Problem No.1

1.1 Let us import the dataset using head function and understand the problem.

	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

We can see that Unnamed variable is of no use at this moment, let us drop and read the dataset

	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

Let us understand the number of rows and columns in the dataset. We can see that there are rows and columns.

```
(1525, 9)
```

Let us check for any null values in the dataset

```
Unnamed: 0                  0
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
```

There are no null values present and also check for any duplicates. Let us also drop the duplicate values in the dataset. There are 8 duplicate values present in the dataset which are of no use .

Let us check the basic info about the dataset and also the statistical summary.

Only two variables are there with string values ,rest every variables are in numerical .

```
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   vote                    1517 non-null   object
 1   age                     1517 non-null   int64
```

```
 2   economic.cond.national   1517 non-null   int64
 3   economic.cond.household  1517 non-null   int64
 4   Blair                    1517 non-null   int64
 5   Hague                    1517 non-null   int64
 6   Europe                   1517 non-null   int64
 7   political.knowledge      1517 non-null   int64
 8   gender                   1517 non-null   object
dtypes: int64(7), object(2)
memory usage: 118.5+ KB
```

Every variables has no missing or non numerical values present.Almost every variables mean and median are equal. Let us check the normal distribution of observation using a histogram.

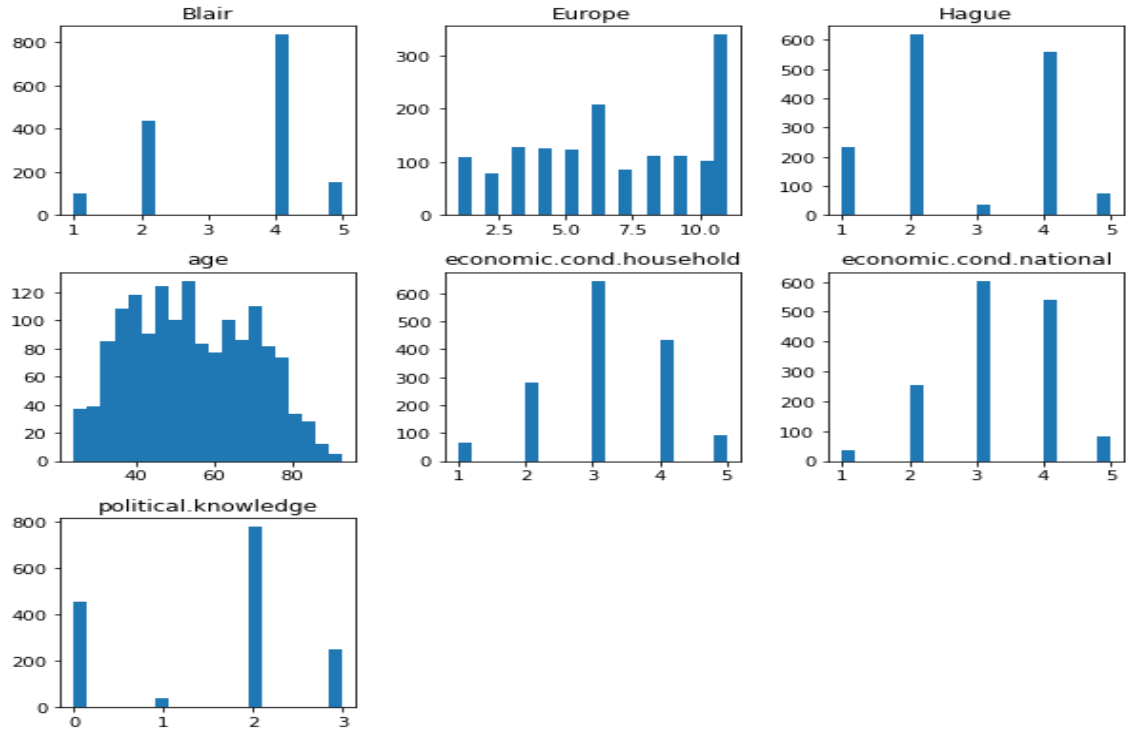
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
vote	1517	2	Labour	1057	NaN	NaN	NaN	NaN	NaN	NaN	NaN
age	1517	NaN	NaN	NaN	54.2413	15.7017	24	41	53	67	93
economic.cond.national	1517	NaN	NaN	NaN	3.24522	0.881792	1	3	3	4	5
economic.cond.household	1517	NaN	NaN	NaN	3.13777	0.931069	1	3	3	4	5
Blair	1517	NaN	NaN	NaN	3.33553	1.17477	1	2	4	4	5
Hague	1517	NaN	NaN	NaN	2.74951	1.23248	1	2	2	4	5
Europe	1517	NaN	NaN	NaN	6.74028	3.29904	1	4	6	10	11
political.knowledge	1517	NaN	NaN	NaN	1.54054	1.08442	0	0	2	2	3
gender	1517	2	female	808	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Let us check for the skewness in the dataset to understand the distribution.

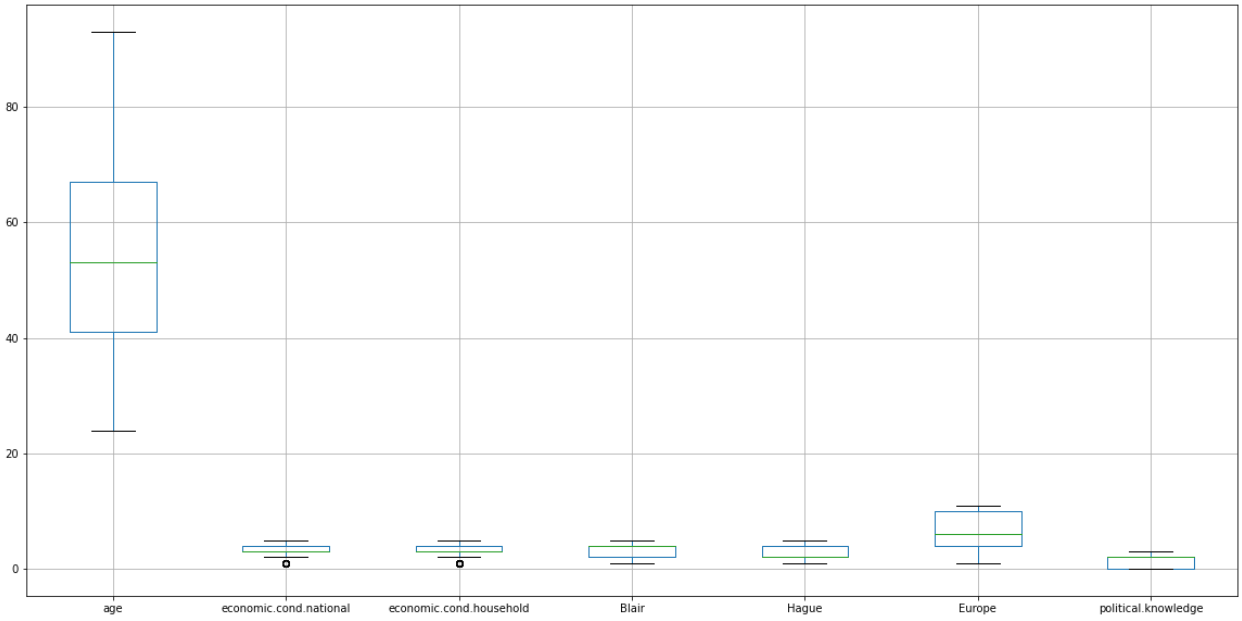
```
age                      0.144621
economic.cond.national  -0.240453
economic.cond.household -0.149552
Blair                   -0.535419
Hague                    0.152100
Europe                  -0.135947
political.knowledge     -0.426838
```

We can see that almost every observation is good to go ahead with the modelling. We have did the basic EDA part. Except age most of the variables has less standard deviation from the mean value. Let us

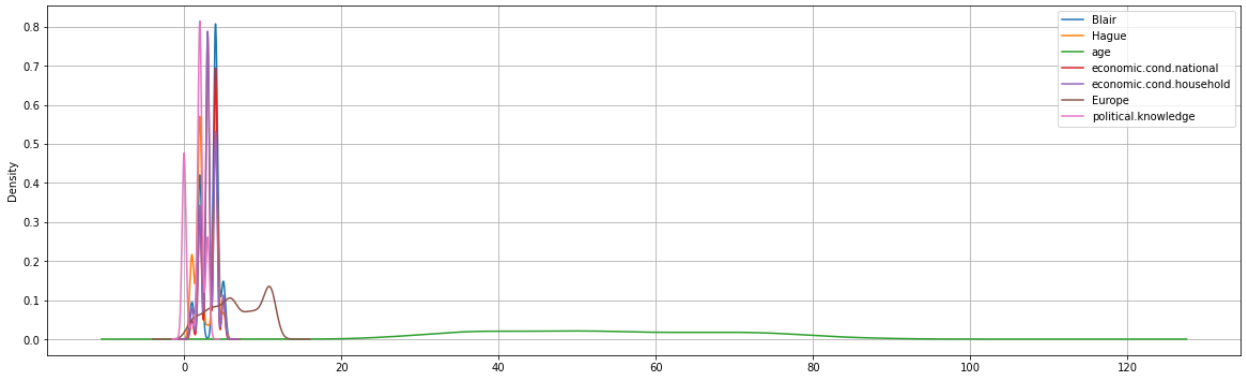
1.2



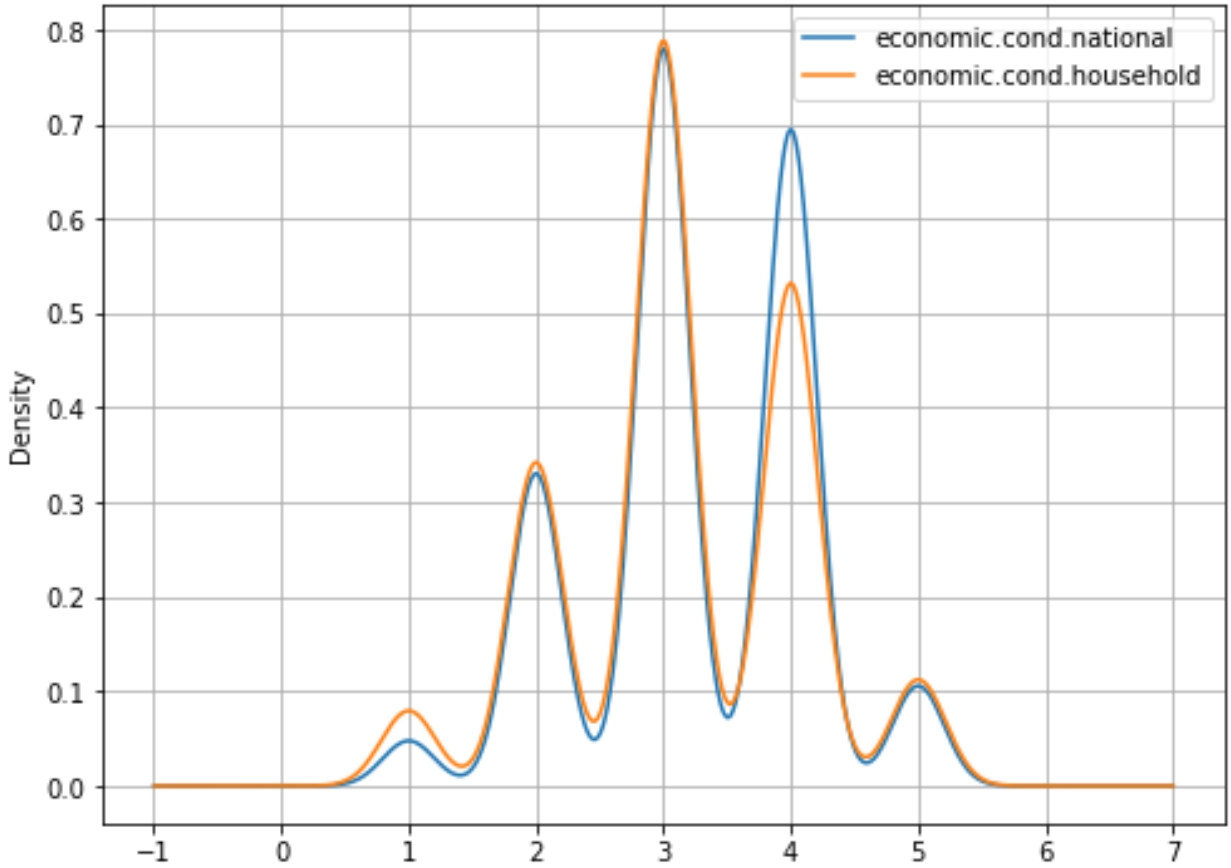
Almost every variable are in the zero skewness side. Let us check for the outliers using boxplot .



