Machine Learning ASSIGMENT

<u>Submitted By</u>

Vinit Sharma

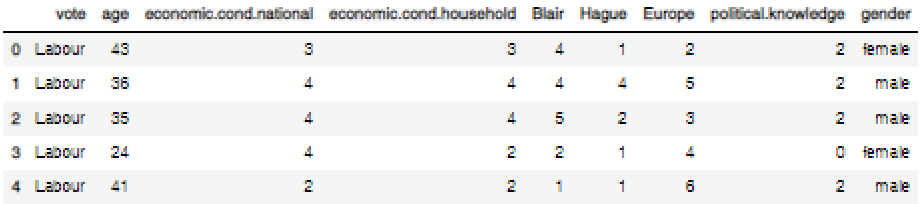
Table of Contents

Problem 1:

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

1.1 Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it. (4 Marks)

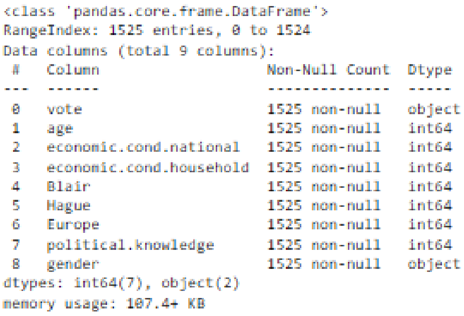
This dataset is based on election poll details for certain city. This dataset has 1525 rows with 9 variables. Out of 9, two variables contain object data type and rest seven variables contain int datatype. In the below figure, we have shown the dataset of voters.



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

`

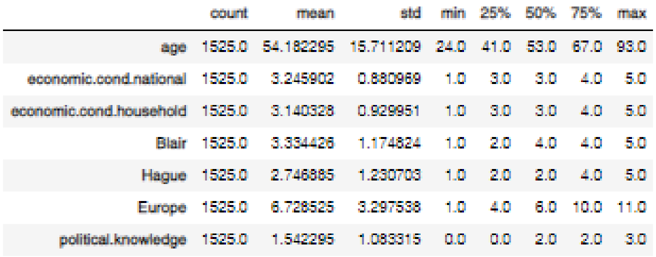
FIGURE 1: VOTERS DATASET



```
<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
```

FIGURE 2 DATATYPE INFORMATION OF VARIABLES

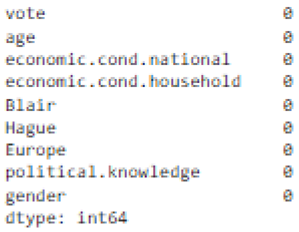
Within 9 variables, seven variables contain int datatype for which description detailed table is mentioned below. From the table, we can infer that minimum voter age is 24 and maximum 93. According to age prospective, 50% casted voter age is in range 24-53. Out of seven numerical variables, 4 variables contain almost same min, max, and other percentiles values. These 4 variables mean denotes almost equally distributed dataset. Dataset contains 1525 row count. Blair and Hague contain approx. same mean, min, max and other percentile details. Mean and Max value of Europe is more than other two area (Blair and Hague).



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

FIGURE 3 DATASET DESCRIPTION

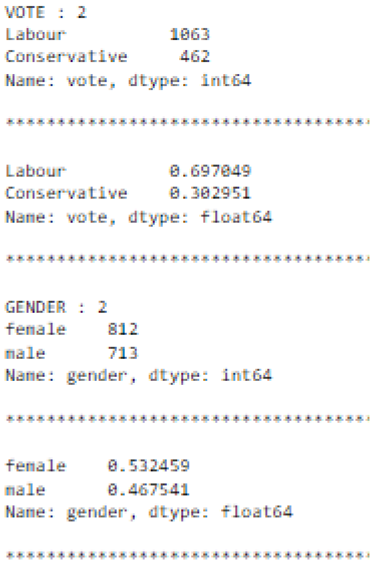
This dataset does not contain null value for any variable.



```
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
```

FIGURE 4 NULL VALUE IDENTIFICATION

Vote and Gender are two categorical variables. Below table shows the count split on both variables. Vote is in 70:30 ratio and gender is in 53:46 ratio.

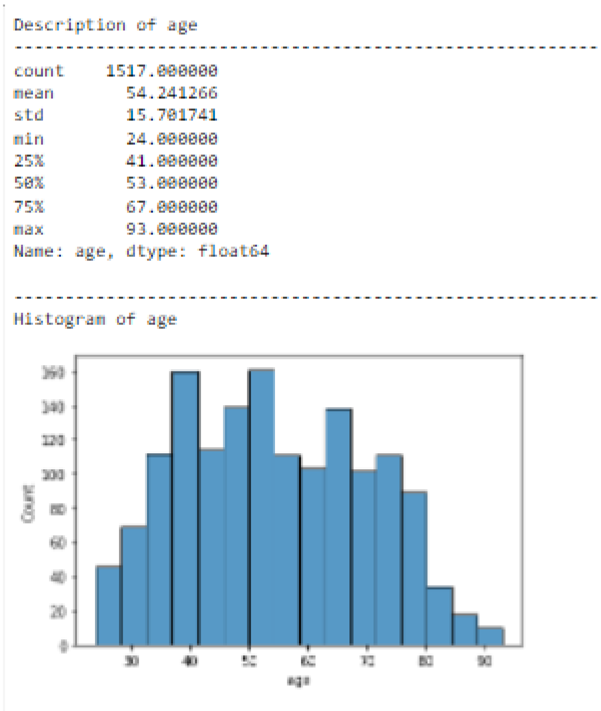


```
VOTE :  2
Labour          1063
Conservative     462
Name: vote, dtype: int64

************************************

Labour          0.697049
Conservative    0.302951
Name: vote, dtype: float64

************************************

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

************************************

female    0.532459
male      0.467541
Name: gender, dtype: float64

************************************
```

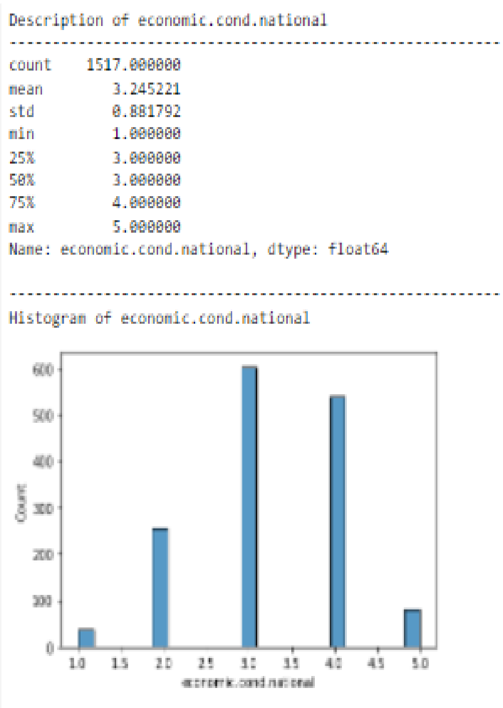
Dataset contains total 1525 row count in which 8 row shows the duplicate. We have to treat the duplicate rows. After dropping the 8 rows, we have 1517 unique rows.

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

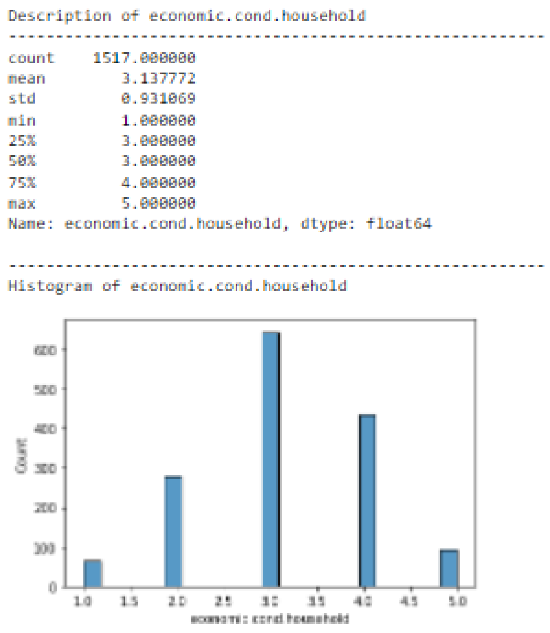
Univariate analysis has been done for all the variable. Each variable descriptive and histograms analysis is shown in below figures.



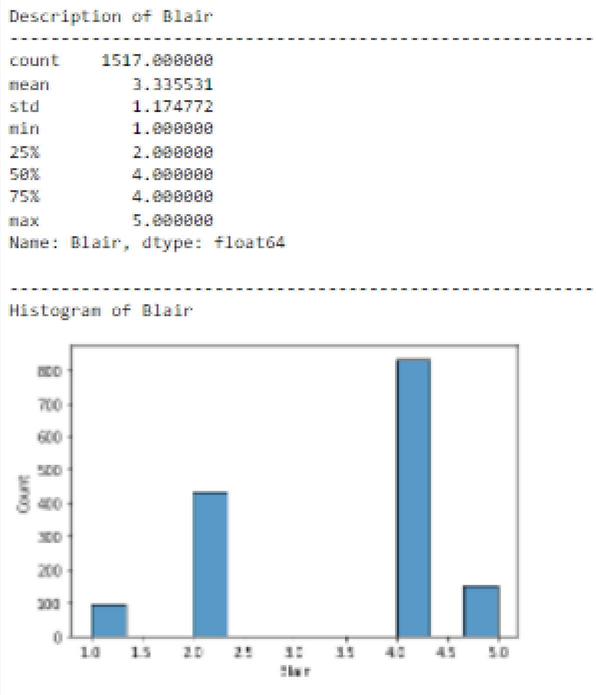
```
Description of age
------------------------------------------------------------
count    1517.000000
mean       54.241266
std        15.701741
min        24.000000
25%        41.000000
50%        53.000000
75%        67.000000
max        93.000000
Name: age, dtype: float64

------------------------------------------------------------
Histogram of age
```



