party and with high deviation among follower
- Labour party follower seems to be holding high political knowledge compare to conservative party followers

Conservative party specific

- More the age, chance of being conservative party follower is high
- Conservative party follower seem to be highly interested in Europe integration comparatively labour party and with less deviation among follower
- Conservative party follower seems to be holding low political knowledge compare to labour party followers

Recommendations

- Dataset seem biased towards labour party's 1063 records compare to 462 records of conservative party, kindly check with business if balanced data is available for analysis.
- Blair seems more popular compare to hauge in all age group
- Blair's followers seem to be not in favour of Europe integration
- Hauge's followers seem to be in favour of Europe integration
- Economical condition household and economical condition national seem to highly positively correlated
- Blair's followers seem to be having good economical condition household and economical condition national

Problem 2:

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

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

- Find the number of characters, words and sentences for the mentioned documents. – 3 Marks

(Hint: use .words(), .raw(), .sent() for extracting counts)

- Remove all the stopwords from all the three speeches. – 3 Marks
- Which word occurs the most number of times in his inaugural address for each president? Mention the top three words. (after removing the stopwords) – 3 Marks
- Plot the word cloud of each of the speeches of the variable. (after removing the stopwords) – 3 Marks [refer to the End-to-End Case Study done in the Mentored Learning Session]

Code Snippet to extract the three speeches:

"

```
import nltk
nltk.download('inaugural')
from nltk.corpus import inaugural
inaugural.fileids()
inaugural.raw('1941-Roosevelt.txt')
inaugural.raw('1961-Kennedy.txt')
inaugural.raw('1973-Nixon.txt')
```

"

Important Note: Please reflect on all that you have learned while working on this project. This step is critical in cementing all your concepts and closing the loop. Please write down your thoughts here.

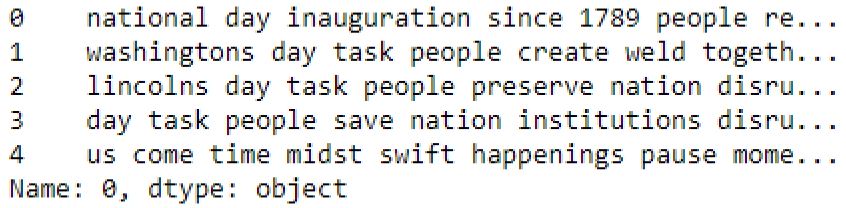
1. Find the number of characters, words and sentences for the mentioned documents. – 3 Marks (Hint: use .words(), .raw(), .sent() for extracting counts)

```
*****************   Roosevelt Speech Results   *********************
Number Of Total Characters:  7262
Number Of Total Words:  1338
Number Of Total Sentence:  38


*****************   Kennedy Speech Results   ***********************
Number Of Total Characters:  7336
Number Of Total Words:  1365
Number Of Total Sentence:  27

*****************   Nixon Speech Results   *************************
Number Of Total Characters:  9646
Number Of Total Words:  1802
Number Of Total Sentence:  51
```

2. Remove all the stopwords from all the three speeches. – 3 Marks

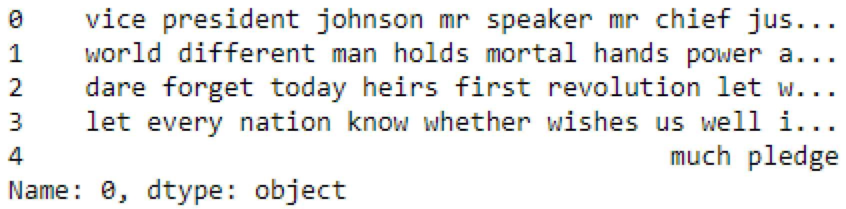
Please check all three speeches after removing stopwords

