Machine Learning Project Analysis by Shruthidhar

Problem-1&2

GreatLearning

Machine Learning Project:

Scoring guide (Rubric) - Machine Learning Project

	Page nc
1.1) Read the dataset. Describe the data briefly. Interpret the inferences for each. Initial steps like head() .info(), Data Types, etc . Null value check, Summary stats, Skewness must be discussed.	8-9
1.2) Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also check for outliers (4 pts). Interpret the inferences for each (3 pts) Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.	10-22
1.3) Encode the data (having string values) for Modelling. Is Scaling necessary here or not?(2 pts), Data Split: Split the data into train and test (70:30) (2 pts). The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for example std dev, variance, etc. Should justify whether there is a necessity for scaling. Object data should be converted into categorical/numerical data to fit in the models. (pd.categorical().codes(), pd.get_dummies(drop_first=True)) Data split, ratio defined for the split, train-test split should be discussed.	23-24
1.4) Apply Logistic Regression and LDA (Linear Discriminant Analysis) (2 pts). Interpret the inferences of both model s (2 pts). Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)	25-28
1.5) Apply KNN Model and Naïve Bayes Model (2pts). Interpret theinferences of each model (2 pts). Successful implementation of each	29-31

List of Tabels:

List of Diagrams

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 part.

****Data Dictionary****

1. vote: Party choice: Conservative or Labour

2. age: in years

3. economic.cond.national: Assessment of current national economic conditions, 1 to 5.

4. economic.cond.household: Assessment of current household economic conditions, 1 to 5.

5. Blair: Assessment of the Labour leader, 1 to 5.

6. Hague: Assessment of the Conservative leader,1 to 5.

7. Europe: an 11-point scale that measures respondents' attitudes toward European integration. High scores represent ‘Eurosceptic’ sentiment.

8. political.knowledge: Knowledge of parties' positions on European integration, 0 to 3.

9. gender: female or male.

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

Head & Tail of Dataset

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

Table1

	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
1522	Conservative	73	2	2	4	4	8	2	male
1525	Conservative	74	2	3	2	4	11	0	female
1523	Labour	37	3	3	5	4	2	2	male
1524	Conservative	61	3	3	1	4	11	2	male
1521	Conservative	67	5	3	2	4	11	3	male

Table 2

No Null Values in the Data set

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

Unique variabels of the dataset

```
vote                        2
age                        70
economic.cond.national      5
economic.cond.household     5
Blair                       5
Hague                       5
Europe                     11
political.knowledge         4
gender                      2
```

Categorical variabels of the dataset:

```
['vote', 'gender']
```

Numerical Values of the Dataset:

```
['age', 'economic.cond.national', 'economic.cond.household', 'Blair', '
Hague', 'Europe', 'political.knowledge']
```

Description of the dataset:

	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

Table 3

1.2) Perform EDA (Check the null values, Data types, shape, Univariate, bivariate analysis). Also check for outliers (4 pts). Interpret the inferences for each (3 pts) Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.

Univariate Analysis:

Pie Chart:

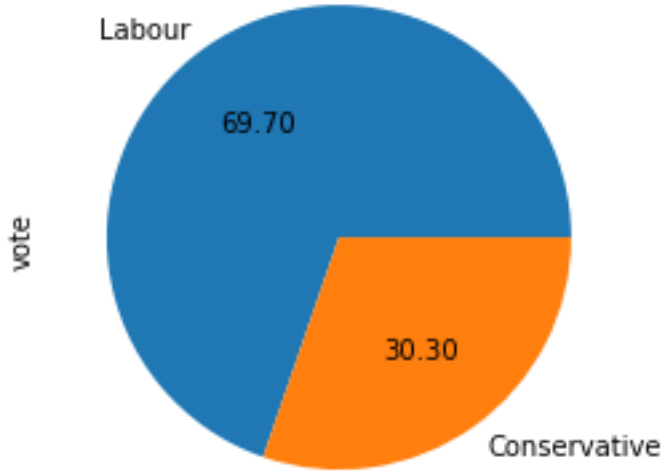


Image :1

From the above Pie chart, we can analyize the vote distribution between ‘Labour’ & ‘Conservative’ for the dependent variable ‘Vote’.

We can clearly see that ‘Labour’ party is having more ‘Vote’ than ‘Conservative’.

Labour : 69.70

Conservative : 30.30

Histplot:

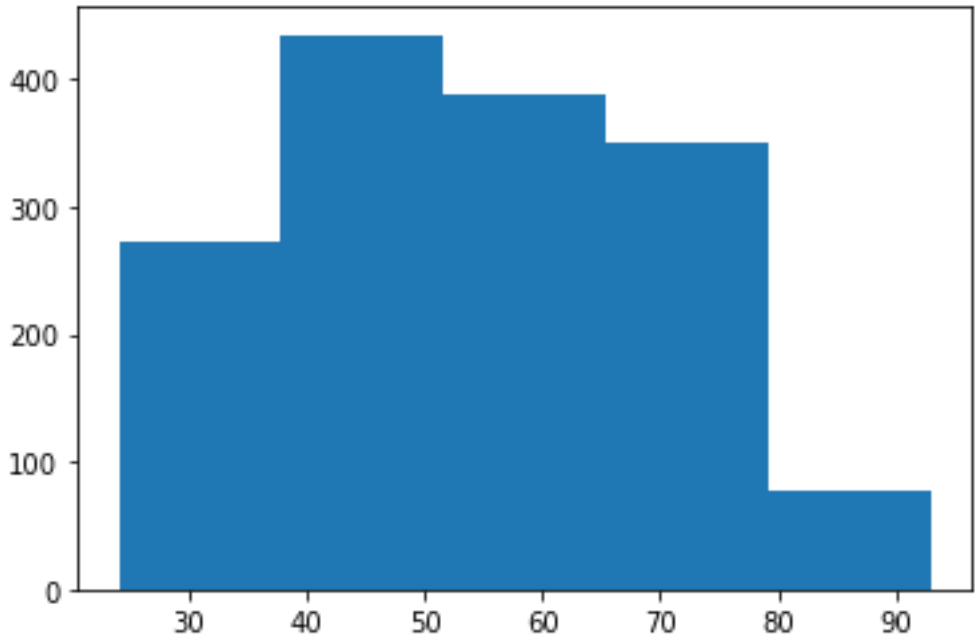


Image:2

This Histplot ,analyses about the ‘Age’, this plot shows from age 40 to 70 the voters are high.

Bivariate Analysis/Multi Variate Analysis:

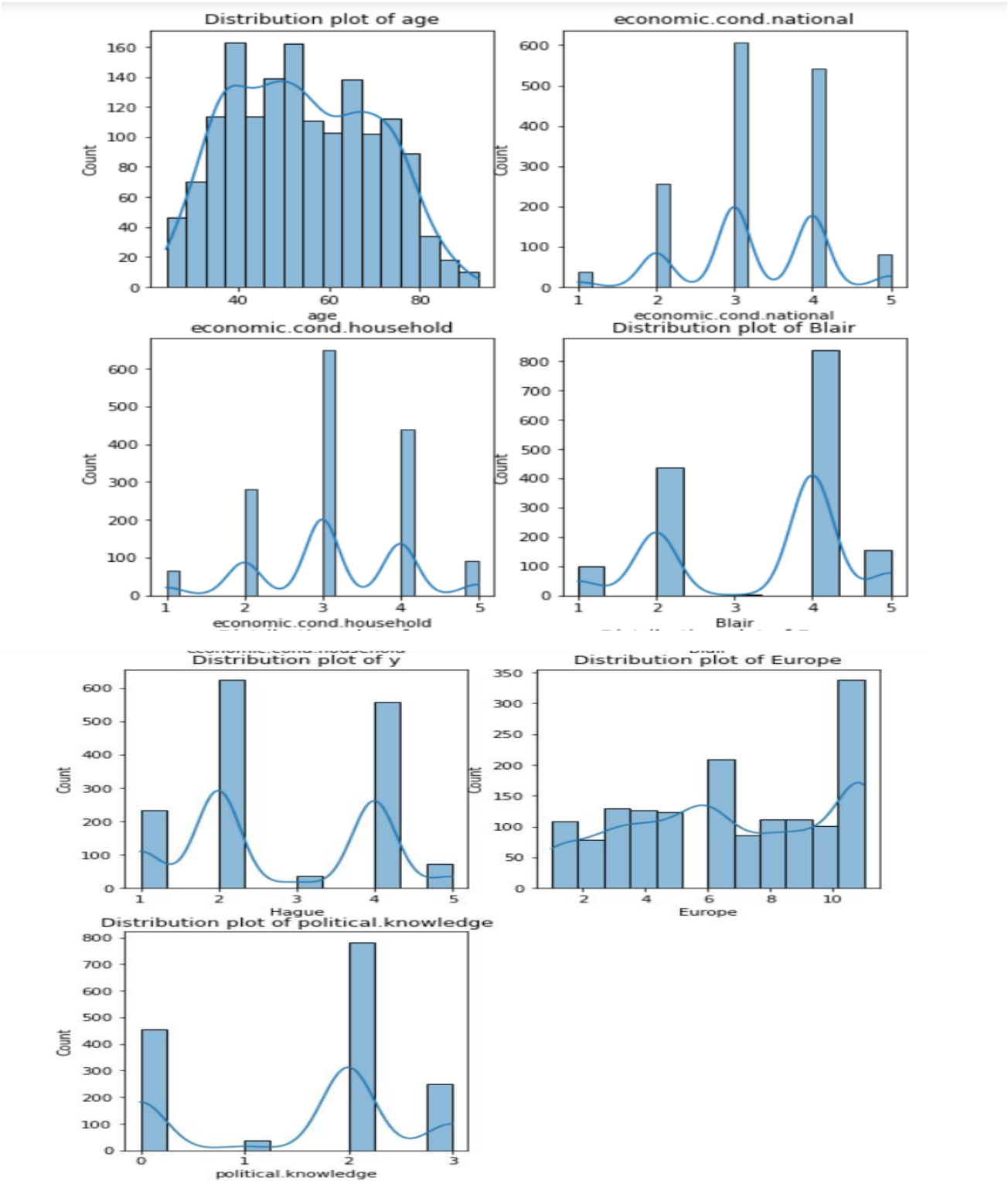


Image: 3

From the above DistPlot we can analyse the distribution of the data, for all the Numerical Variables.