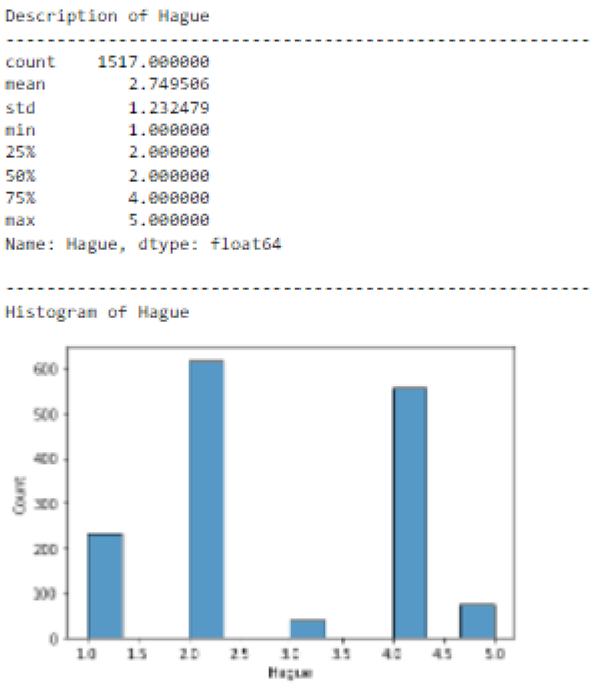
```
Description of economic.cond.national
------------------------------------------------------------
count    1517.000000
mean        3.245221
std         0.881792
min         1.000000
25%         3.000000
50%         3.000000
75%         4.000000
max         5.000000
Name: economic.cond.national,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

------------------------------------------------------------
Histogram of economic.cond.national
```



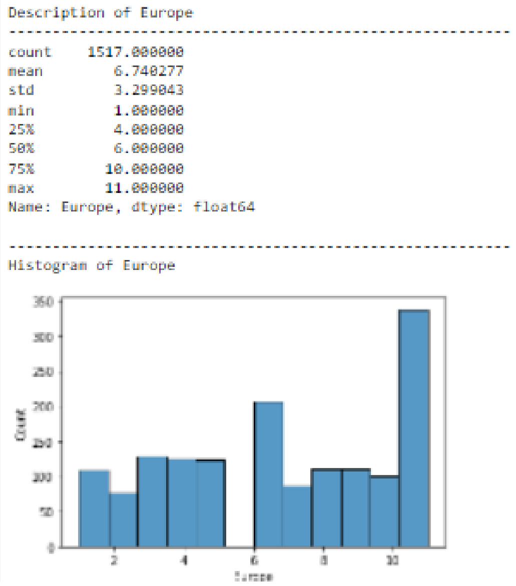
```
Description of economic.cond.household
------------------------------------------------------------
count    1517.000000
mean        3.137772
std         0.931069
min         1.000000
25%         3.000000
50%         3.000000
75%         4.000000
max         5.000000
Name: economic.cond.household,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

------------------------------------------------------------
Histogram of economic.cond.household
```



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

------------------------------------------------------------
Histogram of Blair
```

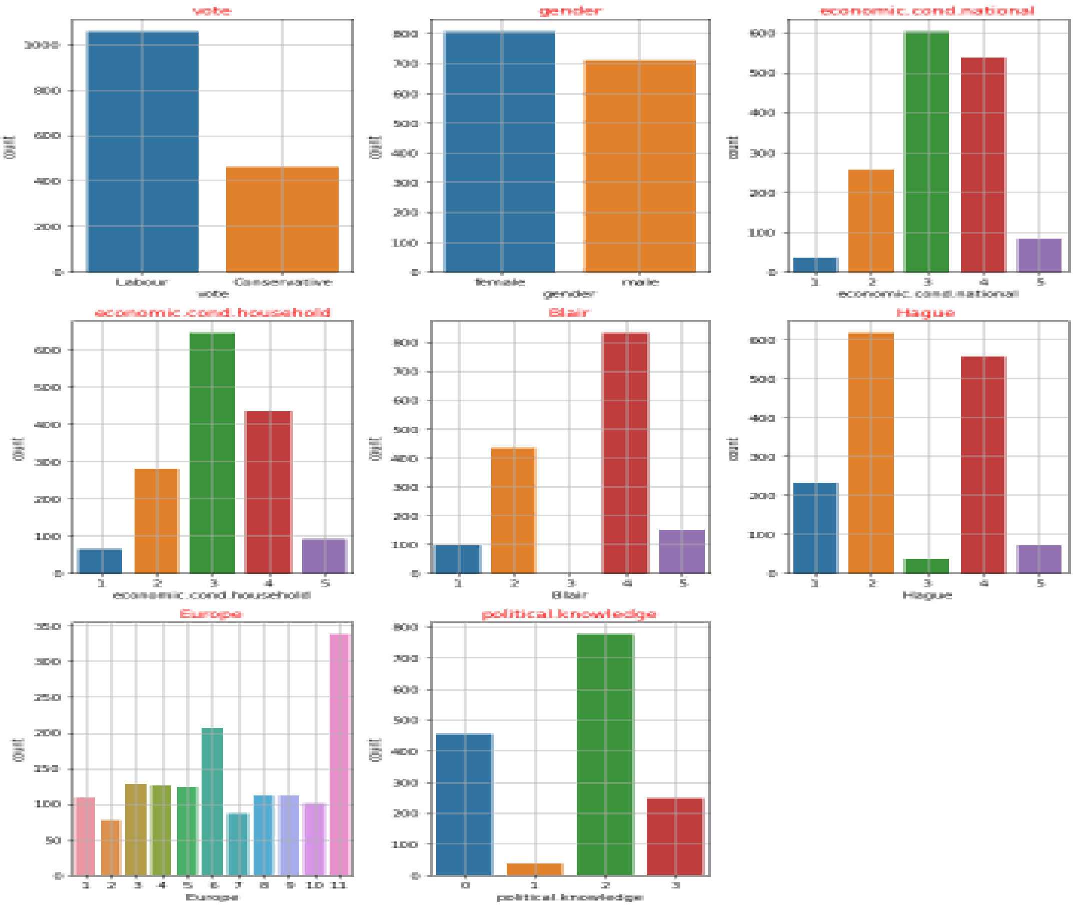


```
Description of Hague
----------------------------------------------------------
count    1517.000000
mean        2.749506
std         1.232479
min         1.000000
25%         2.000000
50%         2.000000
75%         4.000000
max         5.000000
Name: Hague, dtype: float64

----------------------------------------------------------
Histogram of Hague
```



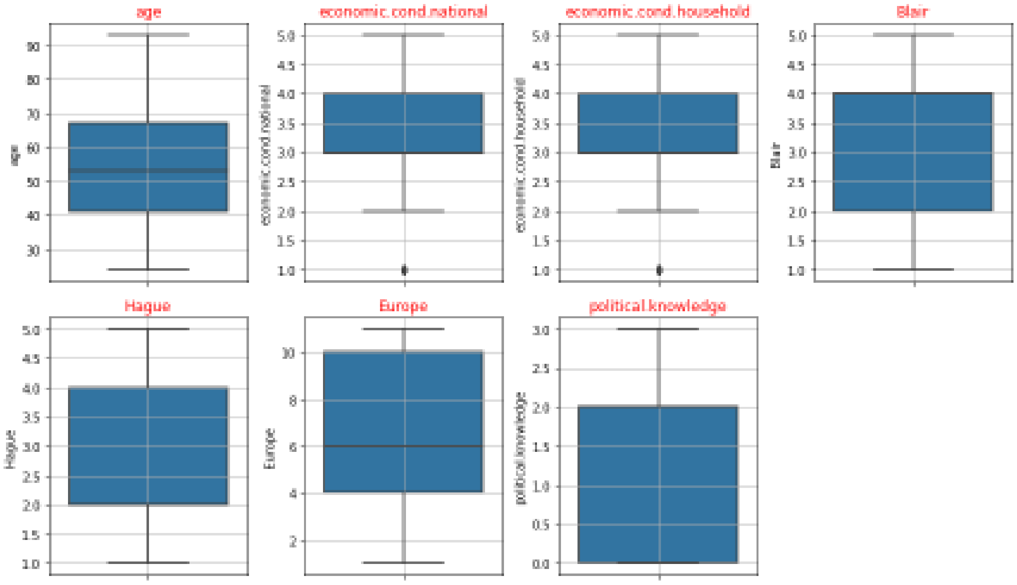
```
Description of Europe
----------------------------------------------------------
count    1517.000000
mean        6.740277
std         3.299043
min         1.000000
25%         4.000000
50%         6.000000
75%        10.000000
max        11.000000
Name: Europe, dtype: float64