*******************   Roosevelt Speech *********************



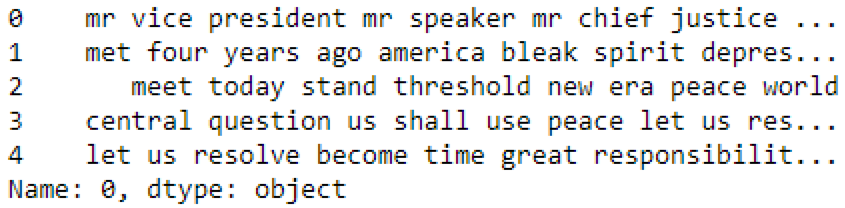
```
0    national day inauguration since 1789 people re...
1    washingtons day task people create weld togeth...
2    lincolns day task people preserve nation disru...
3    day task people save nation institutions disru...
4    us come time midst swift happenings pause mome...
Name: 0, dtype: object
```

*******************   Kennedy Speech ***********************



```
0    vice president johnson mr speaker mr chief jus...
1    world different man holds mortal hands power a...
2    dare forget today heirs first revolution let w...
3    let every nation know whether wishes us well i...
4                                          much pledge
Name: 0, dtype: object
```

*******************   Nixon Speech *************************



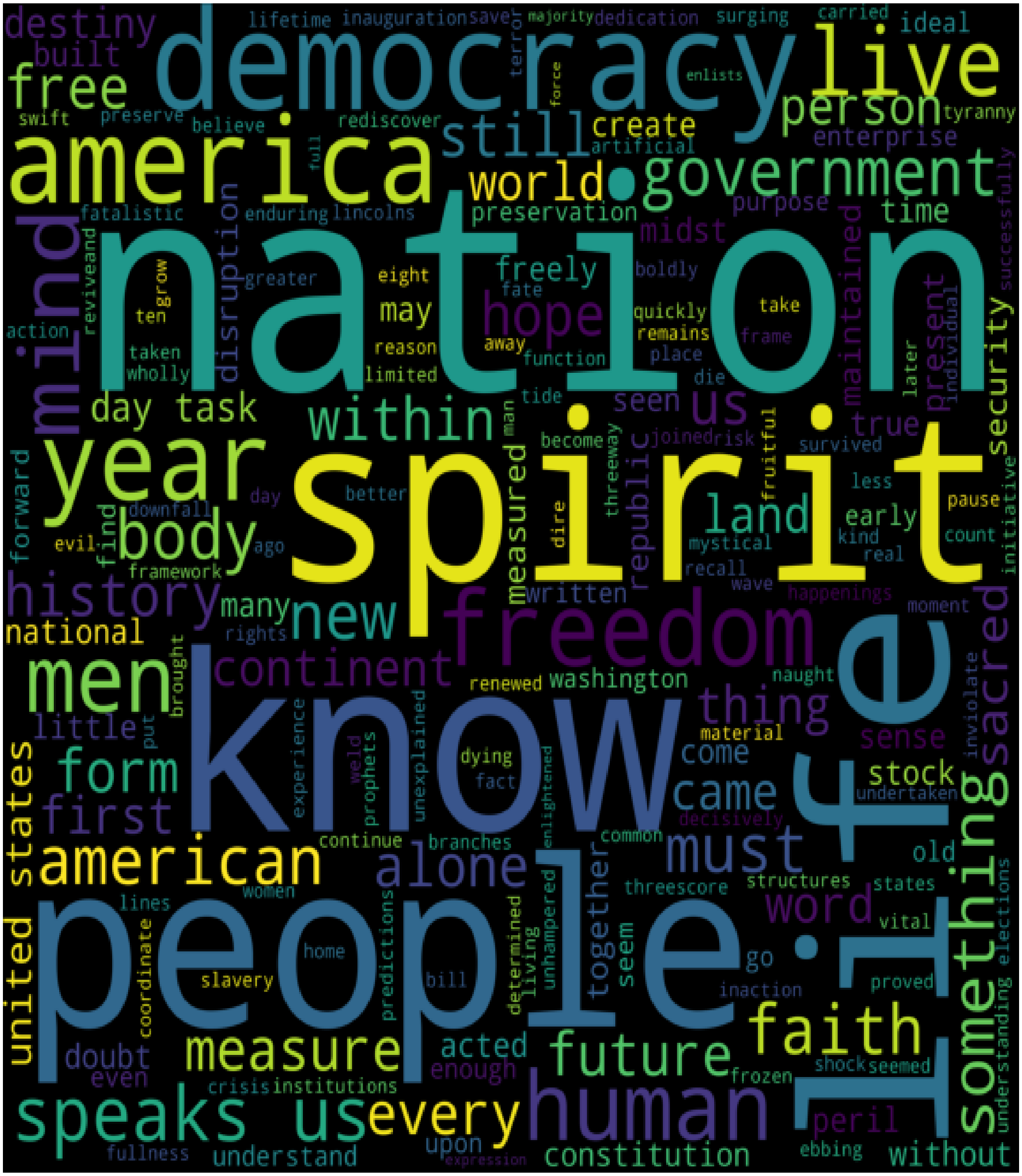
```
0    mr vice president mr speaker mr chief justice ...
1    met four years ago america bleak spirit depres...
2       meet today stand threshold new era peace world
3    central question us shall use peace let us res...
4    let us resolve become time great responsibilit...
Name: 0, dtype: object
```