We can see that apart of '**Age'** variable is having "**Normal Distibution'.**

And all the other Numeric variables, 'economic.cond.national', 'economic.cond .household', 'Blair', 'Hague', 'Europe', 'political.knowledge' are not having any Normal Distribution

Boxplot for checking Outliers:

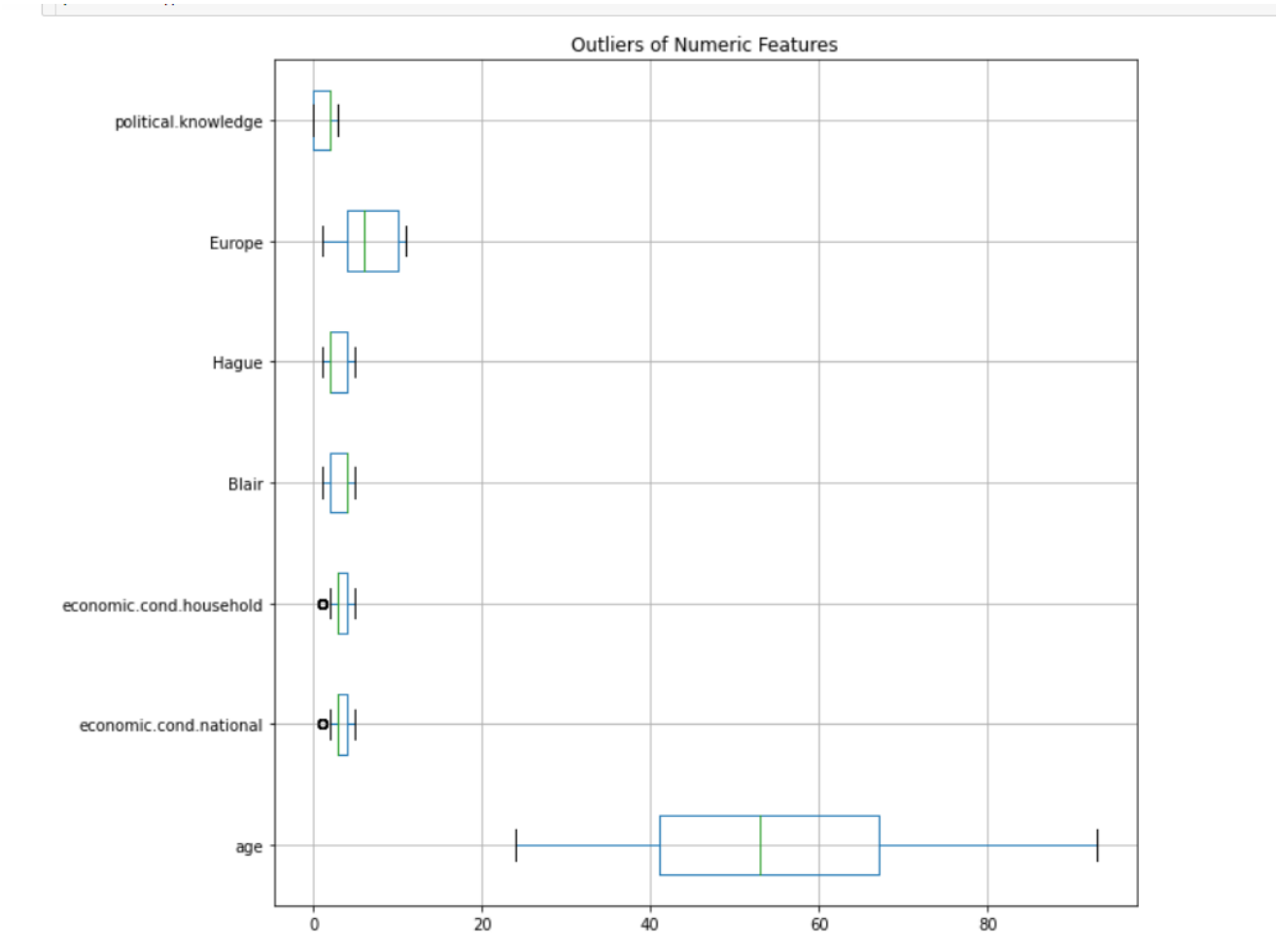


Image :4

From the above boxplot we can analysize that , there very less outliers in the data, & for few variables like 'economic.cond.national', 'economic.cond.household', have outliers, but it can be neglected.