----------------------------------------------------------
Histogram of Europe
```

In an analysis on individual variable has shown in the figure. We have noticed that in vote segment, Labour class count is more than conservative class. In same way, gender is almost equal in count. In Economic condition national, 1 and 5 is almost less and 3 is highest in count. In same way other, bar parameter shows in variations.

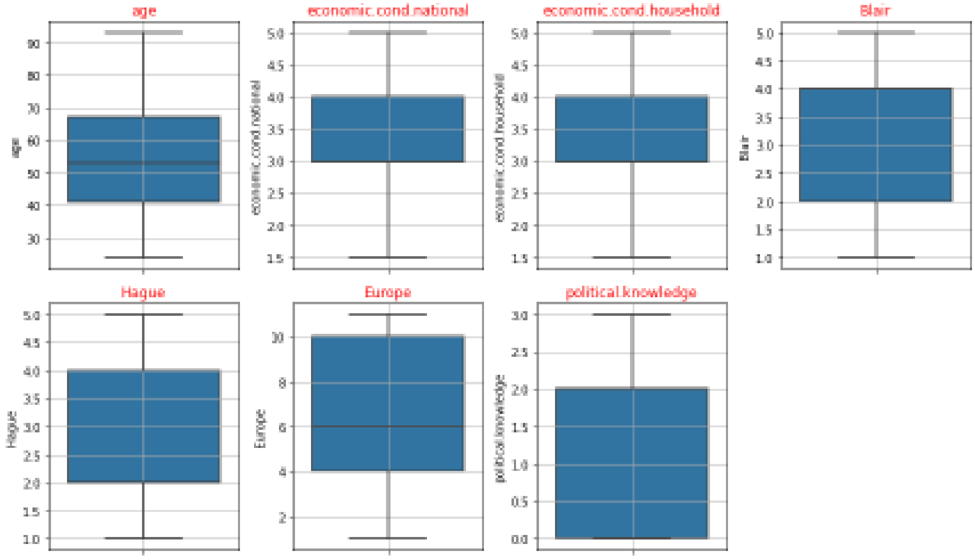


Before Outlier Treatment:

As per below, figure variables contains outliers.



After Outlier Treatment



We have plotted the pairplot among all the variables. In a same way, we also have plotted correlation matrix as well where it reflects the heatmap.

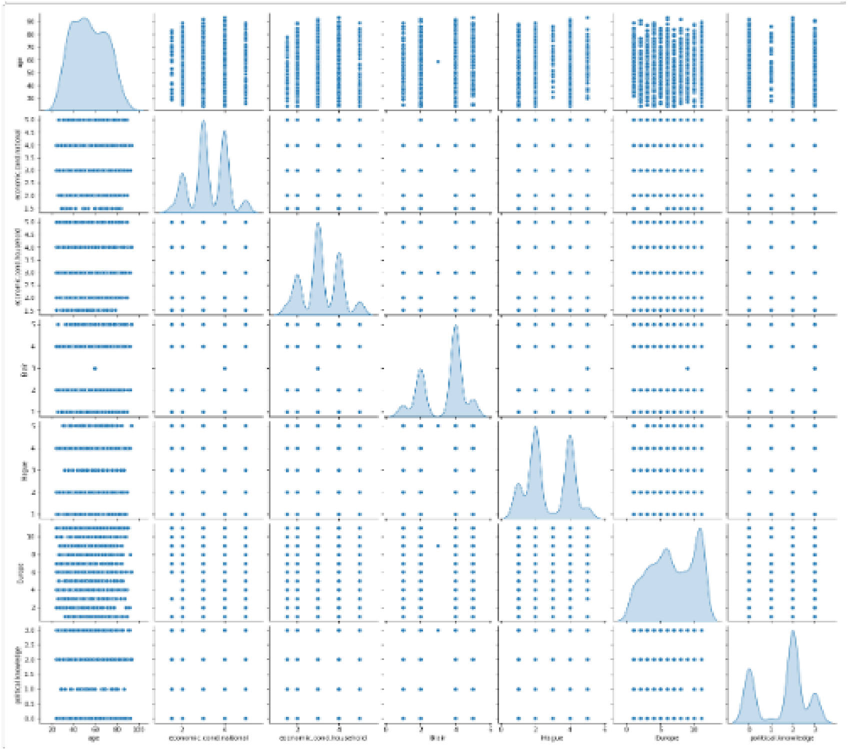


FIGURE 5 PAIRPLOT OF DATASET

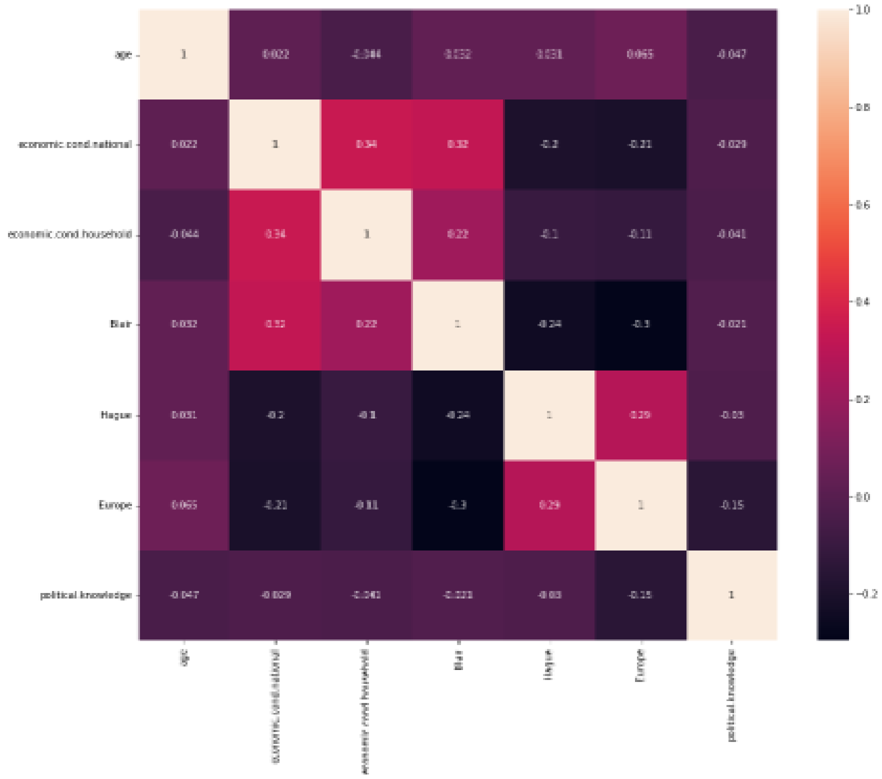
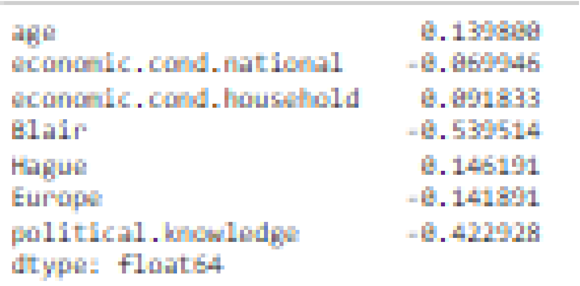


FIGURE 6 CORRELATION MATRIX

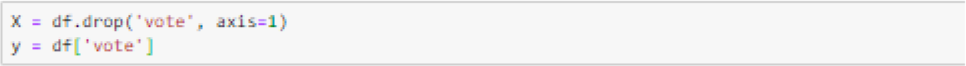
Skewness among the variables have shown in the below table. 4 variables contain negative skewness and 3 variable contain positive skewness.



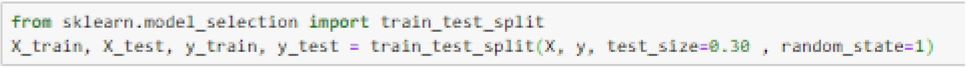
```
age                        0.139800
economic.cond.national    -0.869946
economic.cond.household    0.091833
Blair                     -0.539514
Hague                      0.146191
Europe                    -0.141891
political.knowledge       -0.422928
dtype: float64
```

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). (4 Marks)

We have used the Label Encoding process to encode the string value of Vote and gender variable to numeric form. Detailed process of encoding is available in code file. Dataset variable variation is not much so it is not necessary to do scaling. Data set split in the form of 70:30 is done. Dataset is splitted into train and test.



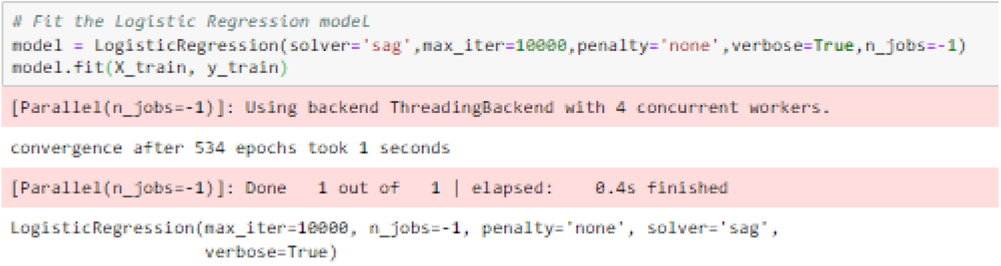
```
X = df.drop('vote', axis=1)
y = df['vote']
```



```
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)
```

1.4 Apply Logistic Regression and LDA (linear discriminant analysis). (4 marks)

Logistic Regression without Grid Search CV:



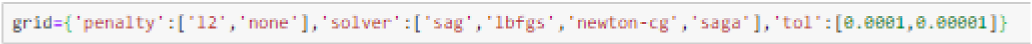
```
# Fit the Logistic Regression model
model = LogisticRegression(solver='sag',max_iter=10000,penalty='none',verbose=True,n_jobs=-1)
model.fit(X_train, y_train)
```

```
[Parallel(n_jobs=-1)]: Using backend ThreadingBackend with 4 concurrent workers.
convergence after 534 epochs took 1 seconds
[Parallel(n_jobs=-1)]: Done   1 out of   1 | elapsed:    0.4s finished
LogisticRegression(max_iter=10000, n_jobs=-1, penalty='none', solver='sag',
                   verbose=True)
```

Accuracy score for train dataset in logistic Regression is 0.8341187

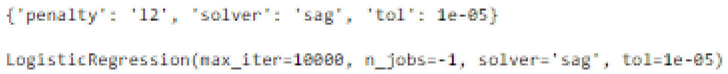
Accuracy score for test dataset in logistic Regression is 0.826754

Logistic Regression with Grid Search CV:



```
grid={'penalty':['l2','none'],'solver':['sag','lbfgs','newton-cg','saga'],'tol':[0.0001,0.00001]}
```

Best Estimators for Logistic Regression:

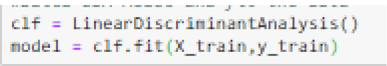


```
{'penalty': 'l2', 'solver': 'sag', 'tol': 1e-05}

LogisticRegression(max_iter=10000, n_jobs=-1, solver='sag', tol=1e-05)
```

Accuracy score for train dataset in logistic Regression is 0.8341187

Accuracy score for test dataset in logistic Regression is 0.826754

LDA without Grid Search CV:



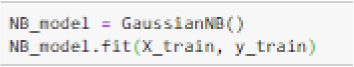
```
clf = LinearDiscriminantAnalysis()
model = clf.fit(X_train,y_train)
```

Accuracy score for train dataset in logistic Regression is 0.8341187

Accuracy score for test dataset in logistic Regression is 0.833333

1.5 Apply KNN Model and Naïve Bayes Model. Interpret the results. (4 marks)

NB Model:

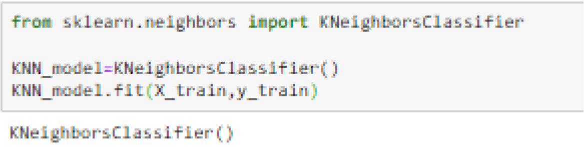


```
NB_model = GaussianNB()
NB_model.fit(X_train, y_train)
```

Accuracy score for train dataset in NB Model is 0.835061

Accuracy score for test dataset in NB Model is 0.82236842

KNN Model:



```
from sklearn.neighbors import KNeighborsClassifier

KNN_model=KNeighborsClassifier()
KNN_model.fit(X_train,y_train)
```

```
KNeighborsClassifier()
```

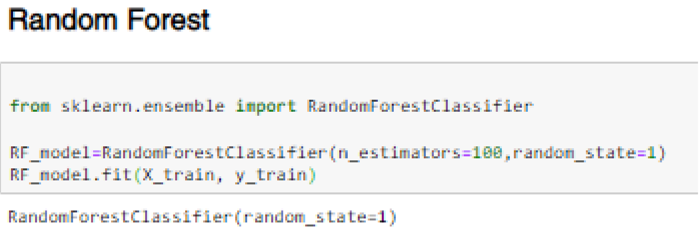
Accuracy score for train dataset in KNN Model is 0.8557964

Accuracy score for test dataset in KNN Model is 0.82456140

Based on result of KNN and NB, we can comment that KNN model has better accuracy compare to NB Model.

1.6 Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting. (7 marks)

Random Forest without Grid Search CV:



Random Forest

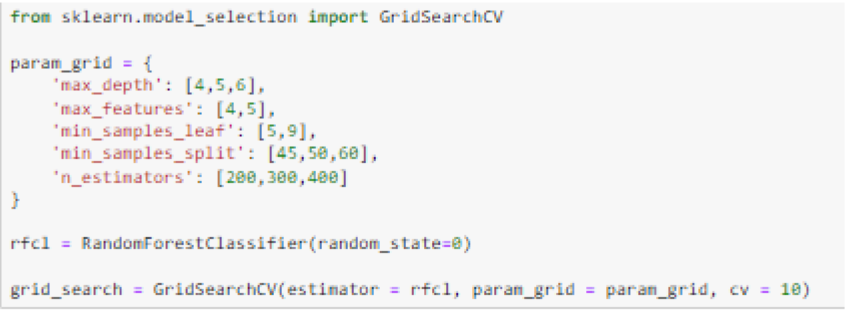
```
from sklearn.ensemble import RandomForestClassifier

RF_model=RandomForestClassifier(n_estimators=100,random_state=1)
RF_model.fit(X_train, y_train)
```

```
RandomForestClassifier(random_state=1)
```

Accuracy score for train dataset in Random Forest is 1.0

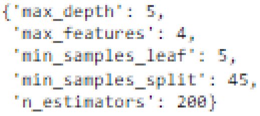
Accuracy score for test dataset in Random Forest is 0.831140

Random Forest with Grid Search CV:



```
from sklearn.model_selection import GridSearchCV

param_grid = {
    'max_depth': [4,5,6],
    'max_features': [4,5],
    'min_samples_leaf': [5,9],
    'min_samples_split': [45,50,60],
    'n_estimators': [200,300,400]
}

rfcl = RandomForestClassifier(random_state=0)

grid_search = GridSearchCV(estimator = rfcl, param_grid = param_grid, cv = 10)
```

Best Estimators for Random Forest:

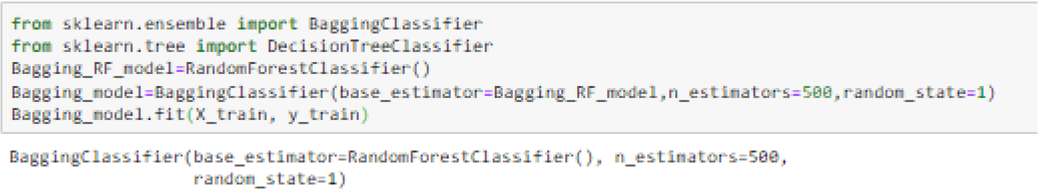


```
{'max_depth': 5,
 'max_features': 4,
 'min_samples_leaf': 5,
 'min_samples_split': 45,
 'n_estimators': 200}
```

Accuracy score for train dataset in Random Forest is 0.856738

Accuracy score for test dataset in Random Forest is 0.8377192

Bagging:



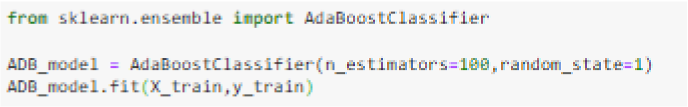
```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
Bagging_RF_model=RandomForestClassifier()
Bagging_model=BaggingClassifier(base_estimator=Bagging_RF_model,n_estimators=500,random_state=1)
Bagging_model.fit(X_train, y_train)
```

```
BaggingClassifier(base_estimator=RandomForestClassifier(), n_estimators=500,
                  random_state=1)
```

Accuracy score for train dataset in Bagging Classifier is 0.969839

Accuracy score for test dataset in Bagging Classifier is 0.83114035

AdaBoosting:

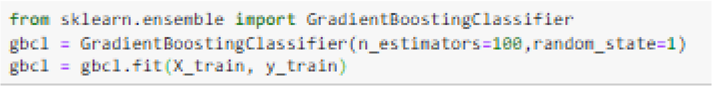


```
from sklearn.ensemble import AdaBoostClassifier

ADB_model = AdaBoostClassifier(n_estimators=100,random_state=1)
ADB_model.fit(X_train,y_train)
```

Accuracy score for train dataset in AdaBoosting is 0.8501413

Accuracy score for test dataset in AdaBoosting is 0.81359649

Gradient Boosting:



```
from sklearn.ensemble import GradientBoostingClassifier
gbcl = GradientBoostingClassifier(n_estimators=100,random_state=1)
gbcl = gbcl.fit(X_train, y_train)
```

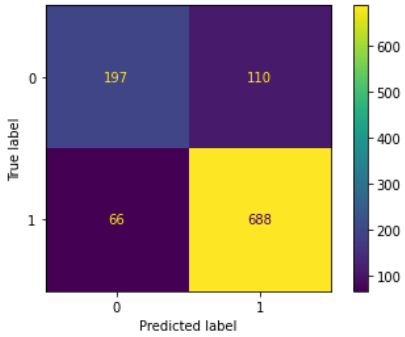
Accuracy score for train dataset in AdaBoosting is 0.8925541

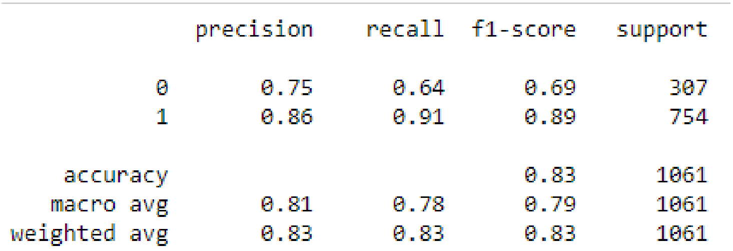
Accuracy score for test dataset in AdaBoosting is 0.83552631

1.7 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized. (7 marks)

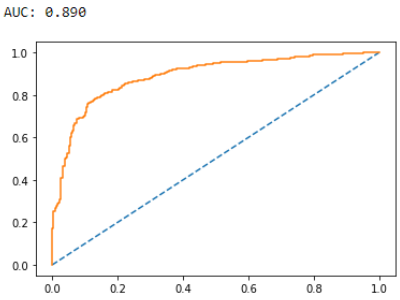
All the models performance metrics and predicitions on train and test set has done. We have also shown Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model.

Logistic Regression without Grid SearchCV: Train Dataset

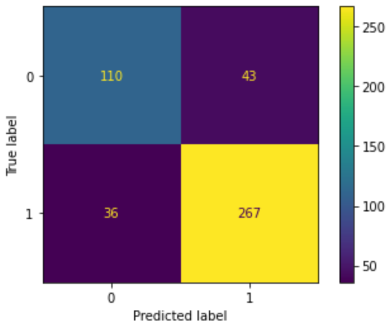


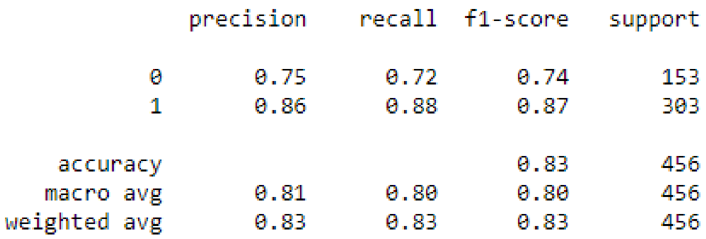


```
              precision    recall  f1-score   support

           0       0.75      0.64      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.83      1061
   macro avg       0.81      0.78      0.79      1061
weighted avg       0.83      0.83      0.83      1061
```

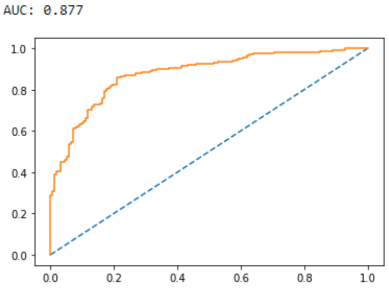


Logistic Regression without Grid Search CV: Test Dataset

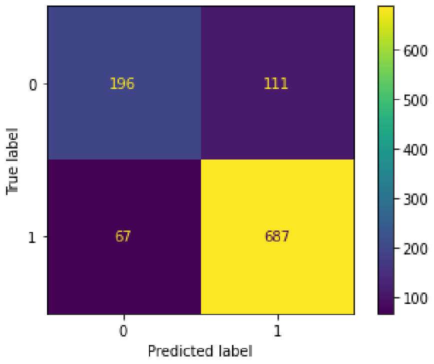


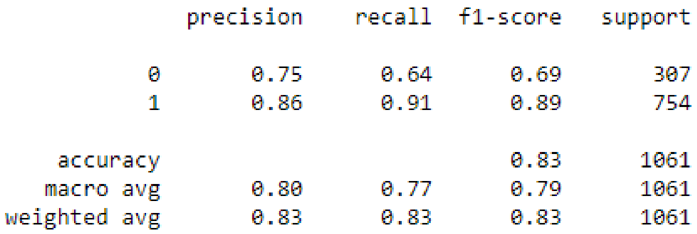


```
              precision    recall  f1-score   support

           0       0.75      0.72      0.74       153
           1       0.86      0.88      0.87       303

    accuracy                           0.83       456
   macro avg       0.81      0.80      0.80       456
weighted avg       0.83      0.83      0.83       456
```



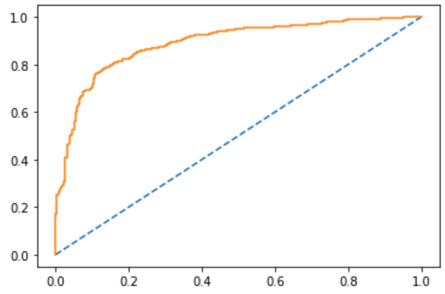
Logistic Regression with Grid SearchCV: Train Dataset



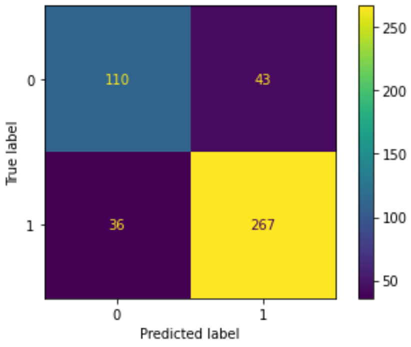