Only two variables has outlier. Rest of the variables doesn't have any outliers.Let us remove the outlier to treat the model.Also let us check for the distribution of the variables.

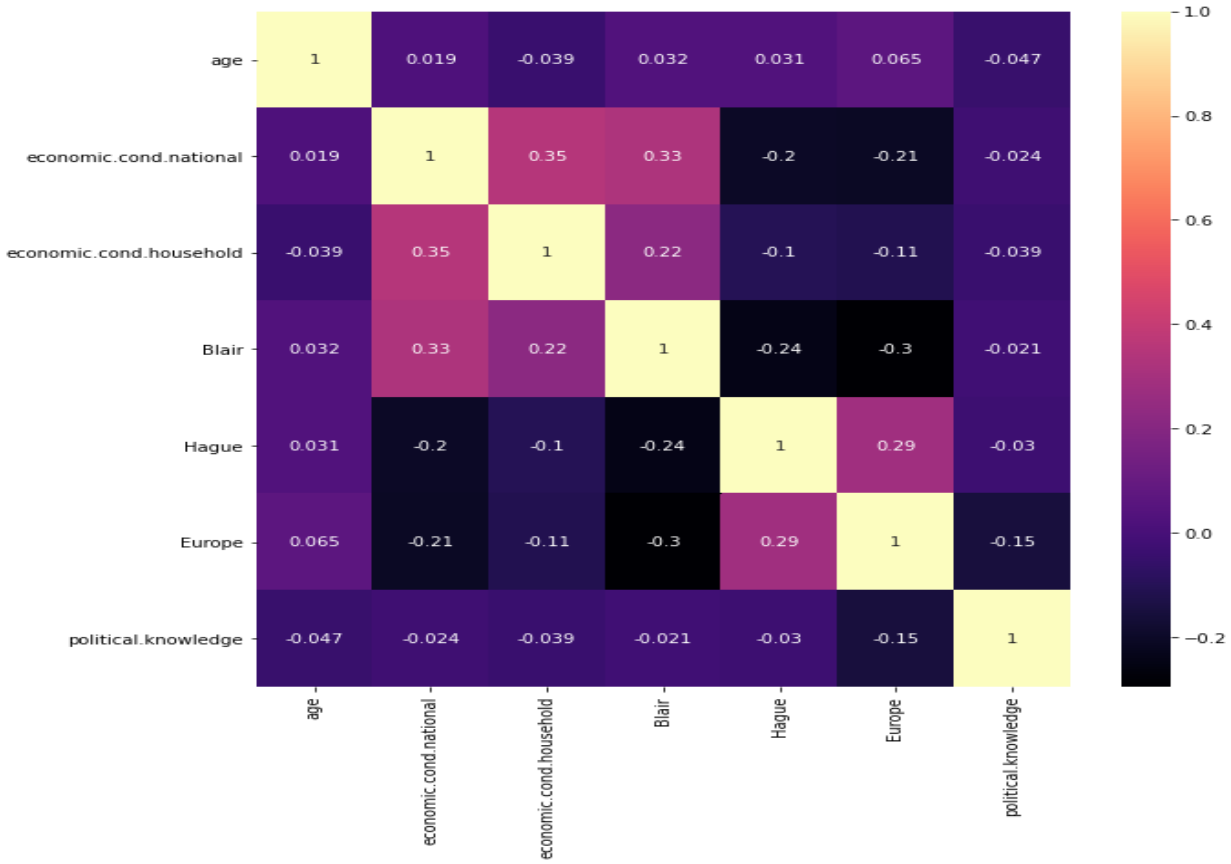


Let us look onto the distribution of variables with outliers .Two variables are almost having the same distribution curve.

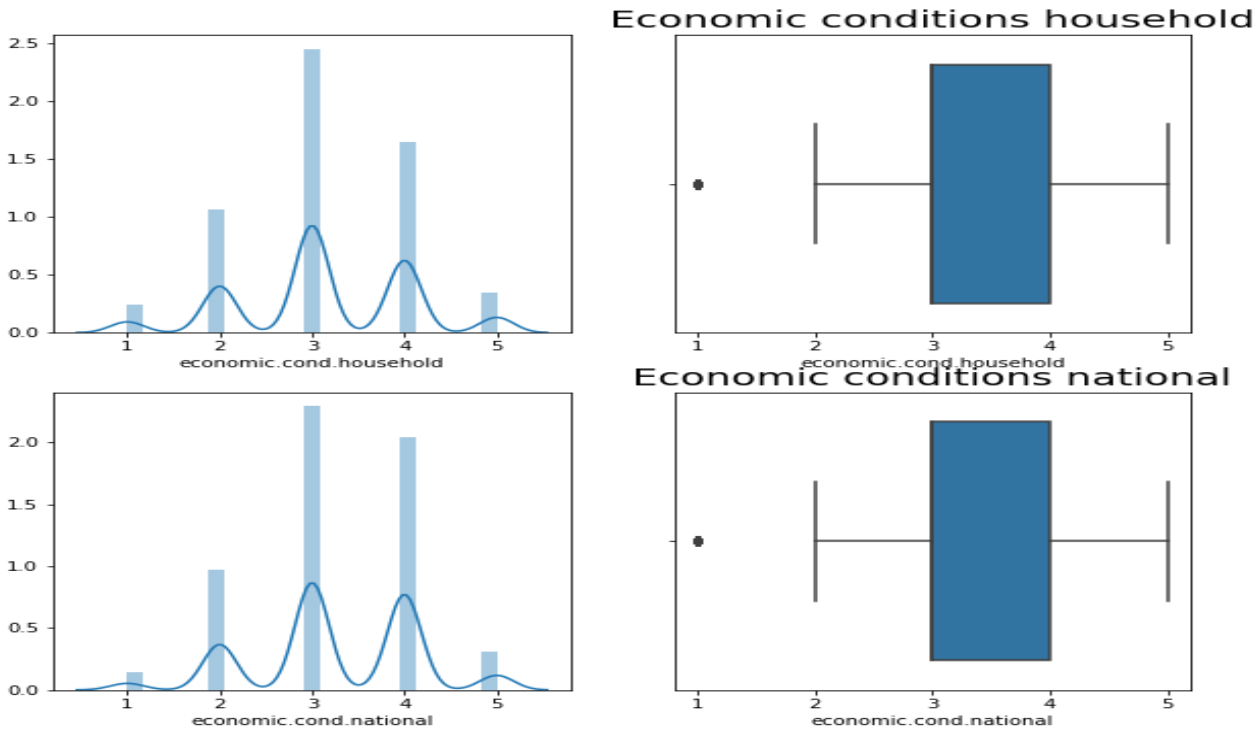


Let us check for the correlation of the variables . Using heatmap we can find out the correlation of variables.

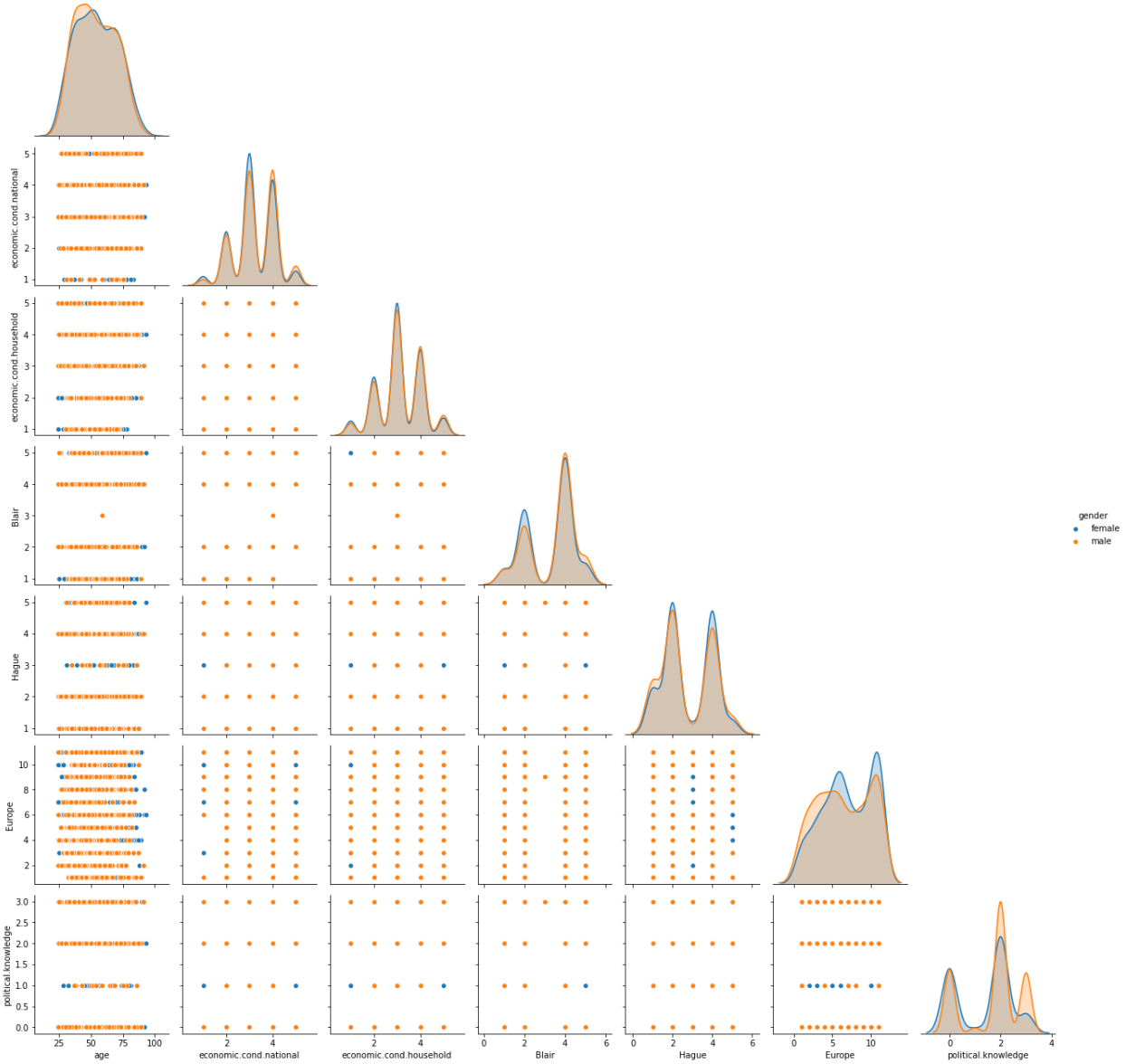
	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
age	1.000000	0.018687	-0.038868	0.032084	0.031144	0.064562	-0.046598
economic.cond.national	0.018687	1.000000	0.347687	0.326141	-0.200790	-0.209150	-0.023510
economic.cond.household	-0.038868	0.347687	1.000000	0.215822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



We can see that most of the variables are negatively correlated and there is less chance of multicollinearity problem in the variables. Let us clearly look onto the distribution of variables with ouliers in detail by plotting histogram and boxplot simultaneously.