Skewness of the Dataset:

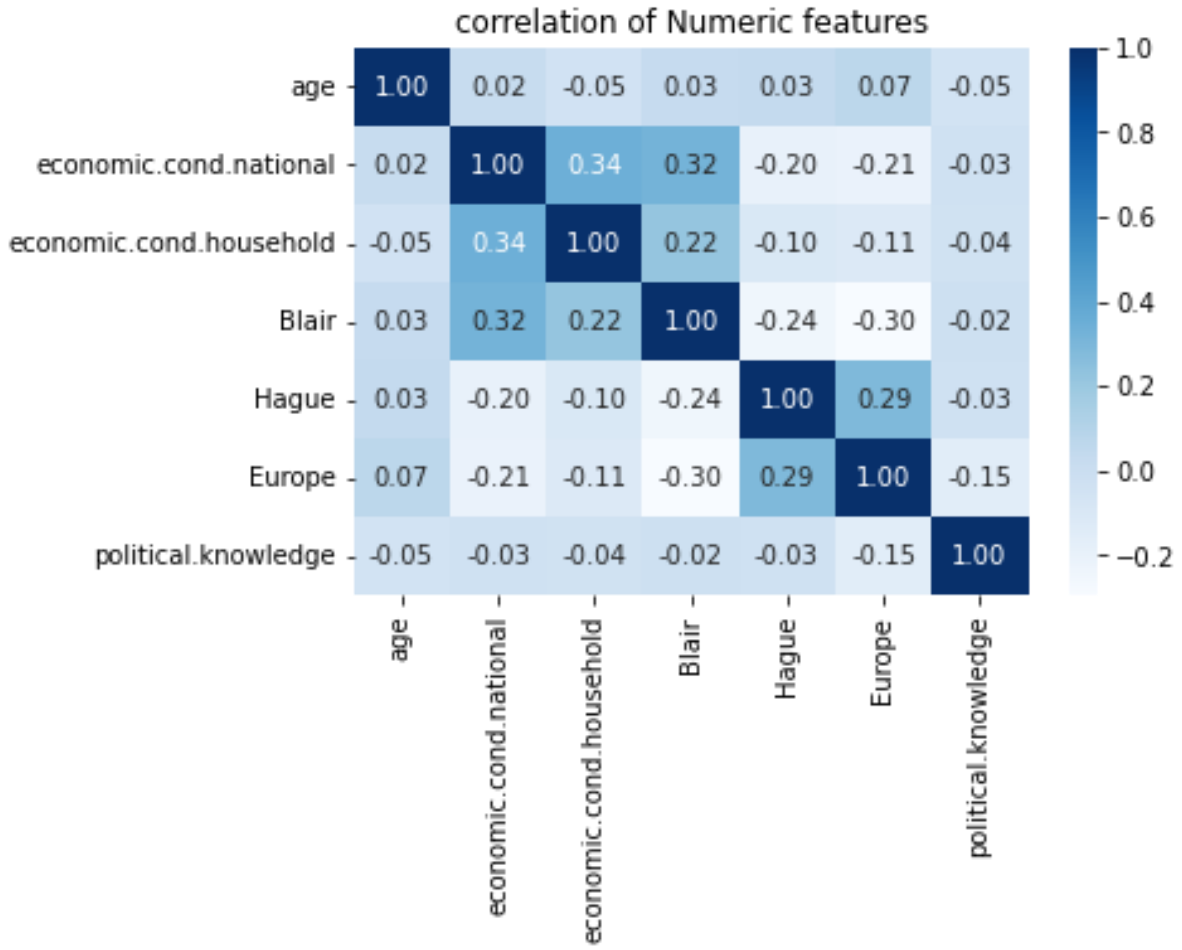
age	0.144621
economic.cond.national	-0.072349
economic.cond.household	0.086170
Blair	-0.535419
Hague	0.152100
Europe	-0.135947
political.knowledge	-0.426838

Table:4

Correlation of the Dataset:

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
age	1.000000	0.022194	-0.047149	0.030218	0.034626	0.068880	-0.048490
economic.cond.national	0.022194	1.000000	0.342942	0.324402	-0.198091	-0.206858	-0.029395
economic.cond.household	-0.047149	0.342942	1.000000	0.216123	-0.101243	-0.114202	-0.039803
Blair	0.030218	0.324402	0.216123	1.000000	-0.243210	-0.296162	-0.020917
Hague	0.034626	-0.198091	-0.101243	-0.243210	1.000000	0.287350	-0.030354
Europe	0.068880	-0.206858	-0.114202	-0.296162	0.287350	1.000000	-0.152364
political.knowledge	-0.048490	-0.029395	-0.039803	-0.020917	-0.030354	-0.152364	1.000000

HeatMap:



From this HeatMap we can analysie **relationships between two variables, one plotted on each axis**.

From the above plot we see that '**economic.cond.household'** is having high correlation with **'economic.cond.national' with 0.34.**

'economic.cond.national' with '**Blair**' are having correlation.

The other variables are having less variation /Negative correlation with each other.