```
              precision    recall  f1-score   support

           0       0.75      0.64      0.69       307
           1       0.86      0.91      0.89       754

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

```
AUC: 0.890
```

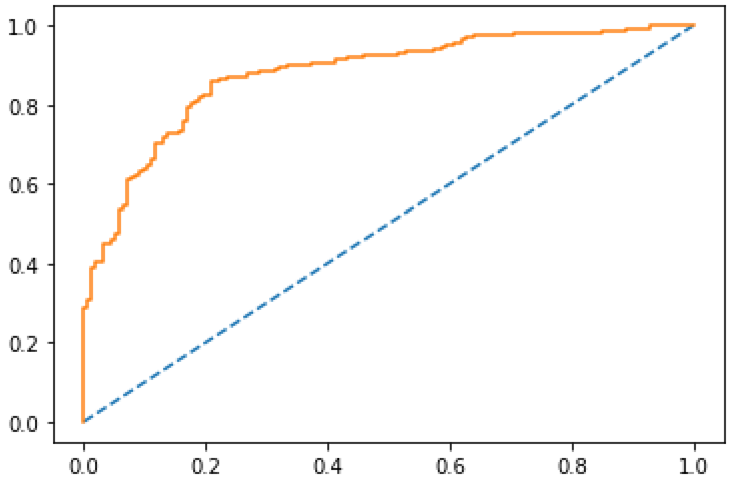


Logistic Regression with Grid SearchCV: Test Dataset

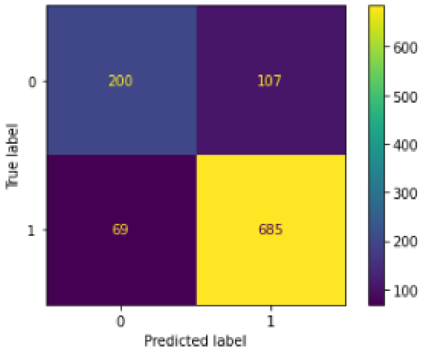


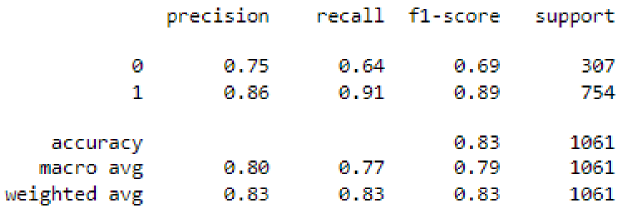
```
              precision    recall  f1-score   support

           0       0.75      0.72      0.74       153
           1       0.86      0.88      0.87       303

    accuracy                           0.83       456
   macro avg       0.81      0.80      0.80       456
weighted avg       0.83      0.83      0.83       456
```

```
AUC: 0.877
```



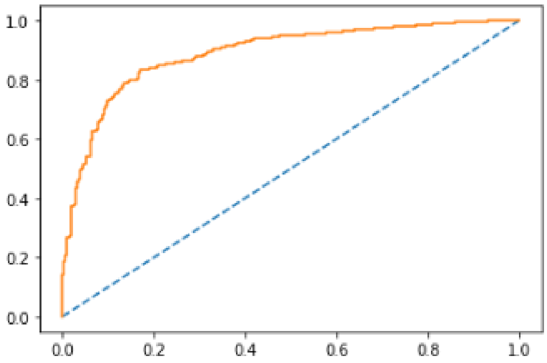
LDA: Train Dataset



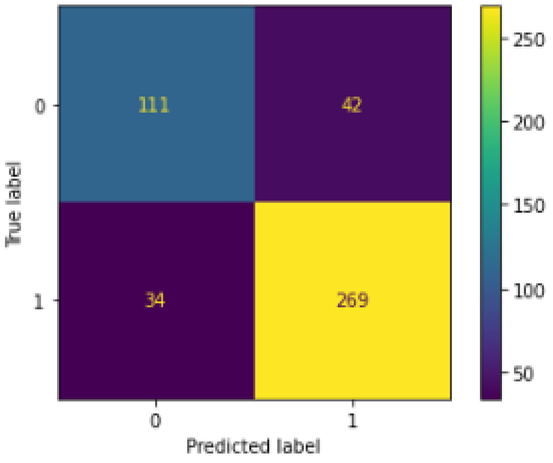


```
              precision    recall  f1-score   support

           0       0.75      0.64      0.69       307
           1       0.86      0.91      0.89       754

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

```
AUC: 0.889
```

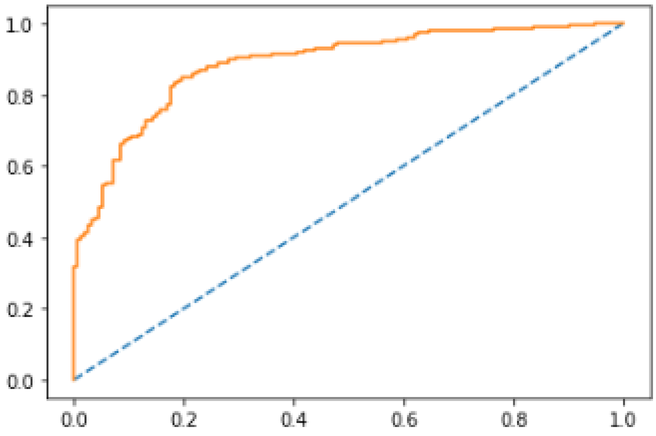


LDA: Test Dataset

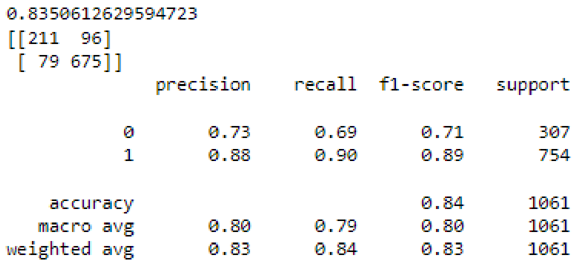


```
Accuracy Score is  0.8333333333333334
```

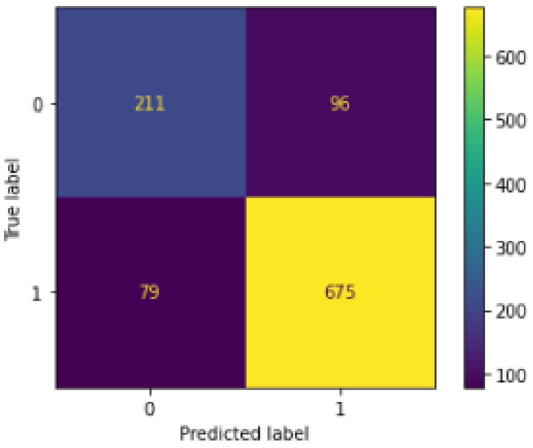
```
AUC: 0.888
```



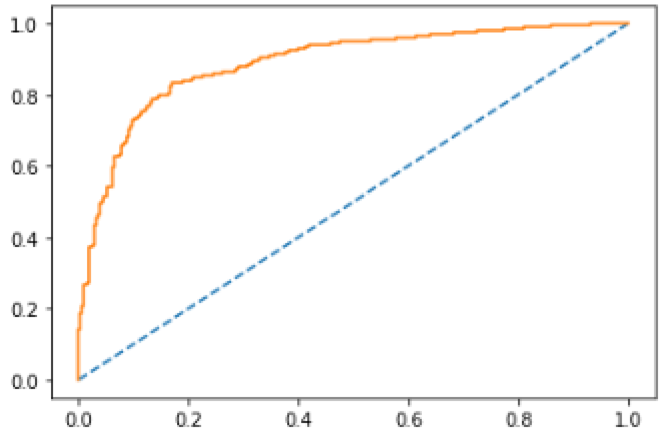
Naïve Bayes Model: Train Dataset



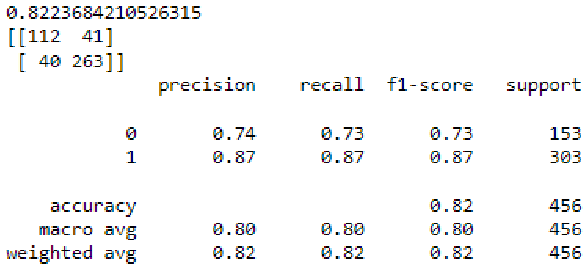
```
0.8350612629594723
[[211  96]
 [ 79 675]]
              precision    recall  f1-score   support

           0       0.73      0.69      0.71       307
           1       0.88      0.90      0.89       754