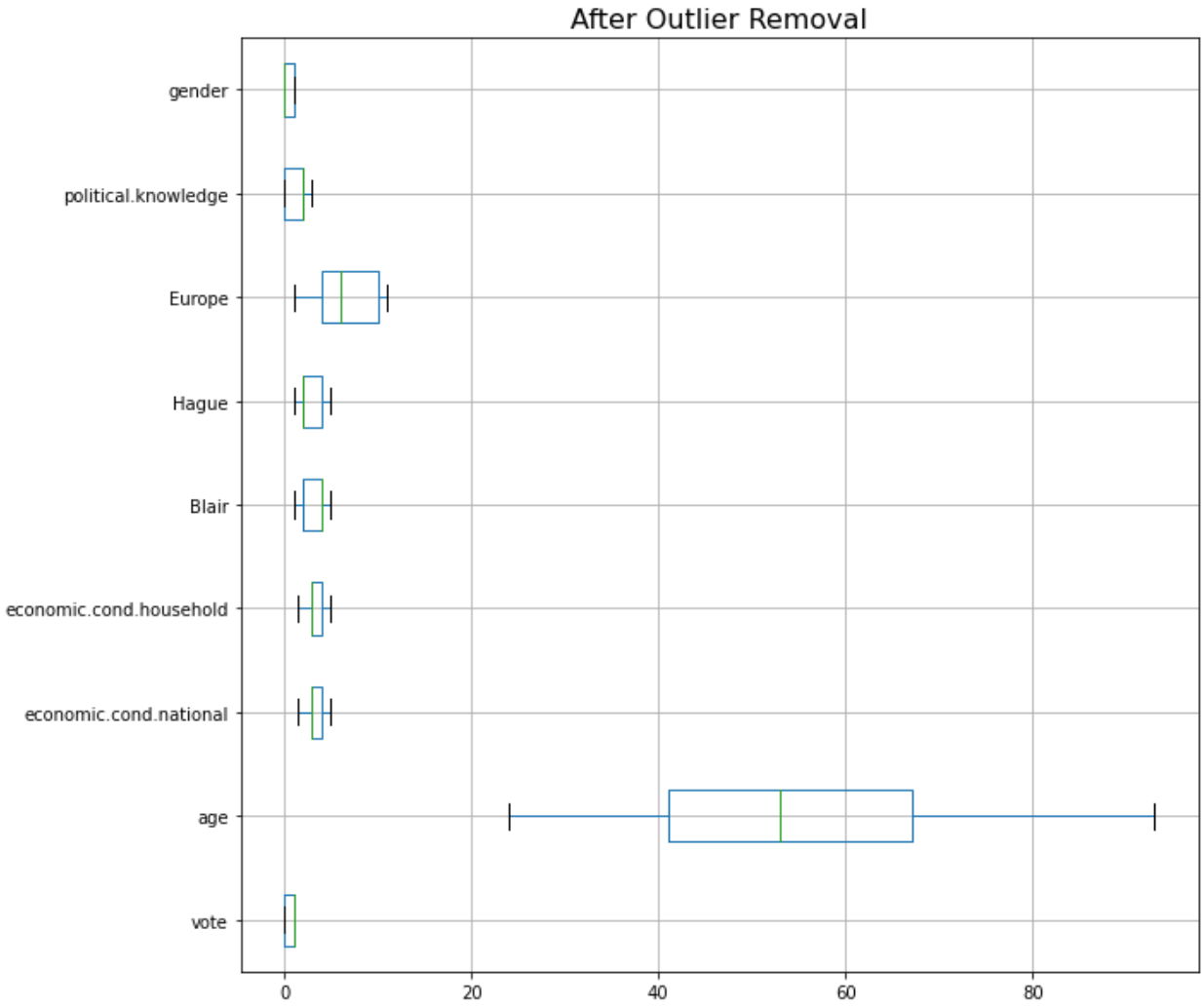
We will remove this outlier on later part. Now let us segregate the all the variables using their gender distribution .



Orange represent the male distribution and blue represent the female distribution. We can see that most of the observations are from male gender.Let us check for any imbalance problem in the dataset. Let us use the vote variable to find out the number of labour and conservative counts in the dataset.

```
Labour          0.69677
Conservative    0.30323
Name: vote, dtype: float64
```

Now lets remove the outlier from the dataset by performing **IQR method. Below figure shows the boxplot of variable after outlier treatment.**



1.3 For this problem statement, vote variable is our target variable and we have to find out the performance metrics for both labour and conservative parties. We can see that most of the people have voted for labour parties and 30% only have voted for conservative. Let us predict the voters using this target variables. For predicting the model we have to split the dataset into training and testing dataset. Mostly importantly , we have to convert the categorical variable into one hot encoding .Let us also convert the object datatypes into numerical values for modelling.

```
VOTE : 2
0     460
1    1057
Name: vote, dtype: int64


GENDER : 2
1    709
0    808
Name: gender, dtype: int64
```

We can see that vote has two codes 0 represent conservative and 1 represent labour votes. GENDER variable is also assigned with respective codes 0s and 1s.Let us check once again about the basic info of dataset.We can see that all variables are converted to numerical values . Now our dataset is good to split into training and testing.

```
#   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   vote                      1517 non-null   float64
 1   age                       1517 non-null   float64
 2   economic.cond.national    1517 non-null   float64
 3   economic.cond.household   1517 non-null   float64
 4   Blair                     1517 non-null   float64
 5   Hague                     1517 non-null   float64
 6   Europe                    1517 non-null   float64
 7   political.knowledge       1517 non-null   float64
 8   gender                    1517 non-null   float64
dtypes: float64(9)
memory usage: 158.5 KB
```

	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	1.0	43.0	3.0	3.0	4.0	1.0	2.0	2.0	0.0
1	1.0	36.0	4.0	4.0	4.0	4.0	5.0	2.0	1.0
2	1.0	35.0	4.0	4.0	5.0	2.0	3.0	2.0	1.0
3	1.0	24.0	4.0	2.0	2.0	1.0	4.0	0.0	0.0
4	1.0	41.0	2.0	2.0	1.0	1.0	6.0	2.0	1.0

Scaling is necessary only for model which is based on distance rule. In this problem , **let us perform scaling only for KNN modelling. LDA and logistics and naives bayes doesn't have any effects by scaling.**

Let us split the dataset into 70% training and 30% test size. Here vote is the target variables and rest are the independent variables.

```
age                         68
economic.cond.national       5
economic.cond.household      5
Blair                        5
Hague                        5
Europe                      11
political.knowledge          4
gender                       2
dtype: int64
```

Above represent the amount of training datset observation.Below represent for testing with same.

```
age                         66
economic.cond.national       5
economic.cond.household      5
Blair                        4
Hague                        5
Europe                      11
political.knowledge          4
gender                       2
```

The size of training and testing datasets are :-

```
(1061, 8)
(456, 8)
```

1.4After importing the logistic regression from sklearn_model library . Let us call the function into a variable called model with a limit of max iteration of 1000. Let us call all the model from the library and assign to a new variable :-

```
Model=LogisticRegression(max_iter=1000)
Model1=LinearDiscriminantAnalysis()
Model2=GaussianNB()
```

Let us call the KNN model later part after scaling. After calling the model function from the library , let us fit the training dataset into the respective model variables and check for the model accuracy for both training and testing dataset.

```
The logistic reg training model accuracy : 0.8454288407163054
The logistic reg testing model accuracy : 0.8223684210526315


The LDA training accuracy : 0.8397737983034873
The LDA testing accuracy: 0.8223684210526315
```

From the above model accuracies , we can see that logistics regression looks better compares to linear discriminant analysis model with a small difference in the training accuracies. Both the model hold same testing accuracy.Both Logistic regression and LDA doesn't requires scaling because they are not affected by scaled values.

1.5 Let us import naives bayes and KNN model from sklearn libaray. Specially for knn model, let us scale the dataset and split the dataset into training and testing .While for naïve bayes ,we wont scale the data. Then lets us fit the training dataset into the model and check for the prediction.

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	-0.716161	-0.301648	-0.179682	0.565802	-1.419969	-1.437338	0.423832	-0.936736
1	-1.162118	0.870183	0.949003	0.565802	1.014951	-0.527684	0.423832	1.067536
2	-1.225827	0.870183	0.949003	1.417312	-0.608329	-1.134120	0.423832	1.067536
3	-1.926617	0.870183	-1.308366	-1.137217	-1.419969	-0.830902	-1.421084	-0.936736
4	-0.843577	-1.473479	-1.308366	-1.988727	-1.419969	-0.224465	0.423832	1.067536
...	...	...	...	...	...	...	...	...
1512	0.812836	2.042014	-0.179682	-1.137217	1.014951	1.291625	1.346290	1.067536
1513	1.195085	-1.473479	-1.308366	0.565802	1.014951	0.381971	0.423832	1.067536
1514	-1.098410	-0.301648	-0.179682	1.417312	1.014951	-1.437338	0.423832	1.067536
1515	0.430587	-0.301648	-0.179682	-1.988727	1.014951	1.291625	0.423832	1.067536
1516	1.258794	-1.473479	-0.179682	-1.137217	1.014951	1.291625	-1.421084	-0.936736

```
The Naive Bayes training accuracy:  0.8435438265786993
```

```
The Naive Bayes testing accuracy: 0.8245614035087719


The KNN training accuracy : 0.8708765315739868
The KNN testing accuracy: 0.8223684210526315
```

From the above model accuracies , KNN model have more accuracy on training dataset compares to all other model. All model have overfit problems. Let us consider the modes with which less than 10% difference in training and testing as a standard model. So, Let us perform the prediction with all models and check their precision, recall, f1-score and model accuracy. We will be considering the F1-score to compare the model efficiency.

1.6 Let us use GRIDSEARCHCV for hypertuning the models. Before applying the bagging classifier , let us use random forest and apply into bagging base estimator.Let us apply gradiest boosting and adaboost classifier as boosting alogirthm . Naives baves doesn't hold with more number of parameters.So it is difficult to tune the naives baves alogorithm.We will be tuning the all the base models with different combination of parameters.

Logistic regression TUNED

After applying the gridsearchCv function, we got the best estimator from the model with all permutation and combination of parameters within models.

```
{'penalty': 'l2', 'random_state': 0, 'solver': 'newton-cg', 'tol': 0.0001}
```

LDA TUNED

After applying the different combination of parameters into the gridsearchcv function, we got as below the best parameter for LDA

```
{'solver': 'lsqr', 'tol': 0.0001}
```

KNN TUNED

With 9 combination of cross validation , we got below parameters as the best parameter for knn model.

```
{'algorithm': 'auto', 'leaf_size': 30, 'p': 1, 'weights': 'uniform'}
```

We will go with the base model parameter for naives bayes. Since naives bayes doesn't have more parameters to tune .

BAGGING

For BAGGING , lets us import randomforestclassifier from sklearnmodel library. Apply all the parameters and fit the model.Before that let us find out out of bag score error to identify the amount of error in the prediction.

```
0.824693685202639
0.824693685202639
0.820923656927427
0.8265786993402451
0.8369462770970783
0.827521206409048
0.8303487276154571
0.8341187558906692
0.8294062205466541
0.8294062205466541
0.8331762488218661
```

Above are the output of OOB score for respective random state, lets us choose random state as 5 to have a better prediction from the above OOB score.After applying the parameter into randomforeset model.Let us fit the training variables into the model and do the bagging operation. RFC used as the base estimator in the bagging classifer model.

```
RandomForestClassifier(n_estimators=501, random_state=5)
```

BOOSTING

We are using two boosting techniques to find out the best model .,Gradient boosting and adaboost classifier.We are calling the gradientboostingclassifier into a variable called

gbcl = GradientBoostingClassifier(n_estimators = 50,random_state=1)

and calling adaboostingclassfier into a variable with parameters :-

abcl = AdaBoostClassifier(n_estimators=10, random_state=1)

1.7 Performance Metrics

Let us evaluate the training and testing accuracy , precision, recall, f1-score respectively.

Logistic regression – Training

```
[[218 103]
 [ 61 679]]
```

	precision	recall	f1-score	support
0.0	0.78	0.68	0.73	321
1.0	0.87	0.92	0.89	740
accuracy			0.85	1061
macro avg	0.82	0.80	0.81	1061
weighted avg	0.84	0.85	0.84	1061

logistic regression (Tuned)- training

```
array([[218, 103],
```

```
       [ 61, 679]], dtype=int64)
              precision    recall  f1-score   support

         0.0       0.78      0.68      0.73       321
         1.0       0.87      0.92      0.89       740

    accuracy                           0.85      1061
   macro avg       0.82      0.80      0.81      1061
weighted avg       0.84      0.85      0.84      1061
```

Logistic regression – testing

```
[[ 92  47]
 [ 34 283]]


              precision    recall  f1-score   support

         0.0       0.73      0.66      0.69       139
         1.0       0.86      0.89      0.87       317

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

Logistic regression (Tuned)- testing

```
array([[ 92,  47],
       [ 34, 283]], dtype=int64)


              precision    recall  f1-score   support

         0.0       0.73      0.66      0.69       139
         1.0       0.86      0.89      0.87       317

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

There is no much difference in the tuned and base model for logistic regression. We have received 87% of f1-score on labour votes and 69% of better prediction for conservative voters.