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) – 3 Marks

```
*****************   Roosevelt Speech *********************
Top three words:
   ○  Nation
   ○  Spirit
   ○  People

*****************   Kennedy Speech ***********************
Top three words:
   ○  Let
   ○  World
   ○  Sides

*****************   Nixon Speech *************************
Top three words:
   ○  World
   ○  Peace
   ○  America
```

4. Plot the word cloud of each of the speeches of the variable. (after removing the stopwords) – 3 Marks [refer to the End-to-End Case Study done in the Mentored Learning Session]

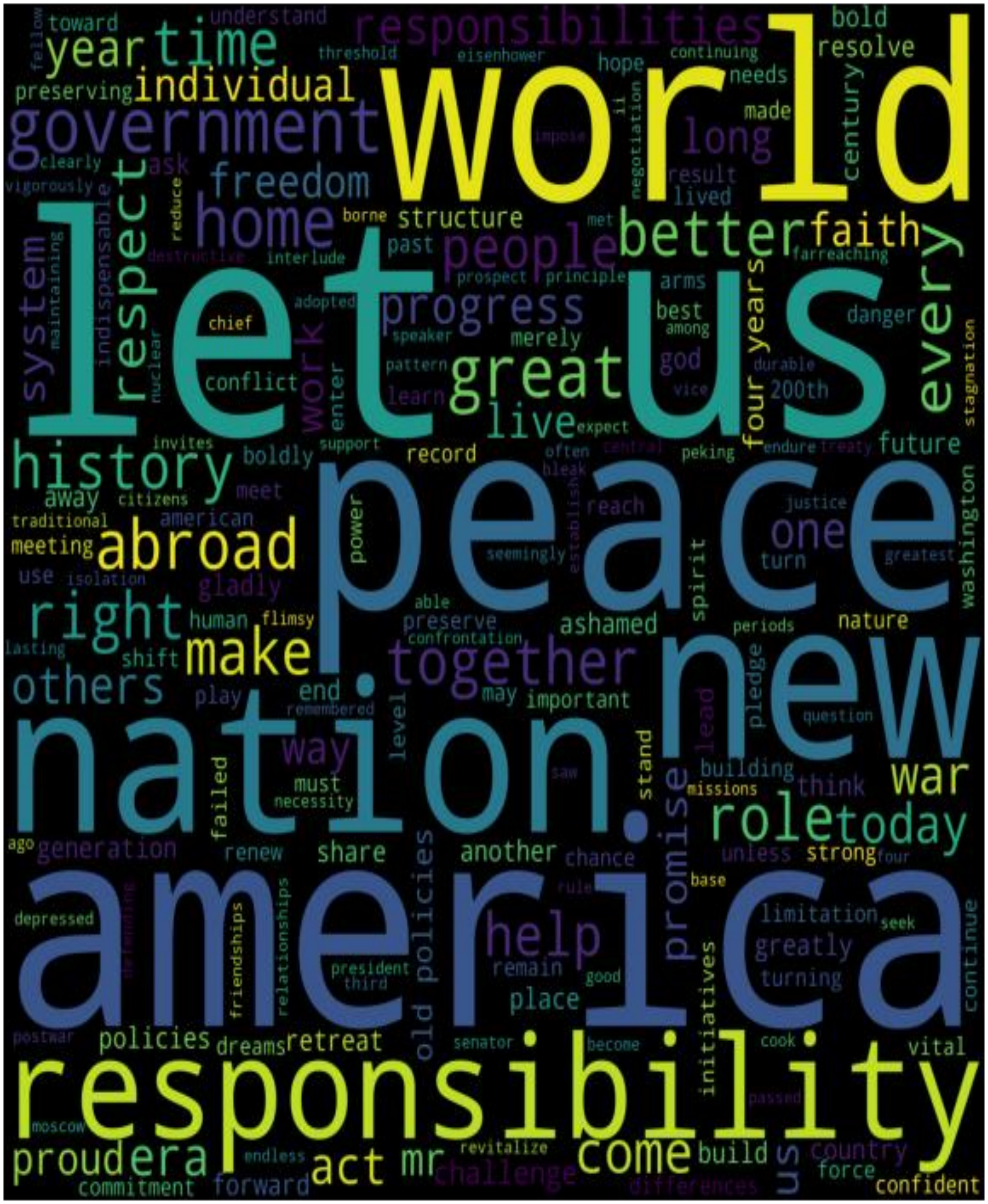
```
*****************   Roosevelt Speech *********************
```