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



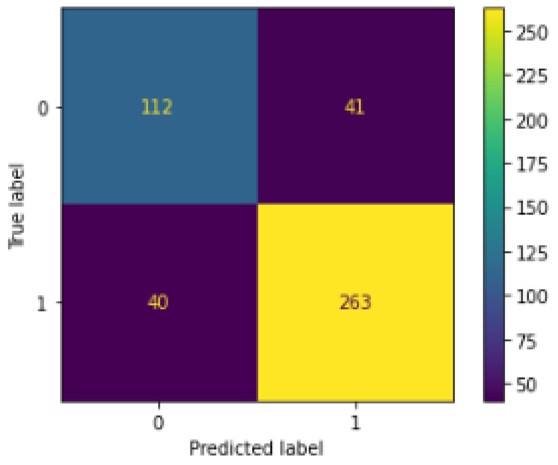
```
AUC: 0.889
```



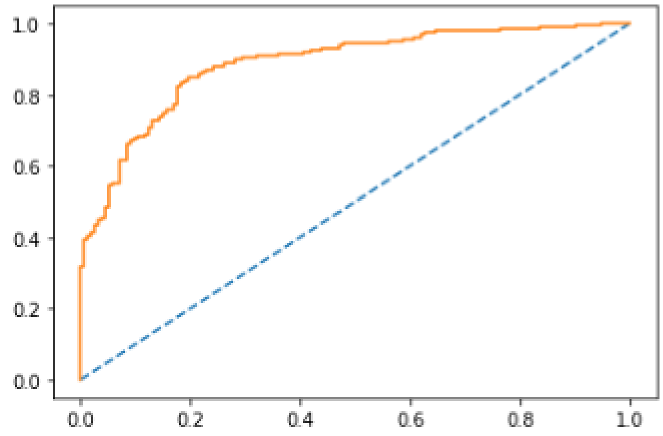
Naïve Bayes Model: Test Dataset



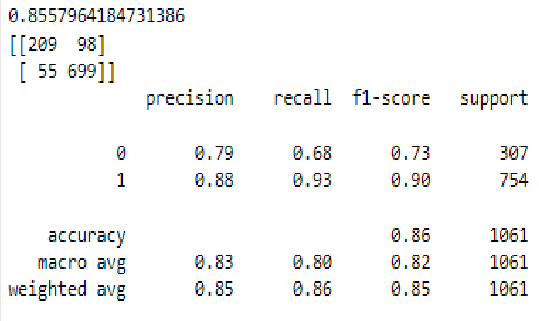
```
0.8223684210526315
[[112  41]
 [ 40 263]]
              precision    recall  f1-score   support

           0       0.74      0.73      0.73       153
           1       0.87      0.87      0.87       303

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



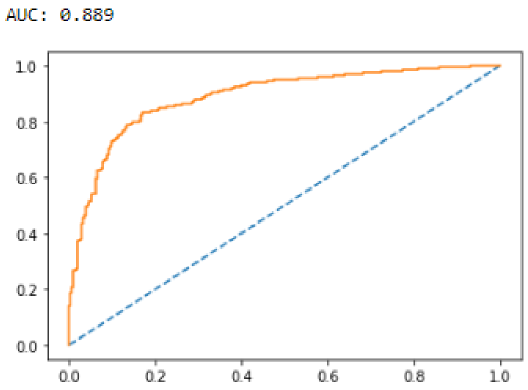
```
AUC: 0.888
```



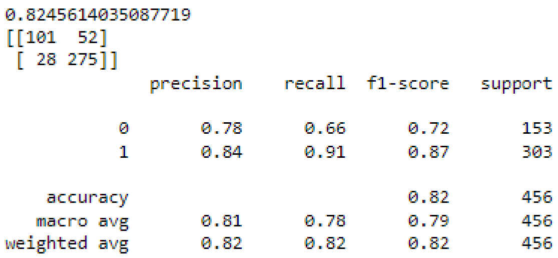
KNN Model: Train Set



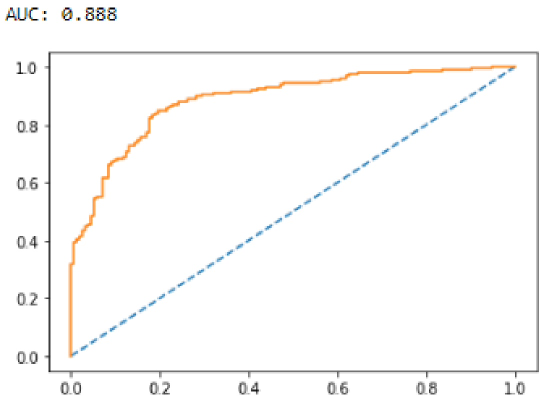
```
0.8557964184731386
[[209  98]
 [ 55 699]]
              precision    recall  f1-score   support

           0       0.79      0.68      0.73       307
           1       0.88      0.93      0.90       754

    accuracy                           0.86      1061
   macro avg       0.83      0.80      0.82      1061
weighted avg       0.85      0.86      0.85      1061
```



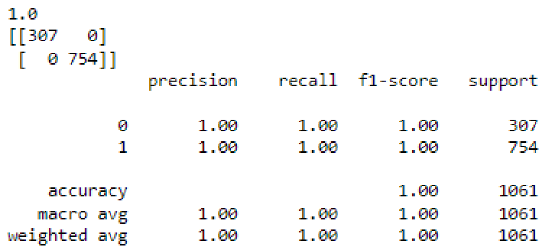
KNN Model: Test Set



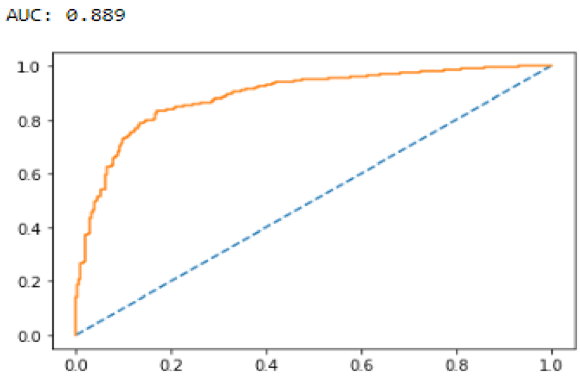
```
0.8245614035087719
[[101  52]
 [ 28 275]]
              precision    recall  f1-score   support

           0       0.78      0.66      0.72       153
           1       0.84      0.91      0.87       303

    accuracy                           0.82       456
   macro avg       0.81      0.78      0.79       456
weighted avg       0.82      0.82      0.82       456
```

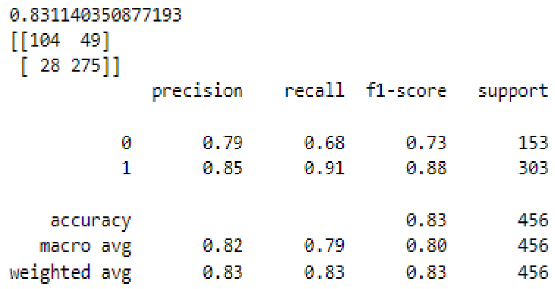


Random Forest without Grid Search CV: Train Set

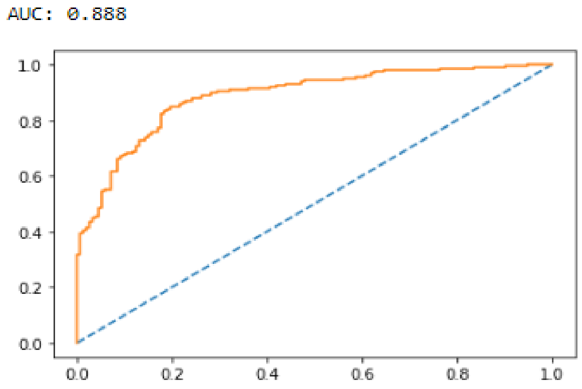


```
1.0
[[307   0]
 [  0 754]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       307
           1       1.00      1.00      1.00       754

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



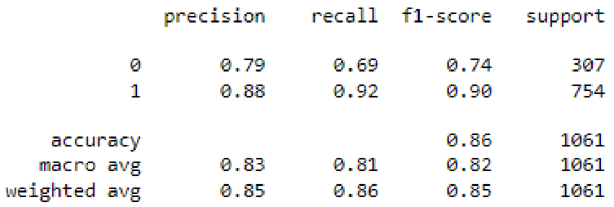


```
0.831140350877193
[[104  49]
 [ 28 275]]
              precision    recall  f1-score   support

           0       0.79      0.68      0.73       153
           1       0.85      0.91      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.79      0.80       456
weighted avg       0.83      0.83      0.83       456
```

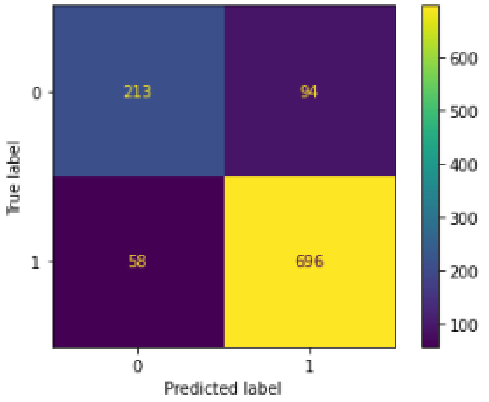


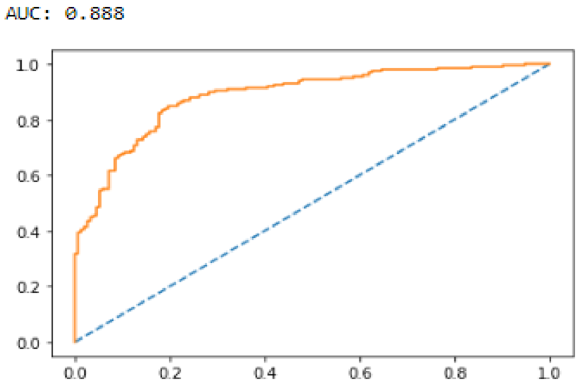
```
AUC: 0.888
```

Random Forest with Grid Search CV:



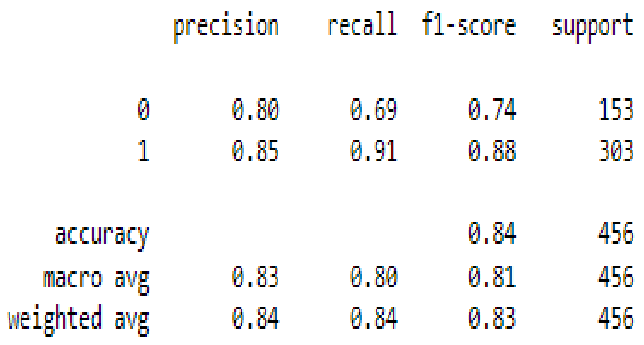
```
              precision    recall  f1-score   support

           0       0.79      0.69      0.74       307
           1       0.88      0.92      0.90       754

    accuracy                           0.86      1061
   macro avg       0.83      0.81      0.82      1061
weighted avg       0.85      0.86      0.85      1061
```