LINEAR DISCRIMINANT ANALYSIS

LDA TRAINING -

```
[[221 100]
 [ 70 670]]
```

	precision	recall	f1-score	support
0.0	0.76	0.69	0.72	321
1.0	0.87	0.91	0.89	740
accuracy			0.84	1061
macro avg	0.81	0.80	0.80	1061
weighted avg	0.84	0.84	0.84	1061

LDA TRANING TUNED-

```
array([[221, 100],
       [ 70, 670]], dtype=int64)
```

	precision	recall	f1-score	support
0.0	0.76	0.69	0.72	321
1.0	0.87	0.91	0.89	740
accuracy			0.84	1061
macro avg	0.81	0.80	0.80	1061
weighted avg	0.84	0.84	0.84	1061

LDA TESTING

```
[[ 94  45]
 [ 36 281]]
```

	precision	recall	f1-score	support
0.0	0.72	0.68	0.70	139
1.0	0.86	0.89	0.87	317
accuracy			0.82	456
macro avg	0.79	0.78	0.79	456
weighted avg	0.82	0.82	0.82	456

LDA TESTING TUNED-

```
array([[ 94,  45],
       [ 36, 281]], dtype=int64)
```

	precision	recall	f1-score	support
0.0	0.72	0.68	0.70	139
1.0	0.86	0.89	0.87	317
accuracy			0.82	456
macro avg	0.79	0.78	0.79	456
weighted avg	0.82	0.82	0.82	456

The base model and tuned model are having no differences. We can see that 87% better prediction is for labour voters and 70% f1-score is conservative voters are obtained from LDA Models. Conservative voters are correctly and more accurately predicted in lda model comparing to logistics regression model.

NAÏVE BAYES MODEL

Naïve bayes model -training

```
[[233  88]
 [ 78 662]]
```

	precision	recall	f1-score	support
0.0	0.75	0.73	0.74	321
1.0	0.88	0.89	0.89	740
accuracy			0.84	1061
macro avg	0.82	0.81	0.81	1061
weighted avg	0.84	0.84	0.84	1061

Naïve bayes model -testing

```
[[ 94  45]
 [ 35 282]]
```

	precision	recall	f1-score	support
0.0	0.73	0.68	0.70	139
1.0	0.86	0.89	0.88	317
accuracy			0.82	456
macro avg	0.80	0.78	0.79	456
weighted avg	0.82	0.82	0.82	456

By observing the classification report of naïve bayes testing classification report , we can see that we have received a better f1-score on labour voters. There is a 1% increase in the prediction for labour voters with naïve bayes model while 70% of f1-score for conservative voters. As of now, this model looks good since there is only 2% difference with the testing and training model accuracies. We can see that naïve bayes is having less overfitting issue. To an extend , we have received a quiet good precision and recall rates. Let us check for knn model performance using the metrics analysis

KNN MODEL

KNN MODEL -TRAINING

```
0.8708765315739868


[[238  83]
 [ 54 686]]


              precision    recall  f1-score   support

         0.0       0.82      0.74      0.78       321
         1.0       0.89      0.93      0.91       740

    accuracy                           0.87      1061
   macro avg       0.85      0.83      0.84      1061
weighted avg       0.87      0.87      0.87      1061
```

KNN MODEL TUNED-TRAINING

```
array([[242,  79],
       [ 53, 687]], dtype=int64)


              precision    recall  f1-score   support

         0.0       0.82      0.75      0.79       321
         1.0       0.90      0.93      0.91       740

    accuracy                           0.88      1061
   macro avg       0.86      0.84      0.85      1061
weighted avg       0.87      0.88      0.87      1061
```

Once tuned, we can see that training model accuracy has been increased

KNN MODEL TESTING

```
0.8223684210526315


[[ 94  45]
 [ 36 281]]


              precision    recall  f1-score   support

         0.0       0.72      0.68      0.70       139
         1.0       0.86      0.89      0.87       317

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

KNN MODEL TUNED TESTING

```
array([[ 94,  45],
       [ 35, 282]], dtype=int64)
```

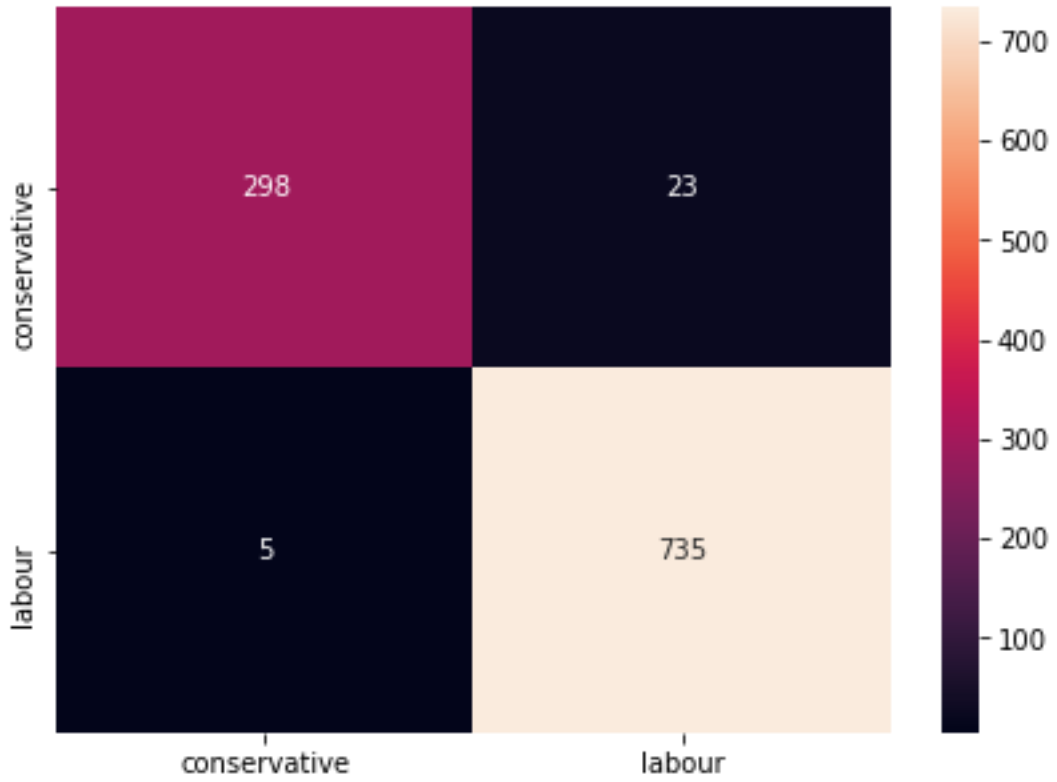
	precision	recall	f1-score	support
0.0	0.73	0.68	0.70	139
1.0	0.86	0.89	0.88	317
accuracy			0.82	456
macro avg	0.80	0.78	0.79	456
weighted avg	0.82	0.82	0.82	456

Once tuned, we can see that after tuning the f1-score for labour voters have increased by 1% on testing report. This model looks perfect since it has better f1-score ,precision ,recall and model accuracies comparing to rest of the models. Let us compare in detail using a tabular format. Before that , let us also check for bagging and boosting classfier models reports.

BAGGING

Before applying the bagging classifier, we have imported randomforest classifier and fitted the training dataset into it. We have to use the randomforestclassifier as the base estimator for the bagging classifer.After inputing the dataset into the training and evaluating with the testing dataset . We have received a confusion matrix like this :-

BAGGING TRAINING



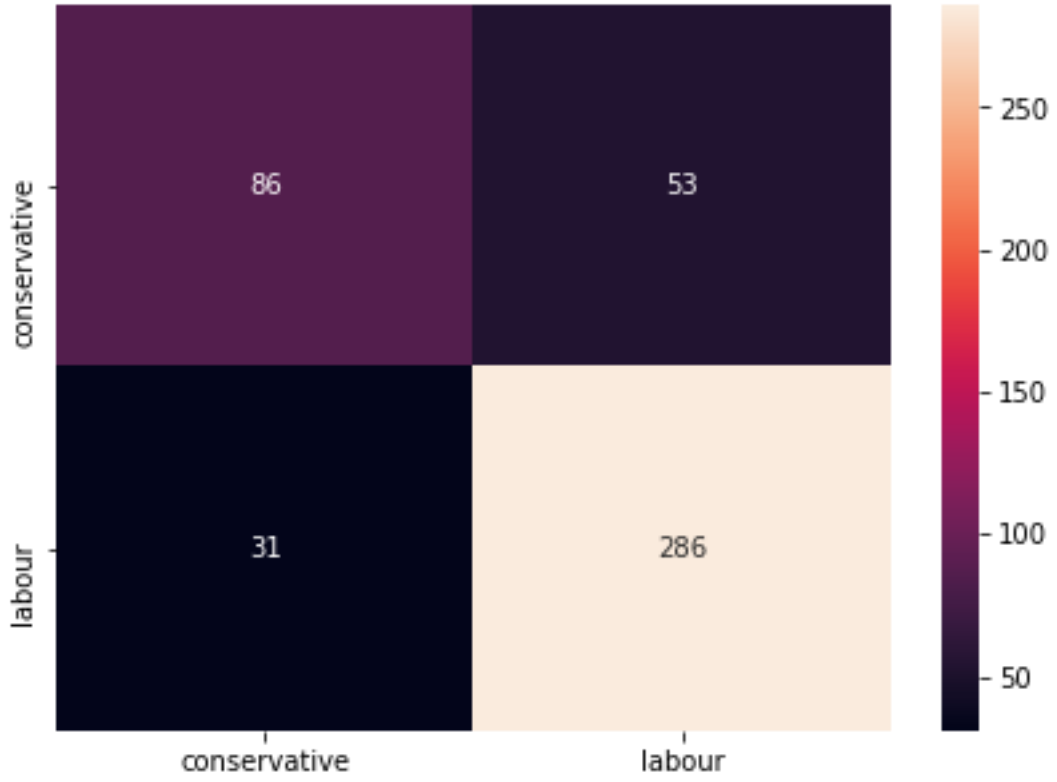
MODEL ACCURACY = 0.9736098020735156

	precision	recall	f1-score	support
0.0	0.98	0.93	0.96	321
1.0	0.97	0.99	0.98	740

```
    accuracy                           0.97      1061
   macro avg       0.98      0.96      0.97      1061
weighted avg       0.97      0.97      0.97      1061
```

BAGGING TESTING

```
0.8157894736842105
```



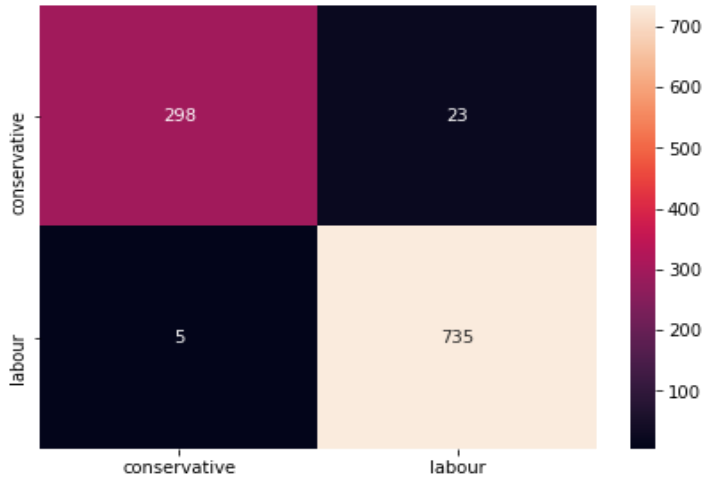

```
array([[ 86,  53],
       [ 31, 286]], dtype=int64)
```

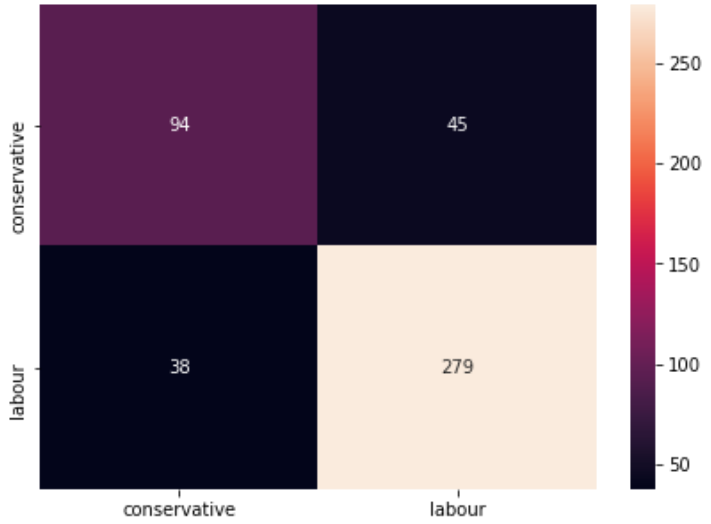
```
MODEL ACCURACY = 0.8157894736842105
```