Pairplot:

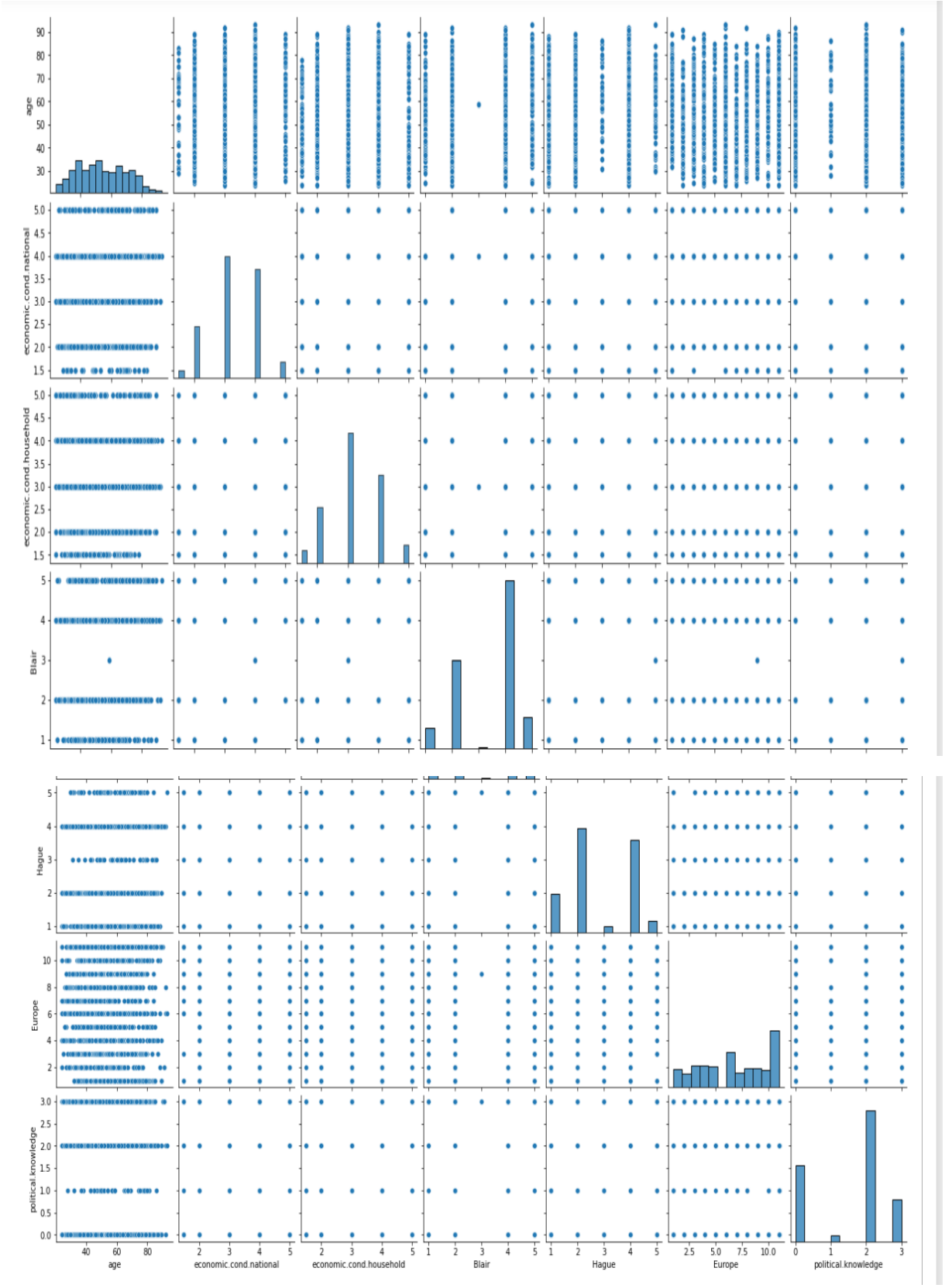
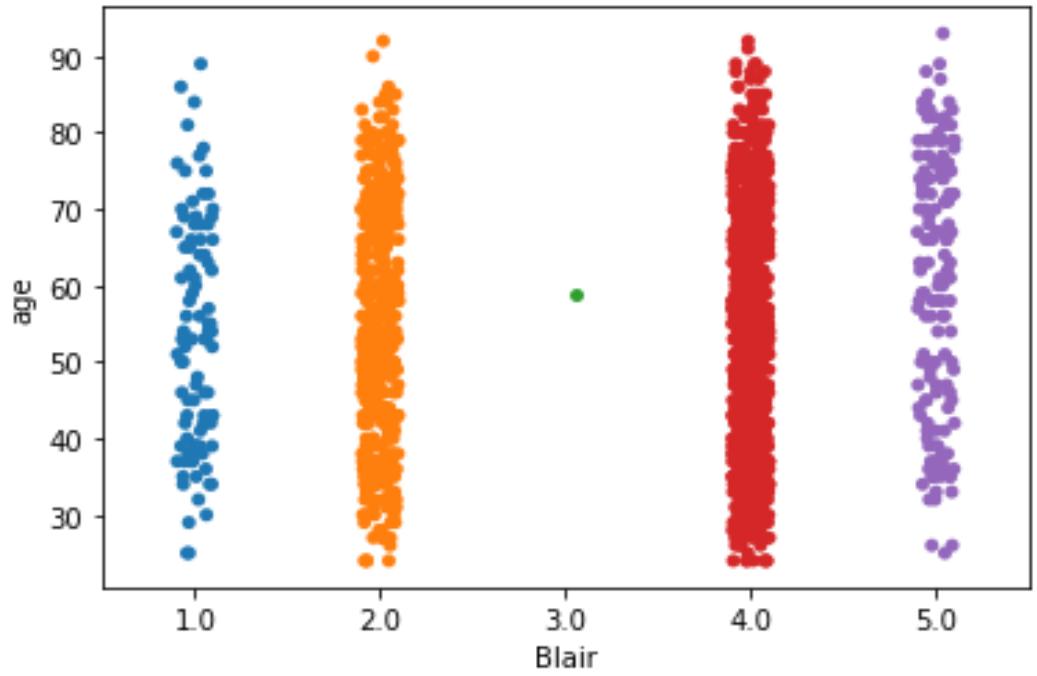

age
economic.cond.national
economic.cond.household
Blair
Hague
Europe
political.knowledge

Image: 5

From the above pairplot we can analyze the relationship for (n, 2) combination of variable in a DataFrame as a matrix of plots and the diagonal plots are the univariate plots.