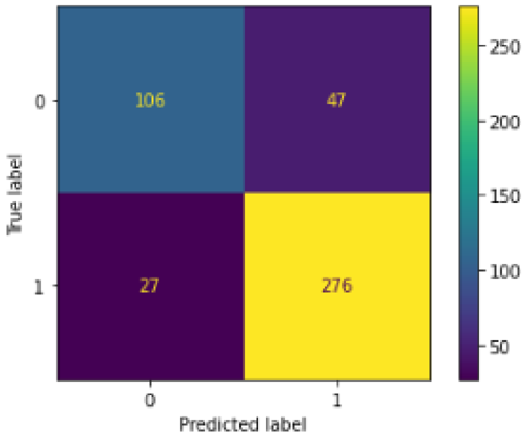


```
AUC: 0.888
```

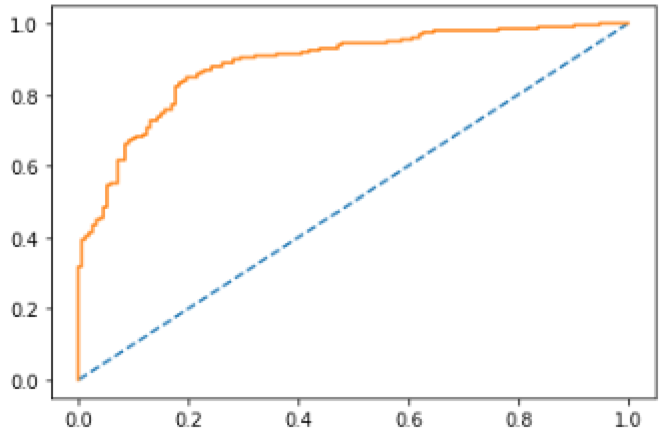
Random Forest with Grid Search CV: Test Dataset



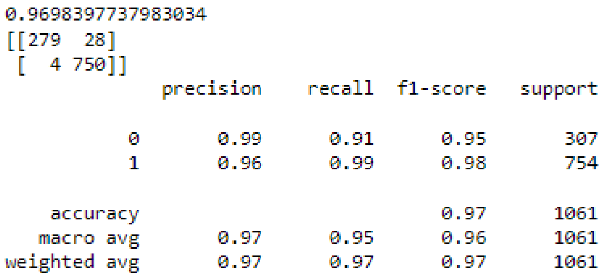
```
              precision    recall  f1-score   support

           0       0.80      0.69      0.74       153
           1       0.85      0.91      0.88       303

    accuracy                           0.84       456
   macro avg       0.83      0.80      0.81       456
weighted avg       0.84      0.84      0.83       456
```



```
AUC: 0.888
```

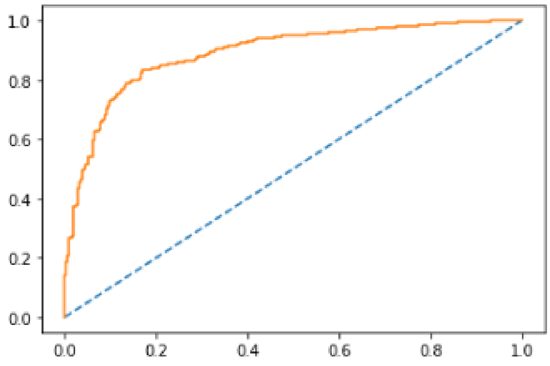


Bagging Classifier:Train Set

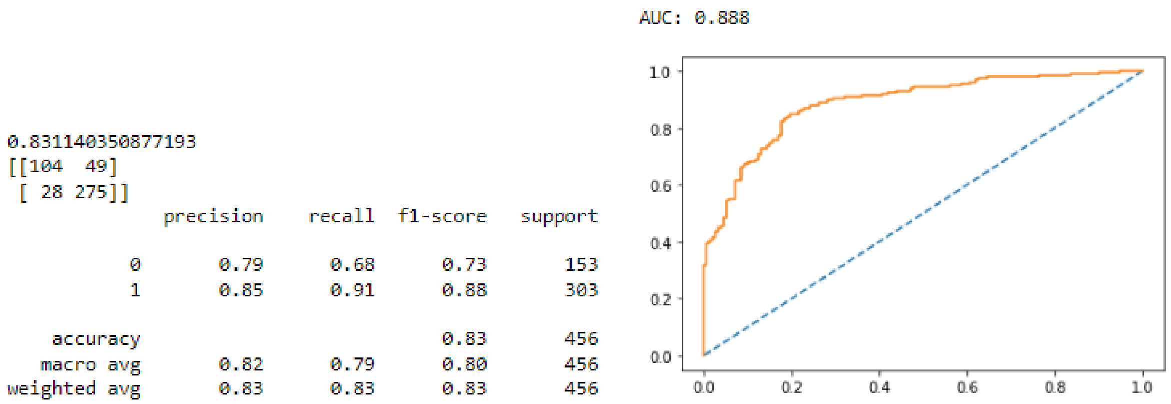


```
0.9698397737983034
[[279  28]
 [  4 750]]
              precision    recall  f1-score   support

           0       0.99      0.91      0.95       307
           1       0.96      0.99      0.98       754

    accuracy                           0.97      1061
   macro avg       0.97      0.95      0.96      1061
weighted avg       0.97      0.97      0.97      1061
```

```
AUC: 0.889
```

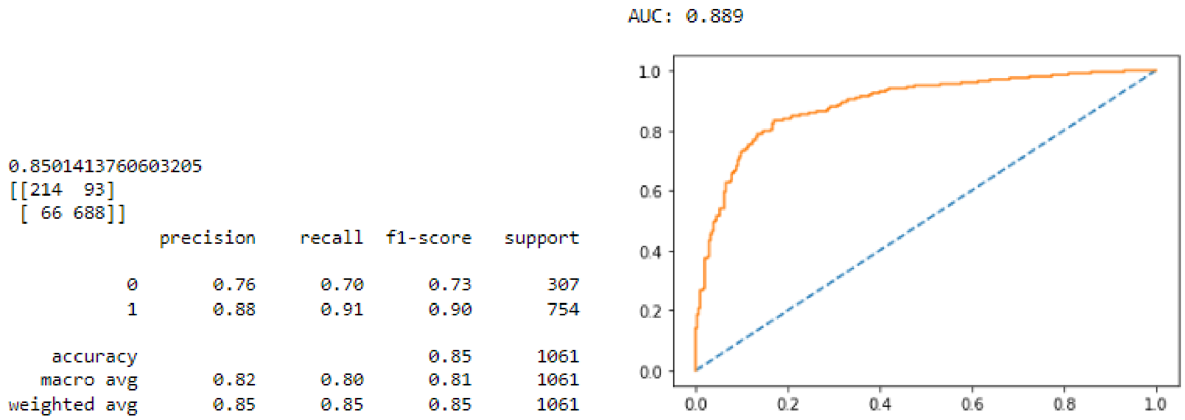


Bagging Classifier:Test Set



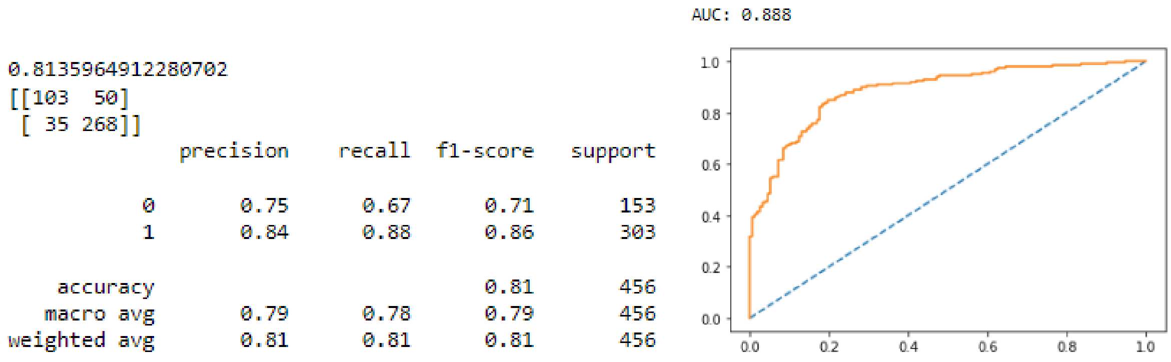
```
0.831140350877193
[[104  49]
 [ 28 275]]
              precision    recall  f1-score   support

           0       0.79      0.68      0.73       153
           1       0.85      0.91      0.88       303

    accuracy                           0.83       456
   macro avg       0.82      0.79      0.80       456
weighted avg       0.83      0.83      0.83       456
```

Ada Boost : Train Set



```
0.8501413760603205
[[214  93]
 [ 66 688]]
              precision    recall  f1-score   support

           0       0.76      0.70      0.73       307
           1       0.88      0.91      0.90       754

    accuracy                           0.85      1061
   macro avg       0.82      0.80      0.81      1061
weighted avg       0.85      0.85      0.85      1061
```

Ada Boost : Test Se



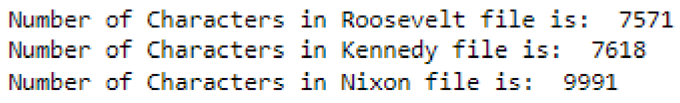
```
0.8135964912280702
[[103  50]
 [ 35 268]]
              precision    recall  f1-score   support

           0       0.75      0.67      0.71       153
           1       0.84      0.88      0.86       303

    accuracy                           0.81       456
   macro avg       0.79      0.78      0.79       456
weighted avg       0.81      0.81      0.81       456
```

Based on the accuracy measure, we can suggest that all most all the model performance is showing the accuracy in a range of 80-86% for all test and train based. Apart form this random forest without Grid Serach and Bagging classifier shows a 95-99% accuracy on train dataset and on test in range of 83-85%. These model shows the promising result.

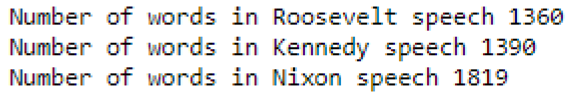
1.8) Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective.

1 As per above anaylsis, ensemble random forest model is performing better compared to other all model for voting part between labour and conservative class.
2 Economic condition houseware and blair is closely related in relation so both are effectively.
3 Hague and Blair are import variables in predicting which party as person votes
4 People with zero political knowledge mostly vote for Labour party.