```
              precision    recall  f1-score   support

         0.0       0.74      0.62      0.67       139
         1.0       0.84      0.90      0.87       317

    accuracy                           0.82       456
   macro avg       0.79      0.76      0.77       456
weighted avg       0.81      0.82      0.81       456
```

After applying the bagging classifier, we have received 97% of model accuracy with better f1-score . But once we check for the performance of testing model , we have only received 82% model accuracy. And also we can see that our f1-score for conservative and labour voters predicted has drop down drastically by comparing with other models.We can see there is a 15% difference in the training and testing model accuracies of bagging classifier.

BOOSTING





Training Model accuracy = 0.9736098020735156

	precision	recall	f1-score	support
0.0	0.98	0.93	0.96	321
1.0	0.97	0.99	0.98	740
accuracy			0.97	1061
macro avg	0.98	0.96	0.97	1061
weighted avg	0.97	0.97	0.97	1061

testing model accuracy= 0.8179824561403509

	precision	recall	f1-score	support
0.0	0.71	0.68	0.69	139
1.0	0.86	0.88	0.87	317
accuracy			0.82	456

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

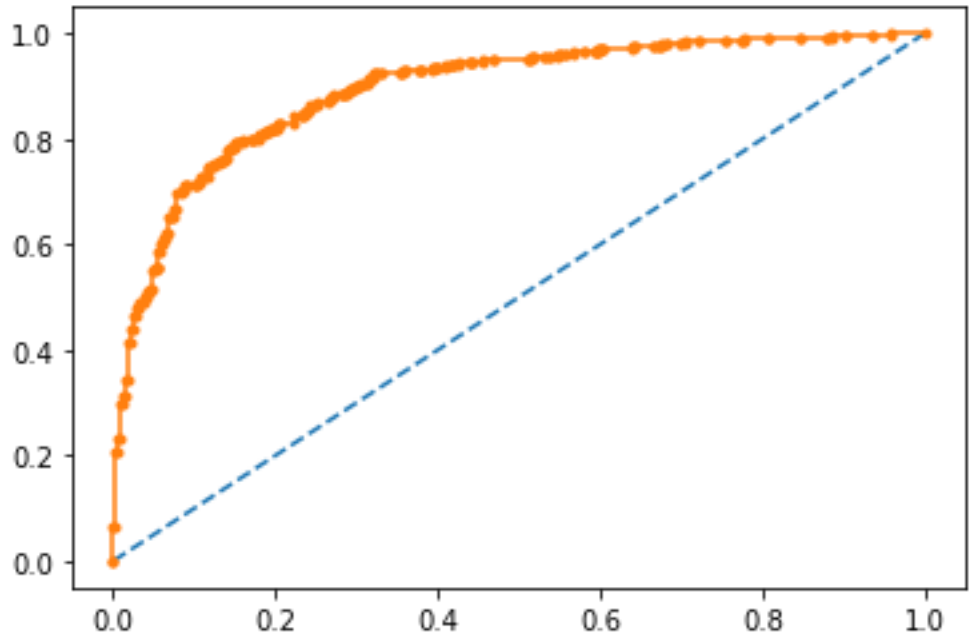
For boosting classifier , the training accuracy is more compares to testing accuracy. Here we can see both bagging and boosting models have overfitting problems. Both the model have same testing accuracy. Since both models have same performance to an extend. But we can notice that , boosting f1score for conservative voters have 2% high compares to bagging. So that we can conclude, boosting is better in this case compares to bagging.

ROC AND AUC CURVE

1.)LOGISTIC REGRESSION

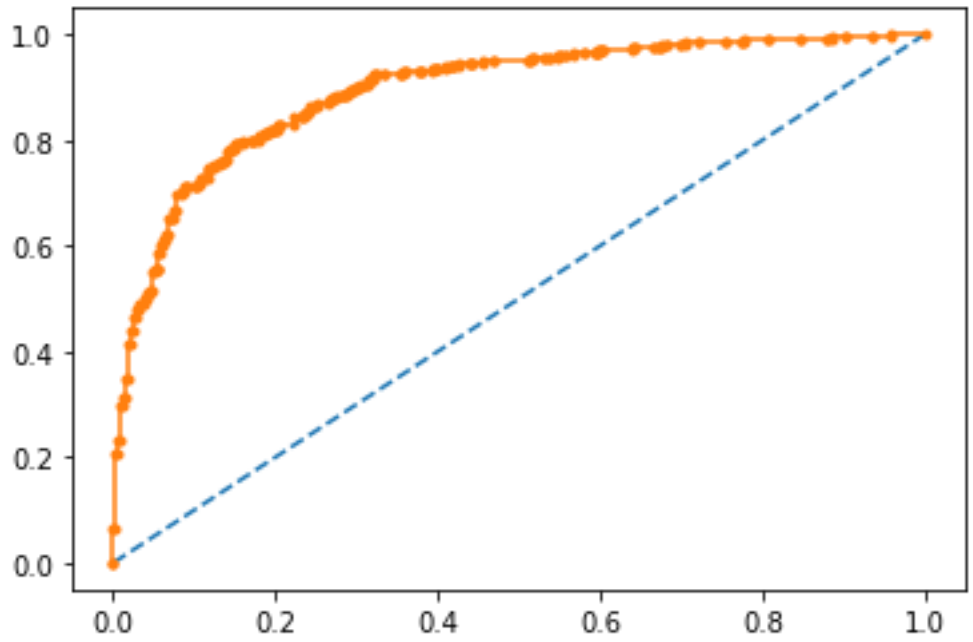
Training

AUC 0.895:



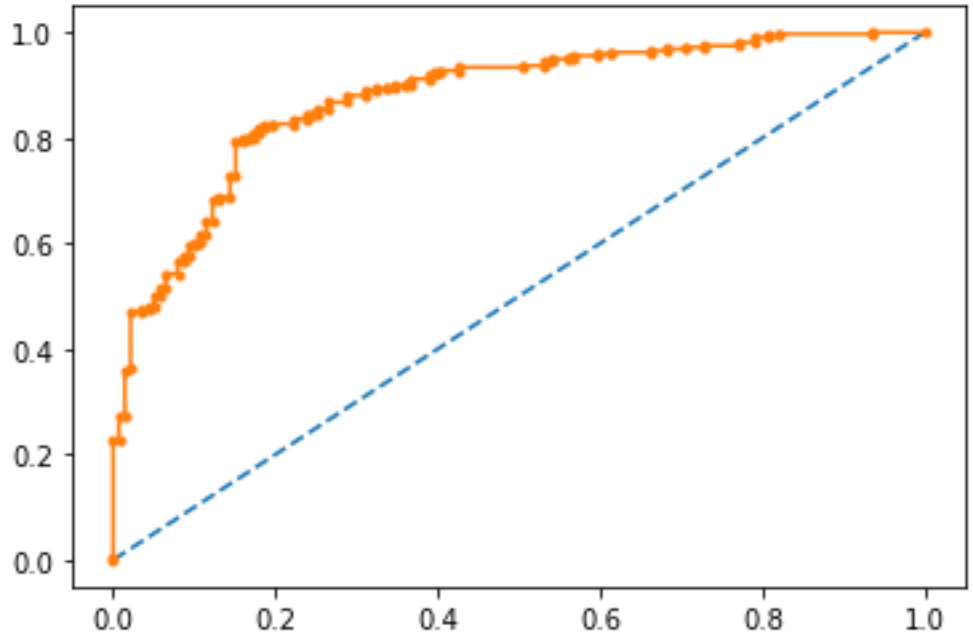
Training tuned

AUC: 0.895



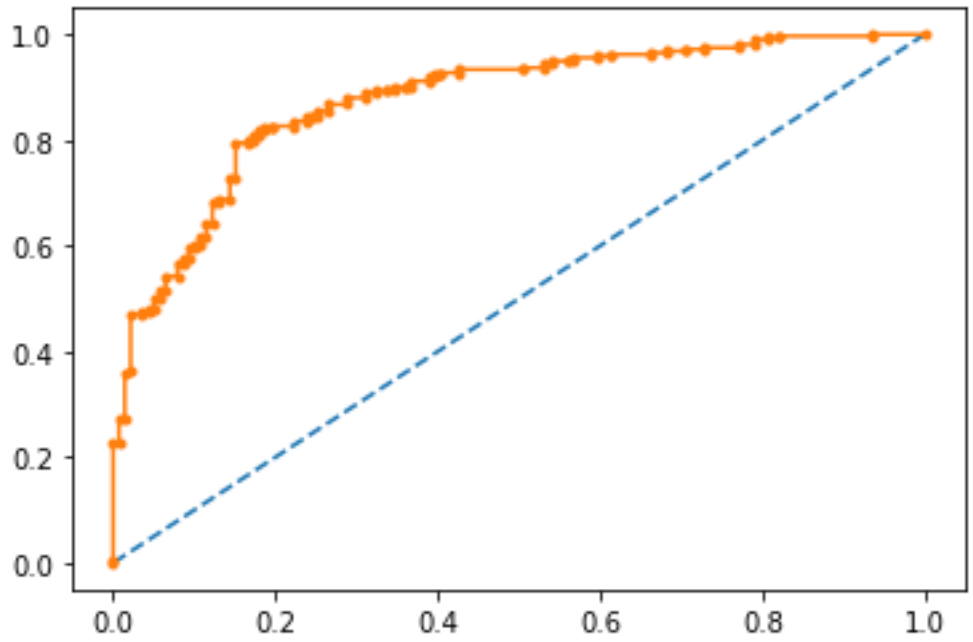
Testing