Comparing the Independent Variables with each other:

Strip Plots:



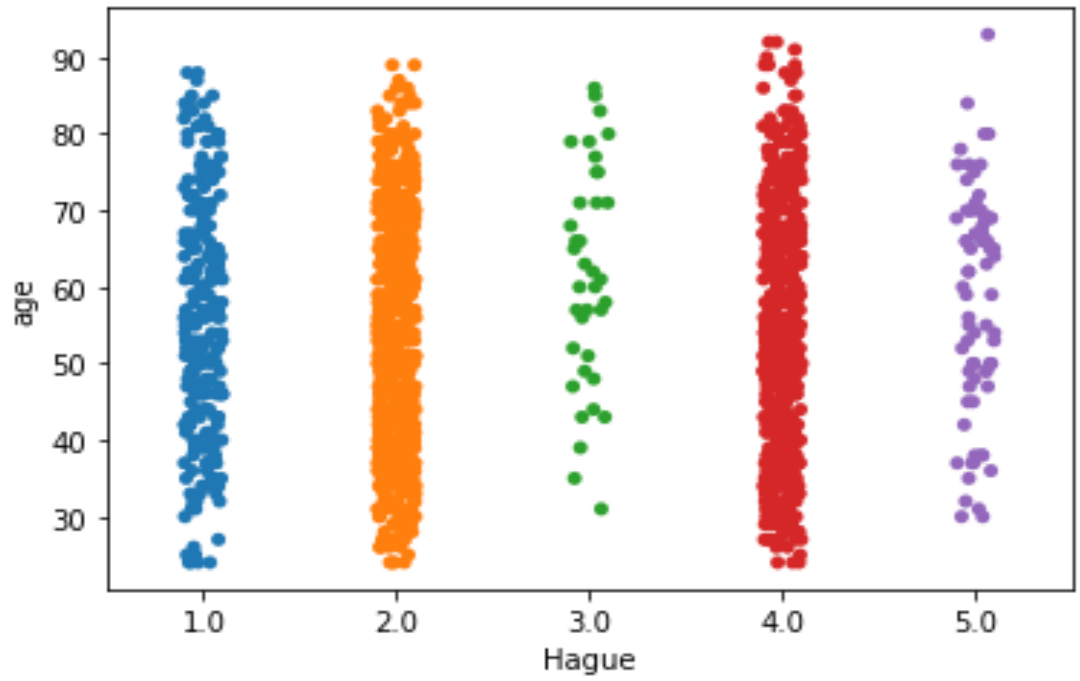


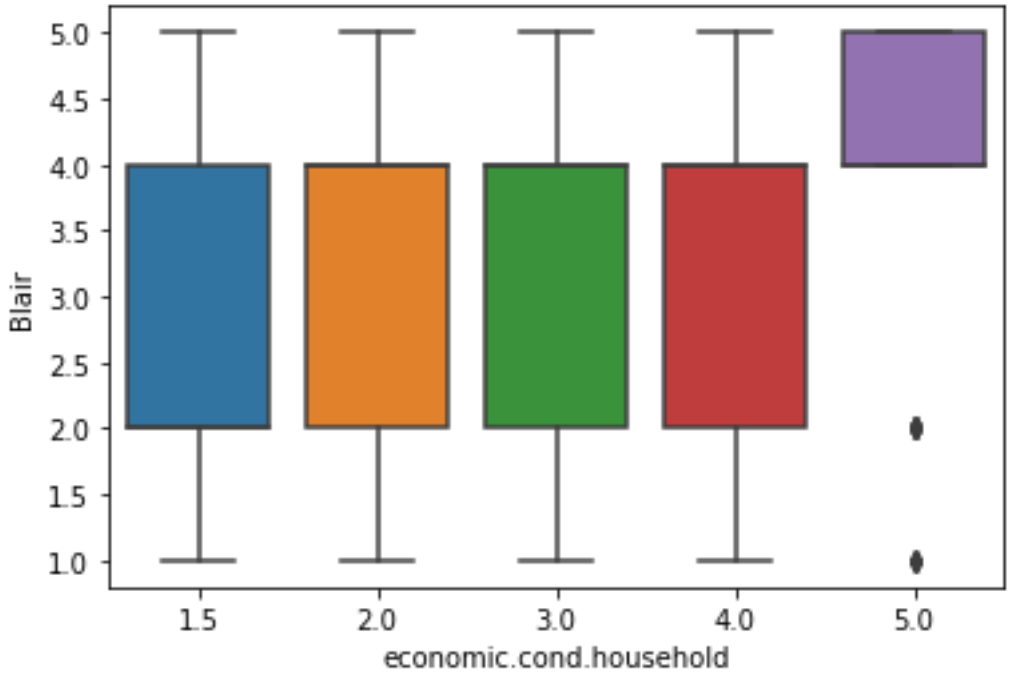
Image: 6

From the above Strip plots we can analyse by comparing with 2 numerical variables **Age**, with **Hague & Blair** .