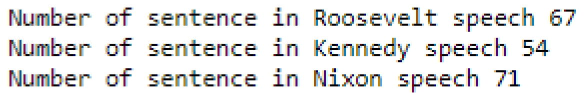
2.1) Find the number of characters, words and sentences for the mentioned documents. (Hint: use .words(), .raw(), .sent() for extracting counts)



```
Number of Characters in Roosevelt file is:  7571
Number of Characters in Kennedy file is:  7618
Number of Characters in Nixon file is:  9991
```

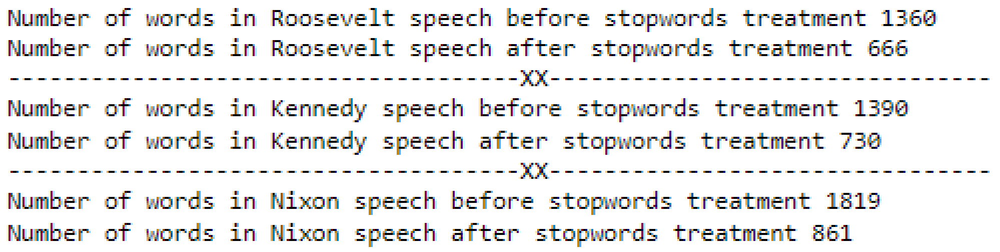


```
Number of words in Roosevelt speech 1360
Number of words in Kennedy speech 1390
Number of words in Nixon speech 1819
```



```
Number of sentence in Roosevelt speech 67
Number of sentence in Kennedy speech 54
Number of sentence in Nixon speech 71
```

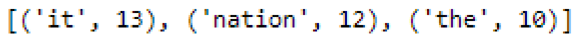
2.2) Remove all the stopwords from the three speeches. Show the word count before and after the removal of stopwords. Show a sample sentence after the removal of stopwords.



```
Number of words in Roosevelt speech before stopwords treatment 1360
Number of words in Roosevelt speech after stopwords treatment 666
------------------------------------XX-------------------------------
Number of words in Kennedy speech before stopwords treatment 1390
Number of words in Kennedy speech after stopwords treatment 730
------------------------------------XX-------------------------------
Number of words in Nixon speech before stopwords treatment 1819
Number of words in Nixon speech after stopwords treatment 861
```

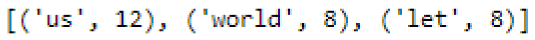
2.3) Which word occurs the most number of times in his inaugural address for each president? Mention the top three words. (after removing the stopwords)

In Roosevlet:



```
[('it', 13), ('nation', 12), ('the', 10)]
```

In Kenndy:



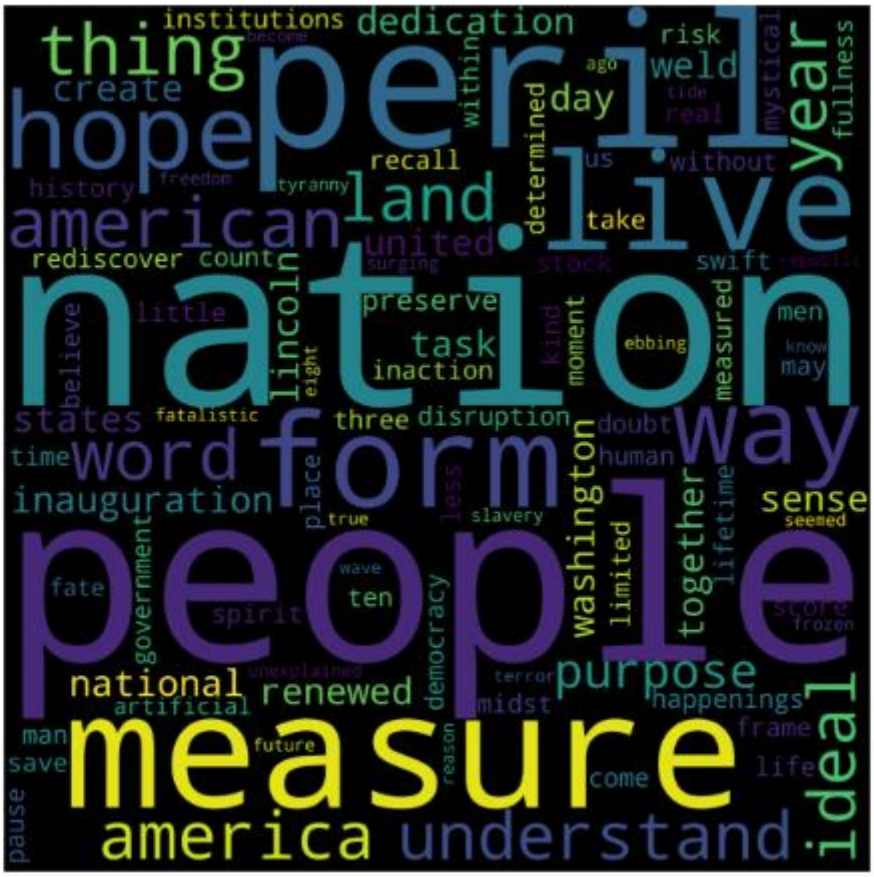
```
[('us', 12), ('world', 8), ('let', 8)]
```

In Nixon:

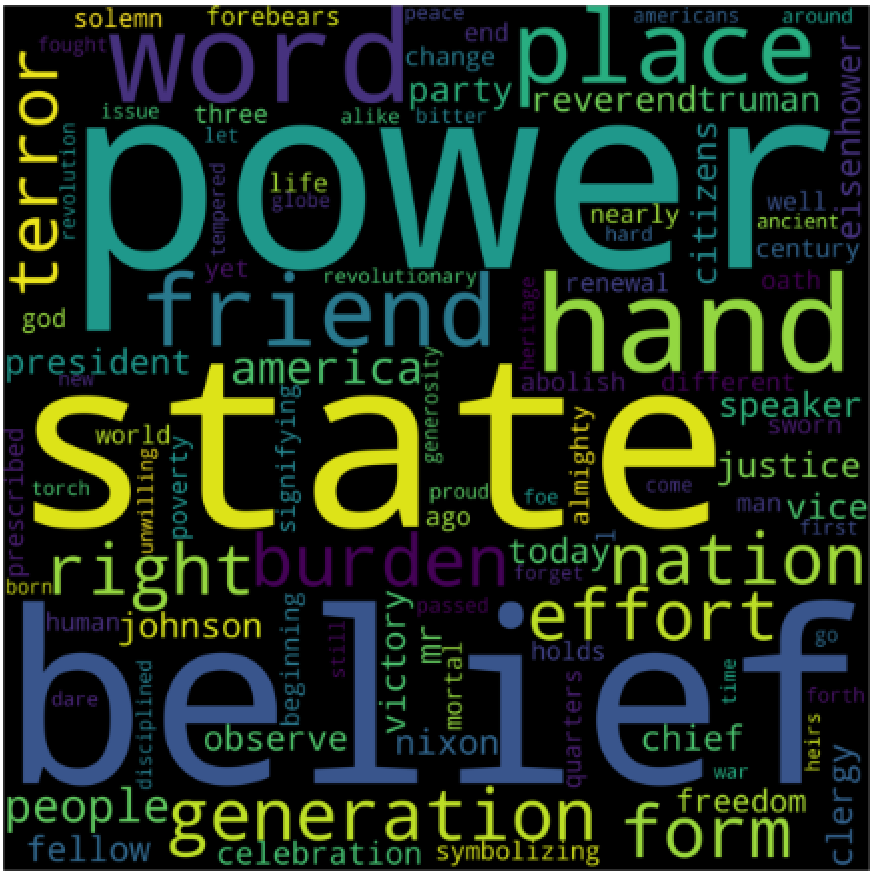
```
[('us', 26), ('america', 21), ('peace', 19)]
```

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

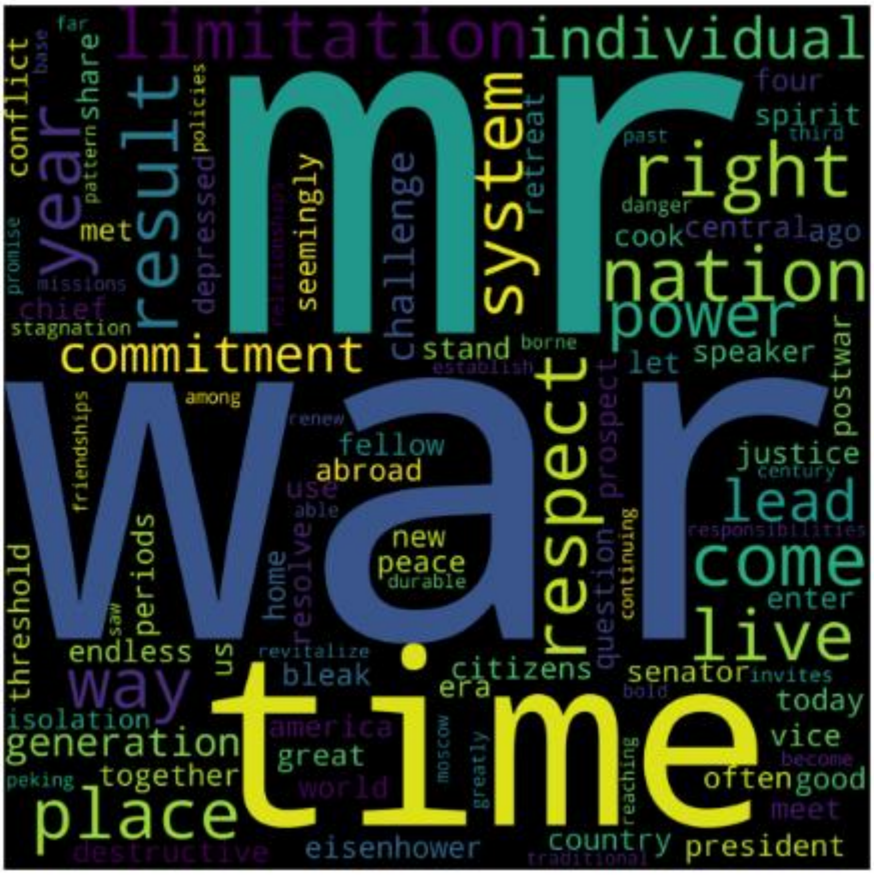
```
Word Cloud for Roosevelt (after cleaning)!!
```



```
Word Cloud for Kennedy (after cleaning)!!
```



Word Cloud for Nixon (after cleaning)!!


war
mr
time
limitation
individual
four
spirit
third
past
right
danger
cook
central
ago
nation
power
speaker
postwar
justice
century
lead
responsibilities
come
enter
live
senator
invites
bold
today
vice
become
often
good
meet
president
traditional
country
eisenhower
destructive
place
together
peking
generation
isolation
way
threshold
endless
periods
saw
friendships
among
stagnation
chief
missions
promise
met
year
conflict
share
pattern
base
far
policies
depressed
result
relationships
seemingly
challenge
system
retreat
commitment
stand
borne
establish
let
prospect
question
continuing
respect
renew
fellow
abroad
use
able
new
peace
durable
resolve
home
us
revitalize
bleak
citizens
era
america
great
world
moscow
greatly
reaching