```
AUC Score : 0.877
```



Testing tuned

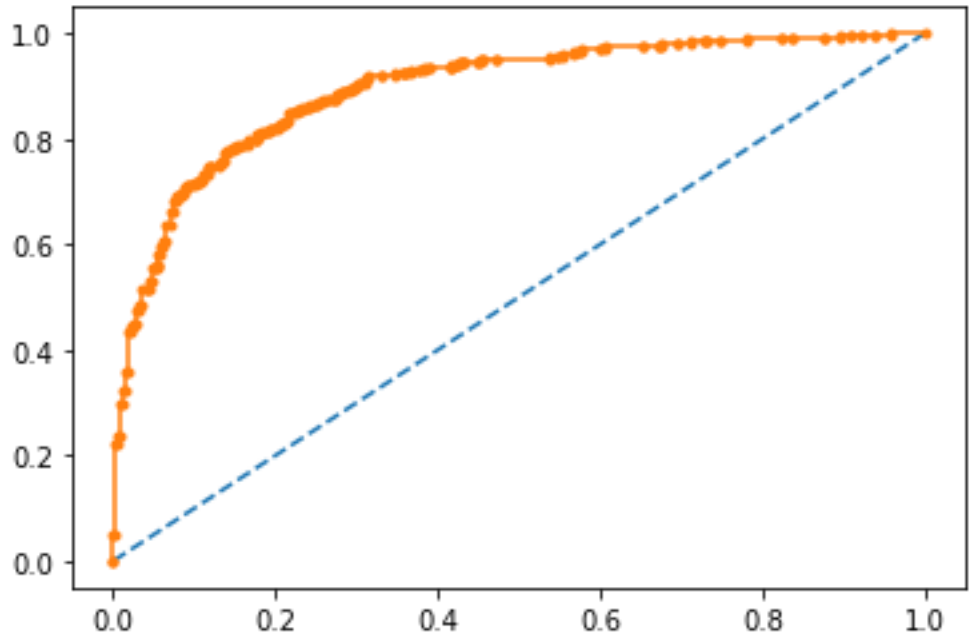
```
AUC: 0.877
```



2.) LDA

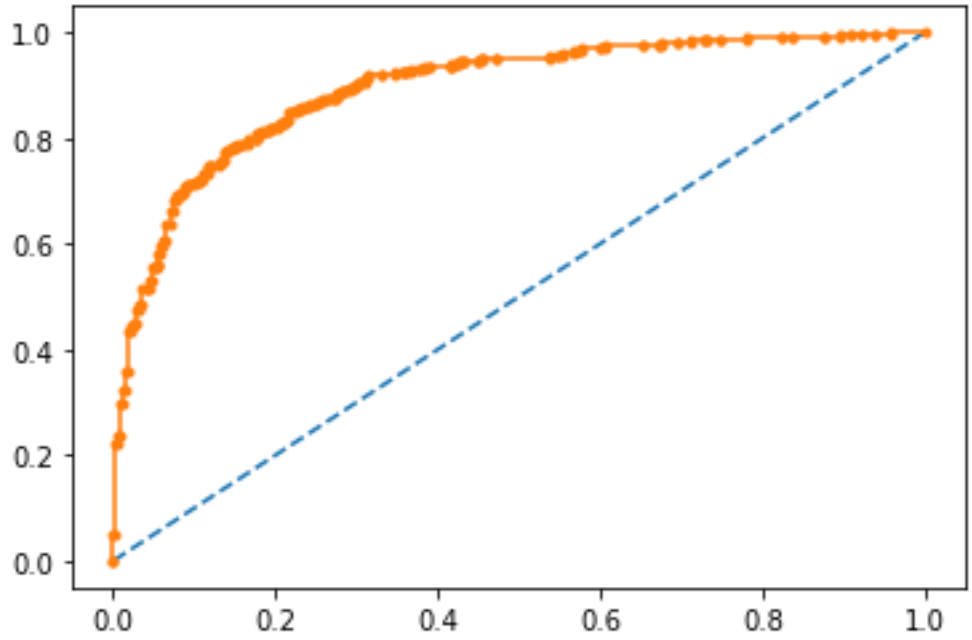
LDA TRAINING

AUC :0.895



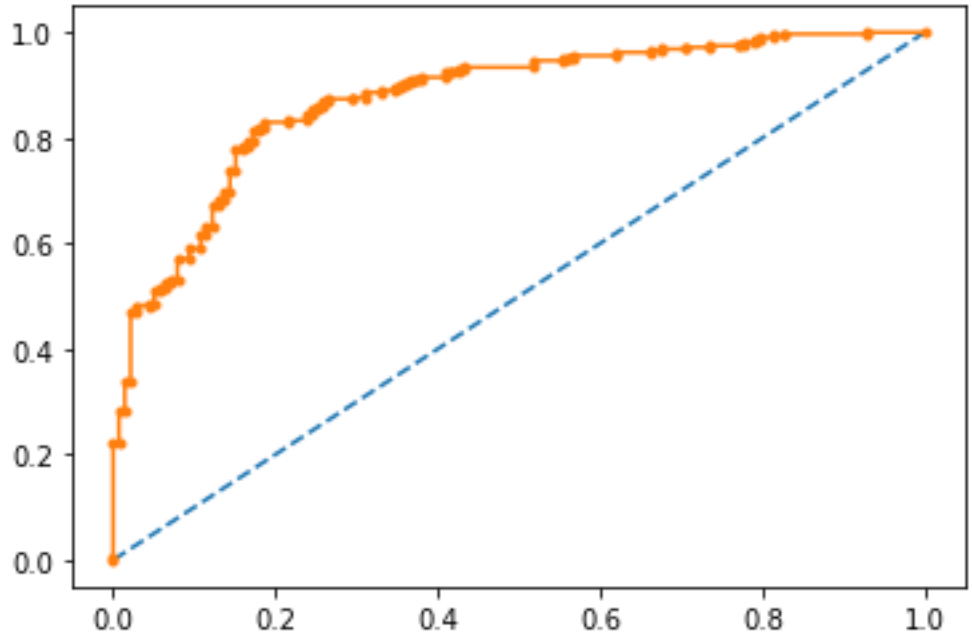
LDA TUNED TRANING

AUC: 0.895



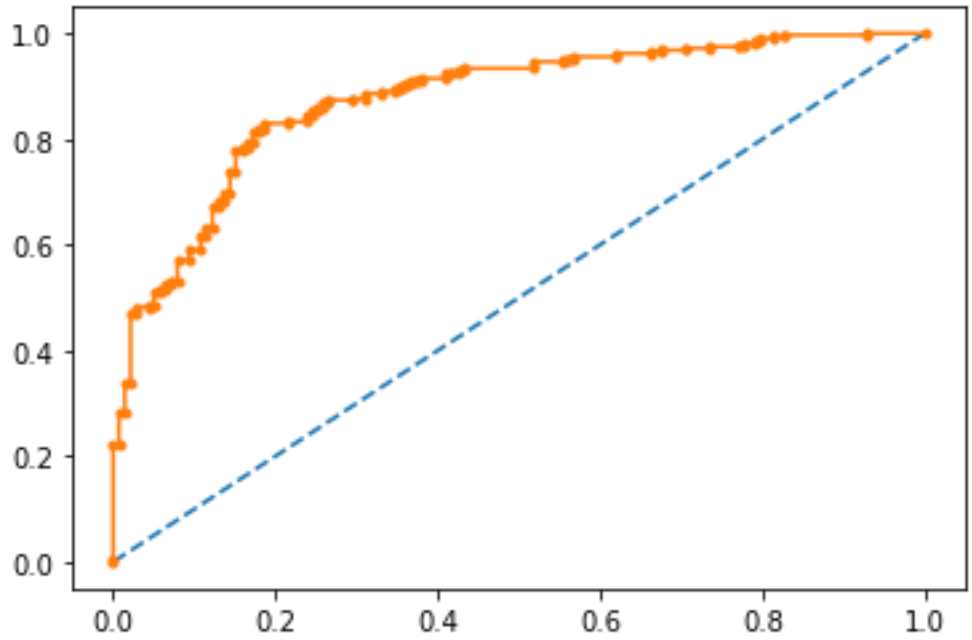
LDA TESTING

AUC :0.876



LDA TUNED TESTING

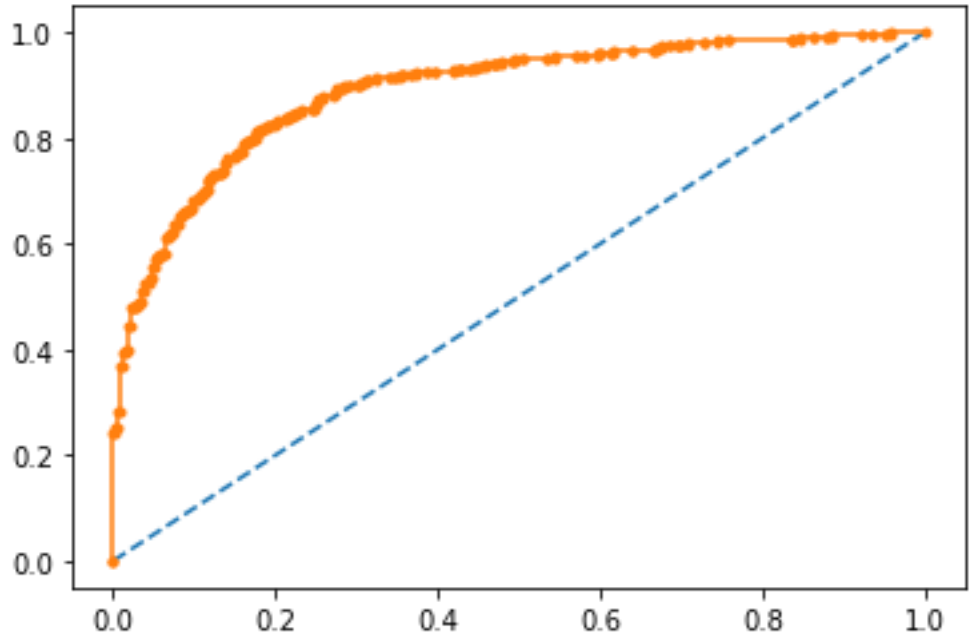
AUC: 0.876



3.) NAÏVE BAYES

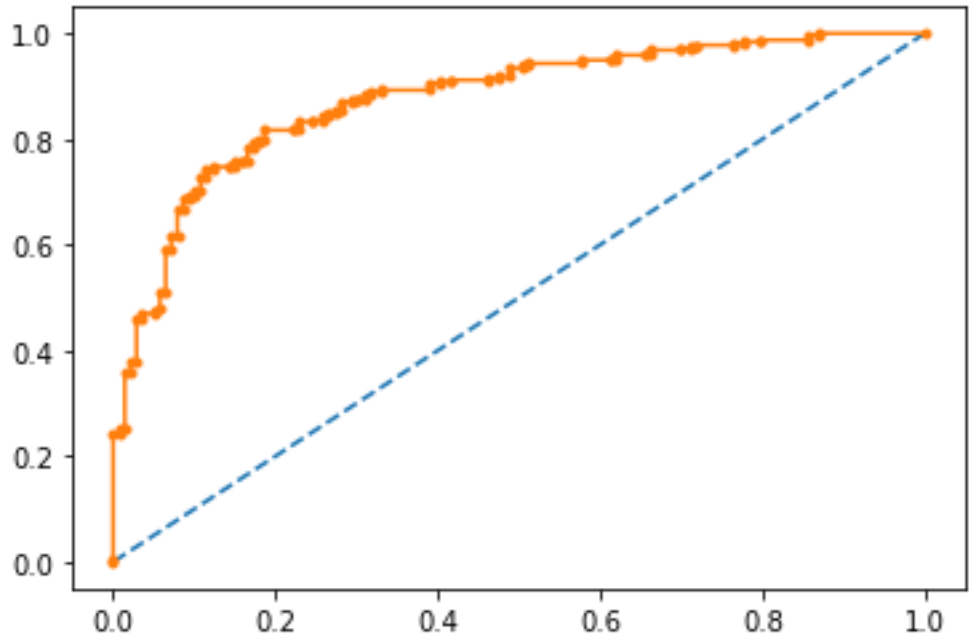
TRAINING

AUC :0.891



TESTING

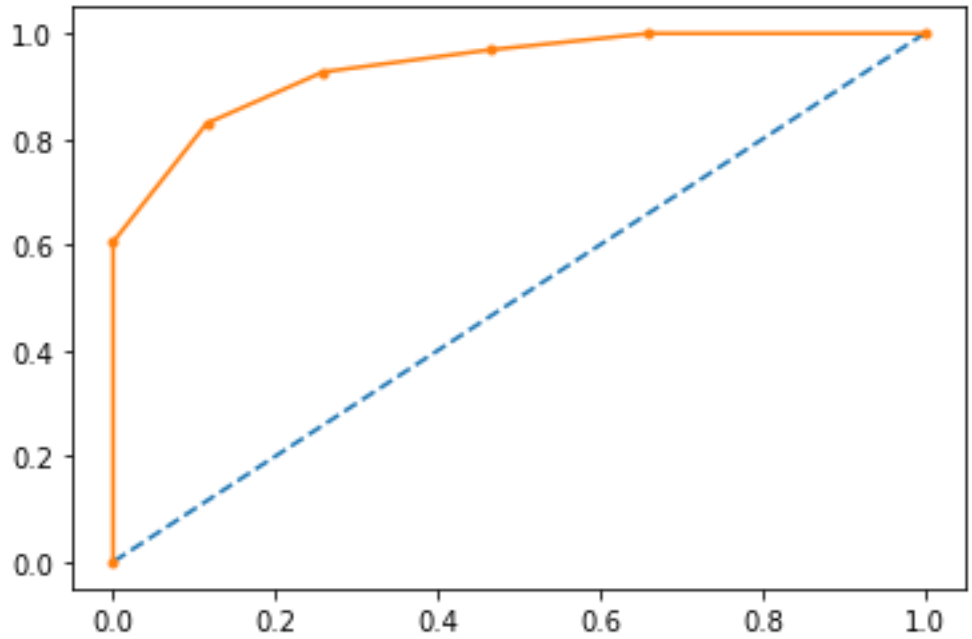
AUC :0.877



4.) KNN(k =5)

KNN TRAINING

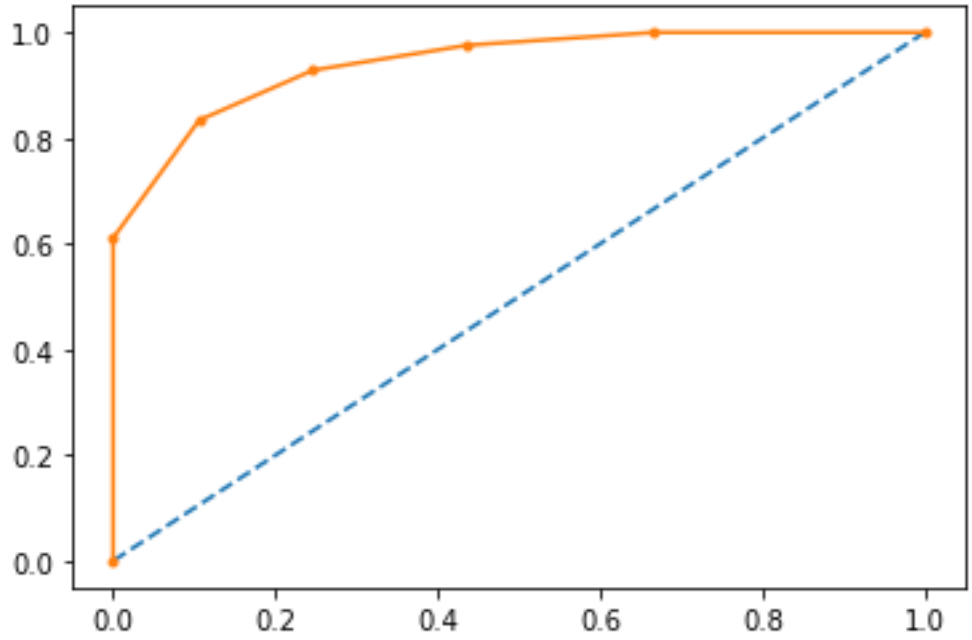
AUC: 0.936



In [770]:

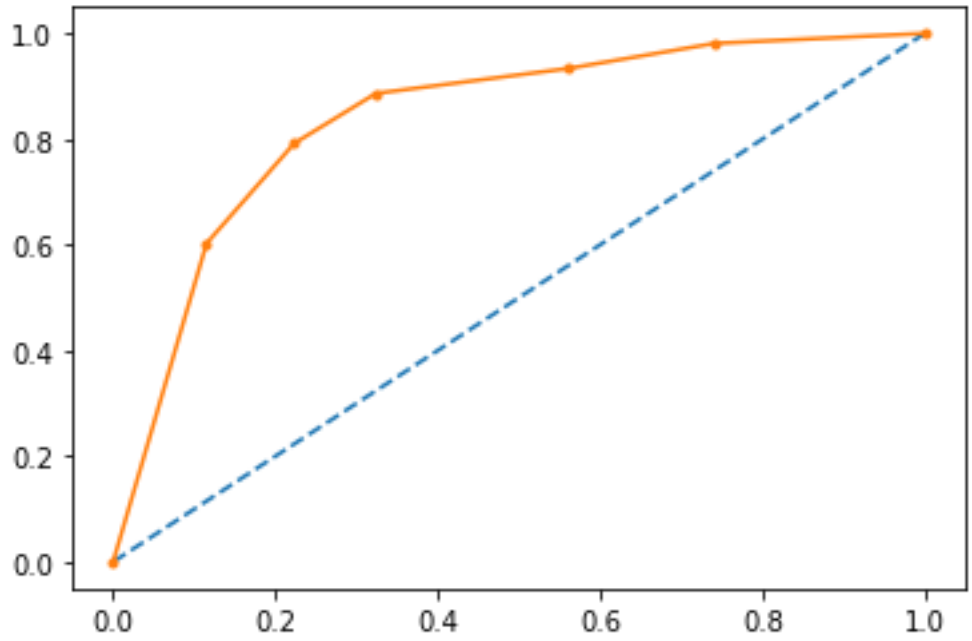
KNN TUNED TRAINING

AUC: 0.942



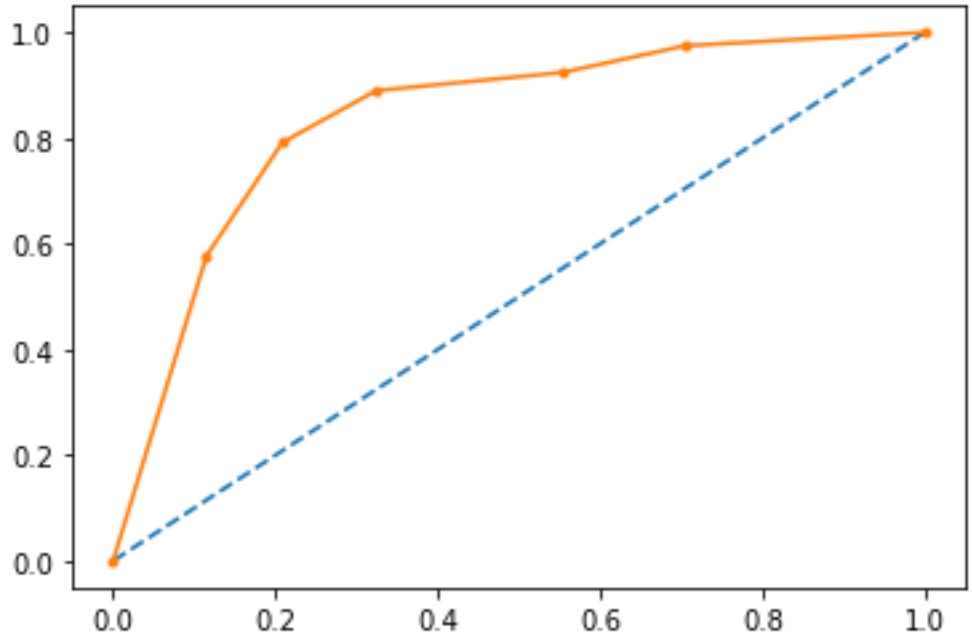
KNN TESTING

AUC: 0.839



KNN TUNED TESTING

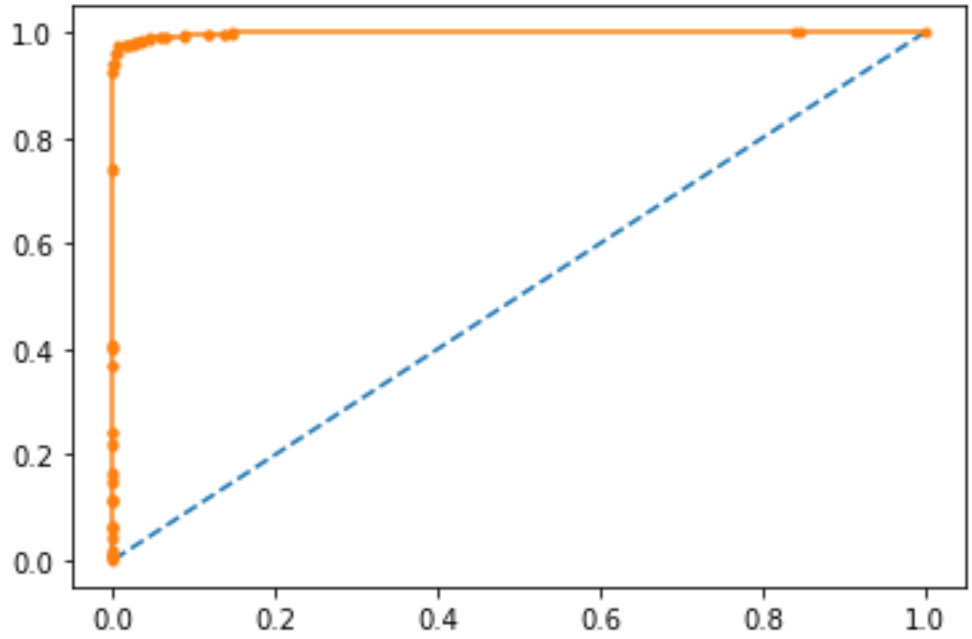
AUC: 0.838



5.) Bagging

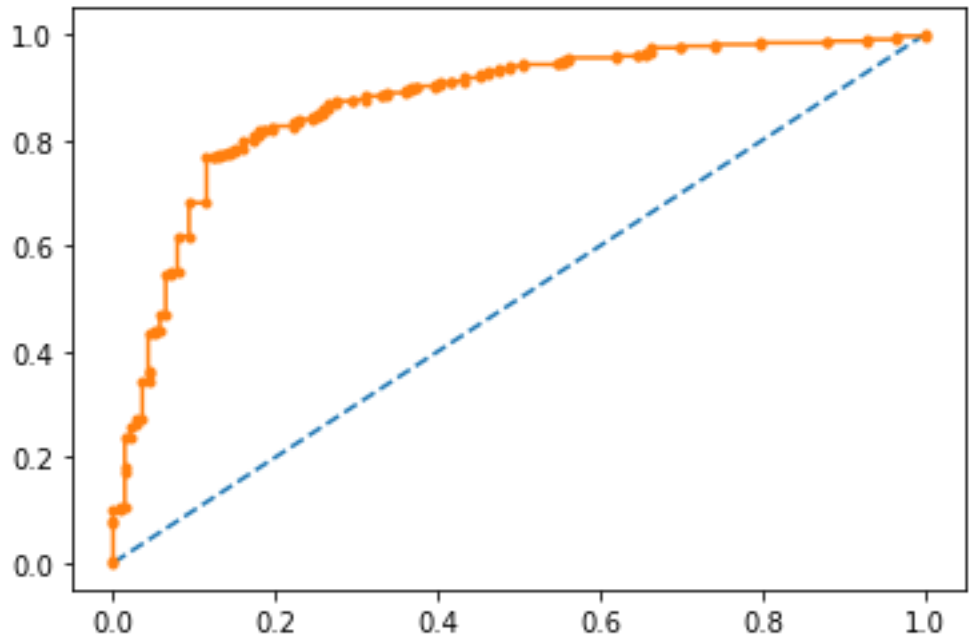
Training