*****************   Kennedy Speech **********************

let
world
sides
nation
pledge
us
power
new
let us
citizens
free
ask
first
generation
president
country
committed
call
hardship
clergy
spiritual
finally
supporting
right
truly
forebears
may
survival
good
science
party
origins
justice
foe
instead
offer
reverend
time
enemies
quarters
mortal
life
weak
final
tyranny
globe
peace
friend
symbolizing
vice
chief
explore
johnson
allies
oath
celebration
fear
burden
whether
forget
find
course
go
passed
hope
beyond
long
ago
join
bear
history
heritage
born
unite
permit
undoing
bring
fellow
help
best
anew
fought
seek
solemn
mankind
bitter
proud
terror
beginning
balance
meet
word
every
problems
disease
heirs
victory
order
today
know
americans
form
little
renewal
alike
effort
witness
man
doubt
ancient
torch
price
nearly
days
home
powerful
merely
negotiate
dare
never
success
people
three
ill
different
belief
earth
disciplined
generosity
prescribed
signifying
south
arms
freedom
poverty
hand
instruments
finished
holds
slow
weakness
control
place
loyalty
almighty
america
state
yet
oppose
tempered
share
endeavor
struggle
always
alliance
together
human
old
change
absolute
end
come
much
around
united
assure
forth
light
issue
well
support
begin
pay
observe
speaker
revolution
far
mr
cultural
year
war
though
back
god
sworn
century
abolish
nixon
eisenhower
wishes
welcome
truman
unwilling
deeds
revolutionary

***************** Nixon Speech *************************


world
let us
peace
new
nation
america
responsibility
responsibilities
government
year
time
toward
understand
bold
resolve
fellow
threshold
eisenhower
continuing
preserving
individual
hope
needs
made
long
century
impose
negotiation
clearly
ask
result
lived
vigorously
freedom
home
people
better
faith
structure
borne
met
reduce
indispensable
respect
system
destructive
interlude
past
farreaching
prospect
principle
arms
maintaining
adopted
progress
best
among
danger
every
chief
speaker
merely
great
four years
nuclear
pattern
god
durable
stagnation
conflict
work
enter
learn
vice
200th
live
expect
history
invites
support
boldly
record
often
bleak
central
peking
endure
treaty
future
away
citizens
meet
establish
justice
washington
traditional
american
reach
one
meeting
abroad
seemingly
turn
greatest
power
spirit
use
isolation
gladly
right
able
human
flimsy
preserve
ashamed
periods
nature
lasting
confrontation
shift
make
together
others
play
end
may
important
remembered
level
lead
pledge
question
way
must
necessity
failed
stand
building
saw
think
war
missions
role
today
promise
ago
generation
renew
share
another
chance
unless
strong
four
old policies
rule
base
depressed
defending
friendships
relationships
help
limitation
seek
greatly
president
third
remain
good
turning
continue
initiatives
place
postwar
policies
dreams
retreat
senator
become
cook
vital
passed
moscow
endless
revitalize
proud
era
act
mr
come
build
us
country
challenge
force
commitment
forward
differences
confident