```
AUC: 0.998
```



Testing

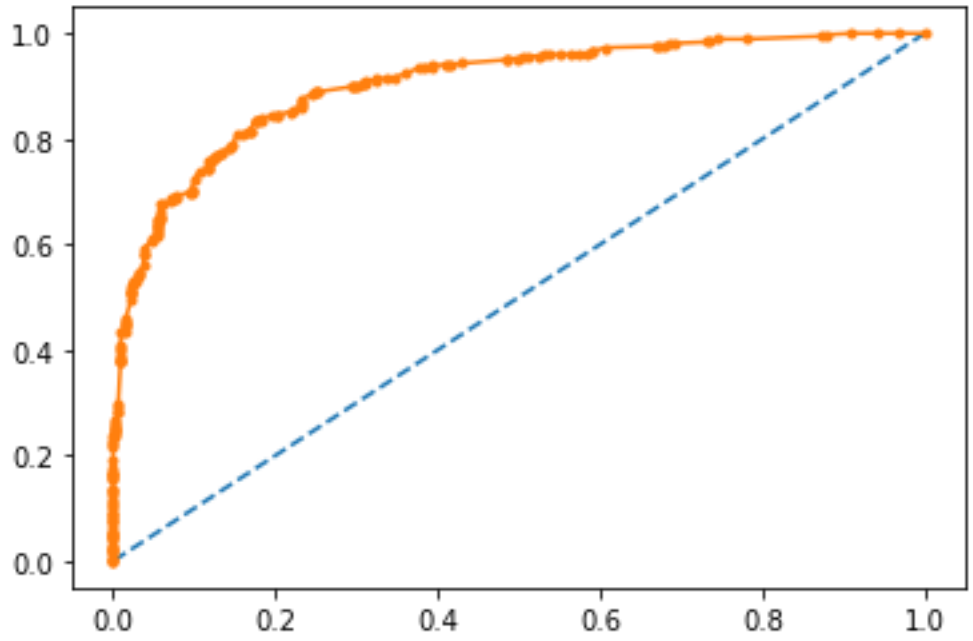
```
AUC: 0.871
```



6.) BOOSTING(Gradientboosting)

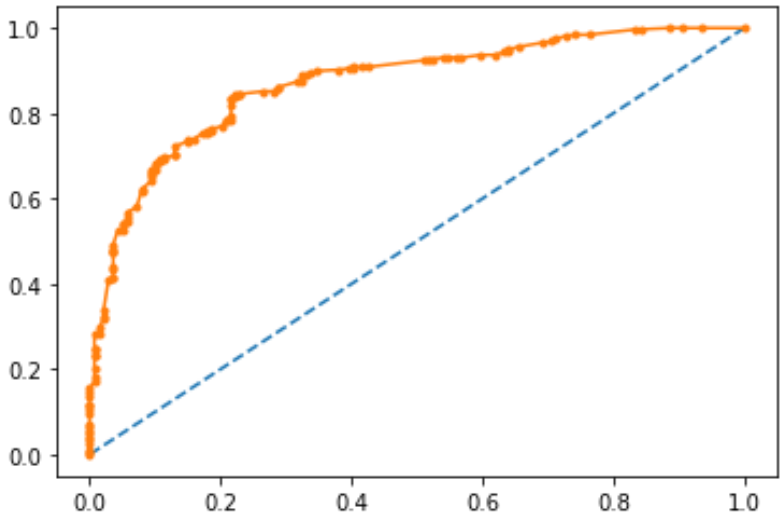
Training

```
AUC: 0.904
```



Testing

AUC: 0.872



MODEL ACCURACY COMPARISON

	LOGISTIC REG		LDA		NAÏVE BAYES		KNN		BAGGING		BOOSTING	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Without tuning	0.84	0.82	0.83	0.82	0.84	0.82	0.87	0.82	0.97	0.81	0.83	0.81
With tuning	0.85	0.82	0.84	0.82	0.84	0.82	0.88	0.82	0.97	0.81	0.83	0.81

All the model are having overfitting problem and similar accuracies too.From the above metrics we can clearly see that knn model with k=5 has better accuracy compares to all other model. Overfitting to an extended we have reduced once we are tuned . Whereas the bagging model increased the training accuracy , but performs poor on testing side.

For knn model once we tuned with different parameters, we have increased 1%of training accuracy ,but testing accuracy still remains same. A model with less than 10% difference in training and testing model accuracies can be considered as a good model . With that condition , we can conclude, knn will be right model to predict the voters from this given dataset.

MODEL F1-SCORE COMPARISON

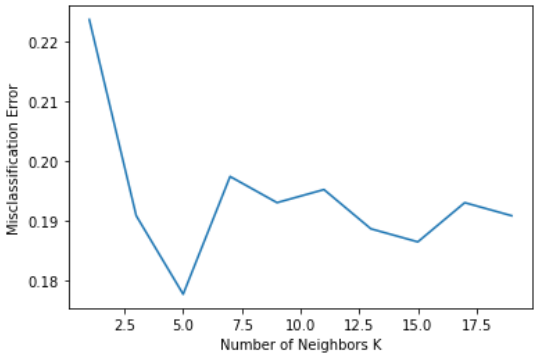
NON-TUNED	LOGISTIC REG	LDA	NAÏVE BAYES	KNN	BAGGING	BOOSTING

	CONSERVATIVE	LABOUR	CONSERVATIVE	LABOUR	CONSERVATIVE	LABOUR	CONSERVATIVE	LABOUR	CONSERVATIVE	LABOUR	CONSERVATIVE	LABOUR
Training	73	89	72	89	74	89	78	91	96	98	72	89
Testing	69	87	70	87	70	88	70	87	67	87	69	87

if we closely observe the f1-score for both conservative and labour parties, KNN model with k=5 perform the best comparing to rest of the model. Since there is an imbalance problem in the target variable. We are able to predict 70% of conservative voters prediction and 88% of labour voters from the given dataset. After tuning we have increased the f1score for labour voters by 1% using knn model.

TUNED	LOGISTIC REG		LDA		NAÏVE BAYES		KNN		BAGGING		BOOSTING	
	CONSERVATIVE	LABOUR	CONSERVATIVE	LABOUR	CONSERVATIVE	LABOUR	CONSERVATIVE	LABOUR	CONSERVATIVE	LABOUR	CONSERVATIVE	LABOUR
TRAINING	73	89	72	89	74	89	79	91	-	-	-	-
TESTING	69	87	70	87	70	88	70	88	-	-	-	-

After calculating the best number of k nearest neighbour , we got to see k=5 have better performance compares to k=6



Above figure represent the misclassification error with respect to number of k values. We can see that between 5 and 6 there is a stable value of error. When k=5, the error is also less(0.18). This why , our model is best compares to all other k values. Rest of the models are also good , but training accuracies are less compares to knn model. If we look onto the bagging and boosting classifier, they are performing very good in terms of training model , but the performance is less compares to testing side.Using knn model with k value =5, we are able to predict the labour and conservative voters voting to their respective parties.

PROBLEM NO.2

2.1We have imported the three text files from inaugural fieldIDs.

inaugural.raw('1941-Roosevelt.txt')

inaugural.raw('1961-Kennedy.txt')

inaugural.raw('1973-Nixon.txt')

we are assigned these text files into variable called ‘x’ converting them to list files.After that we have to call these text files into a new dataframe. So that we can easily perform the rest of the task.Let us call the new dataframe as ‘y’. The speech inside each files are converted into the list format and applied to the text column of the ‘y’ dataFrame.

Roosevelt speech

Number of words :-

	Text	totalwords
0	On each national day of inauguration since 178...	1360

Character count :-

	Text	char_count
0	On each national day of inauguration since 178...	7571

Average words :-

	Text	avg_word
0	On each national day of inauguration since 178...	4.539706

Nixon Speech

Number of words:-

	Text	totalwords
0	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...	1819

Character counts

	Text	char_count
0	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...	9991

Avg words

	Text	avg_word
0	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...	4.465091

kennedy Speech

Let us call the text file into dataframe y.Insert the list of speech values into the text column.

	Text
0	Vice President Johnson, Mr. Speaker, Mr. Chief...

Number of words:-

	Text	totalwords
0	Vice President Johnson, Mr. Speaker, Mr. Chief...	1390

Character counts:-

	Text	char_count
0	Vice President Johnson, Mr. Speaker, Mr. Chief...	7618

Avg words:-

	Text	avg_word
0	Vice President Johnson, Mr. Speaker, Mr. Chief...	4.461871

All representation

	Text	totalwords	char_count	avg_word	No_of_stopwords
0	vice president johnson mr speaker mr chief jus...	1390	7618	4.461871	618

2.2 For Roosevelt speech we have around 632 stopwords . We have to remove all the stopwords and punctuation from the speech to find out the frequent words in the speech.

From nixon speech we got around 899 stopwords. And for kennedy speech there are around 618 stopwords. Before removing the stopwords, we are first converting the speech files into lower cases and splitting them into words. After that special characters, numerical values and punctuations are removed from the text files. Using nltk corpus library we are importing stopword function and calling them to remove stopwords.

Roosevelt speech

	Text	No_of_stopwords
0	On each national day of inauguration since 178...	632

Nixon speech

	Text	No_of_stopwords
0	Mr. Vice President, Mr. Speaker, Mr. Chief Jus...	899

Kennedy speech

	Text	No_of_stopwords
0	Vice President Johnson, Mr. Speaker, Mr. Chief...	618

2.3) The most frequent words coming in the three speeches are :-

Roosevelt speech

```
nation       11
know         10
democracy     9
spirit        9
life          8
us            8
people        7
america       7
freedom       6
years         6
dtype: int64
```

roosevelt top three words used are **nation , know and democarcy**

Nixon speech

```
us                26
let               22
peace             19
world             16
new               15
america           13
responsibility    11
government        10
great              9
home               9
dtype: int64
```

nixon has used three words frequently in his speech like ,.**us, let and peace**

Kennedy speech

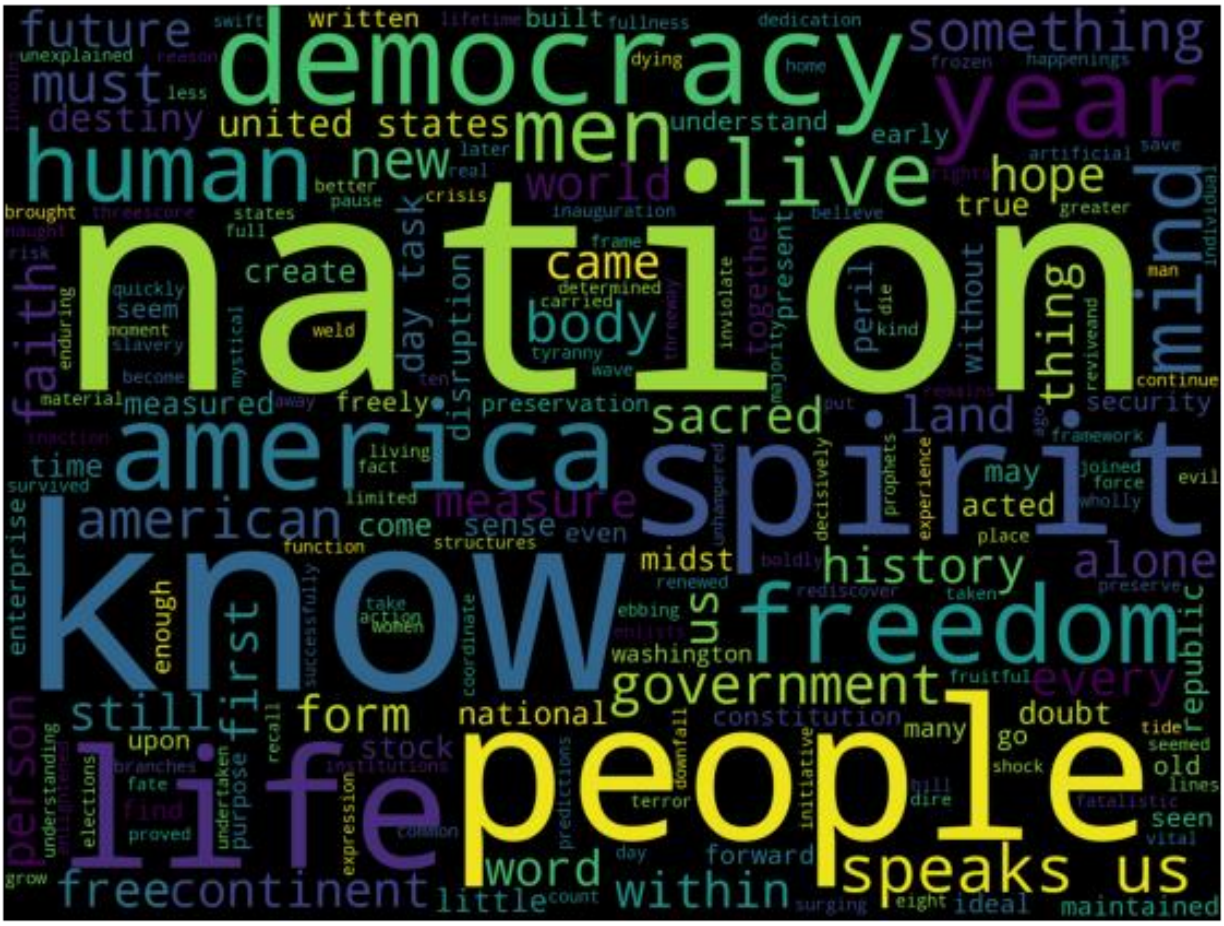
```
let         16
us          12
world        8
sides        8
pledge       7
new          7
citizens     5
nations      5
shall        5
power        5
dtype: int64
```

kennedy has used the words **let , us ,world** as the top three words during his speech

From the above three speeches 'US' is the word which is used by three presidents in their speeches.

2.4) We have used wordcloud library in python to apply the repetitive words in the speeches of respectives presidents during the election. Each president has their own perspective during their speeches. But mostly everyone has tried to highlight the unity. ‘US’ , ‘world’ these are words which they used frequently to highlight the strength of unity.

Roosevelt



Nixon

Kennedy