From the above plot we can analyse that @ 3.0 Hague is having more datapoints than Blair.

@1.0 & 5.0 data points Blair is having more variables , than Hague.

From the above data we can analysize that “Age “ Column starts from 30 to 80 nearly.



BoxPlots:

Image :8

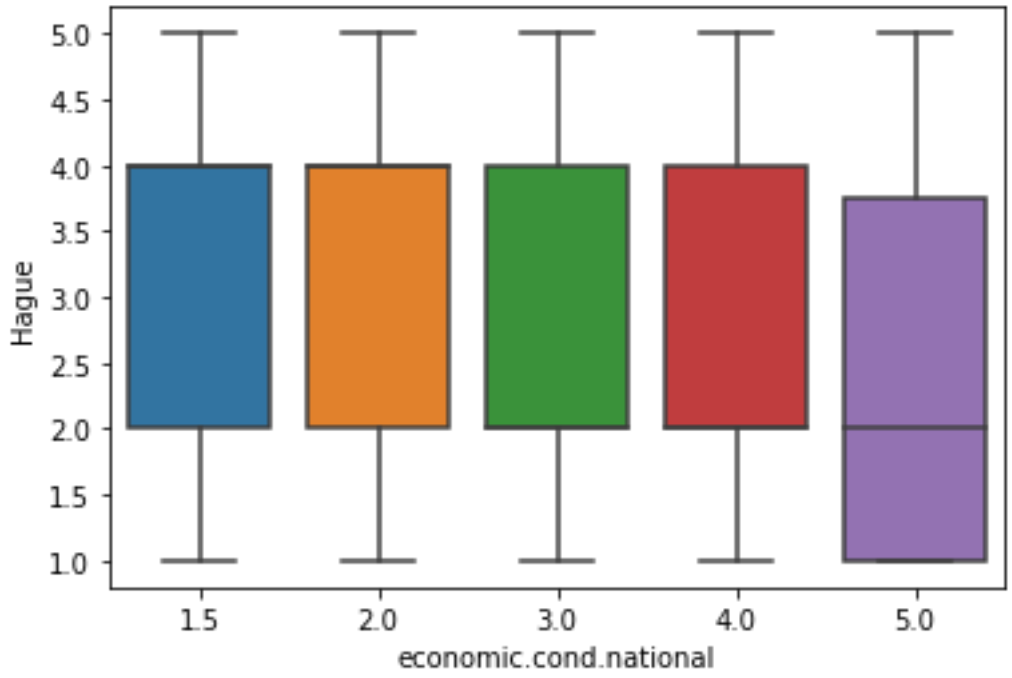


Image:

BarPlots:

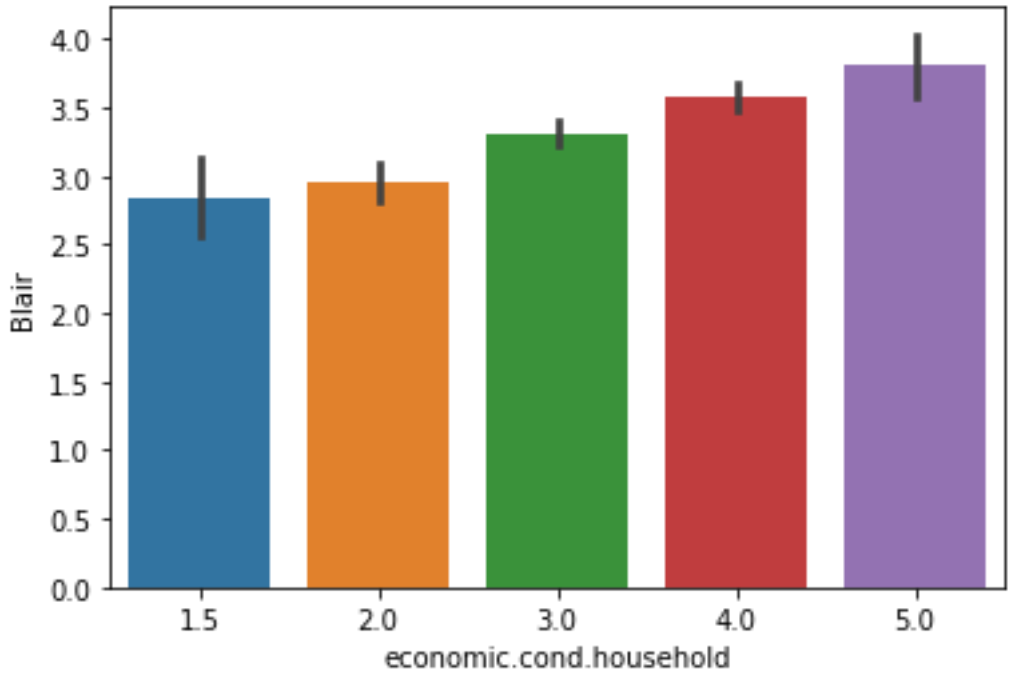


Image :9

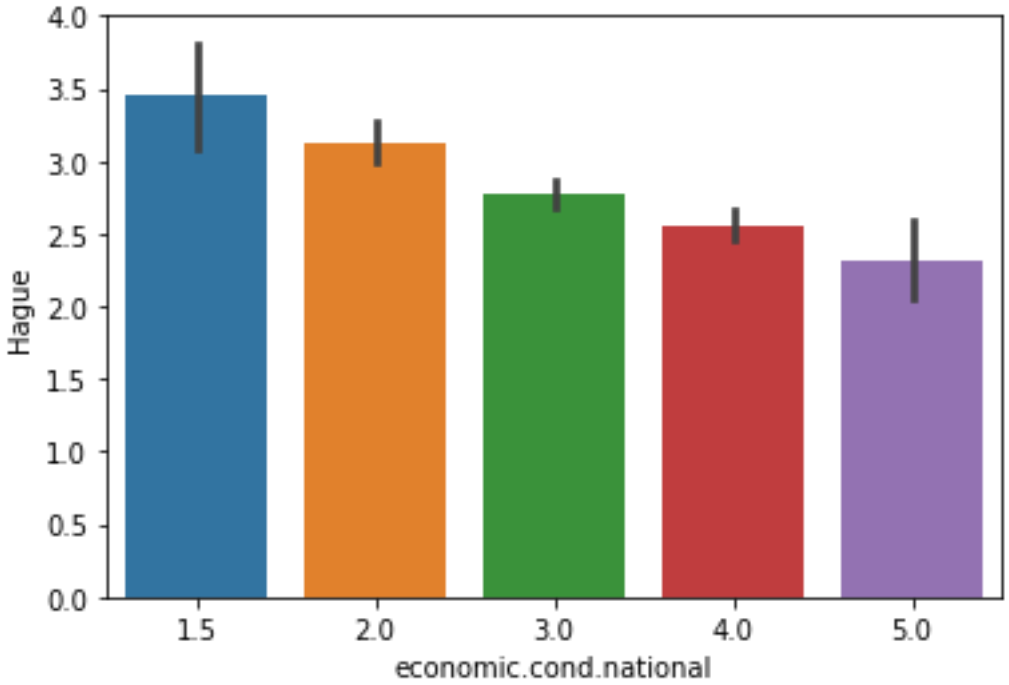

4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Hague
1.5
2.0
3.0
4.0
5.0
economic.cond.national

Image: 10

Histplots:

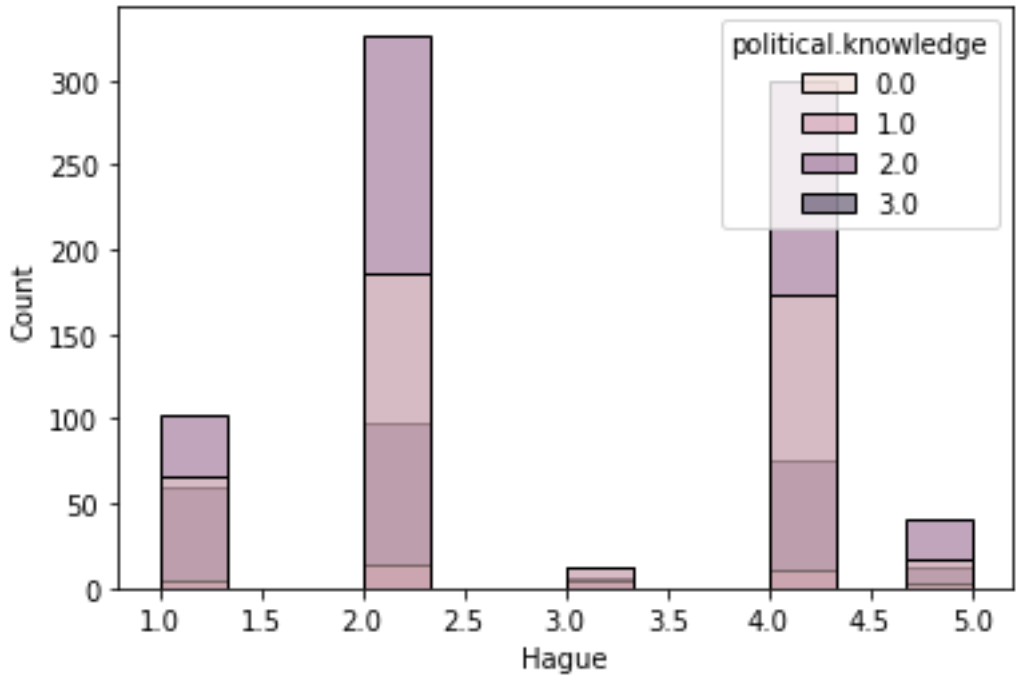


Image: 12

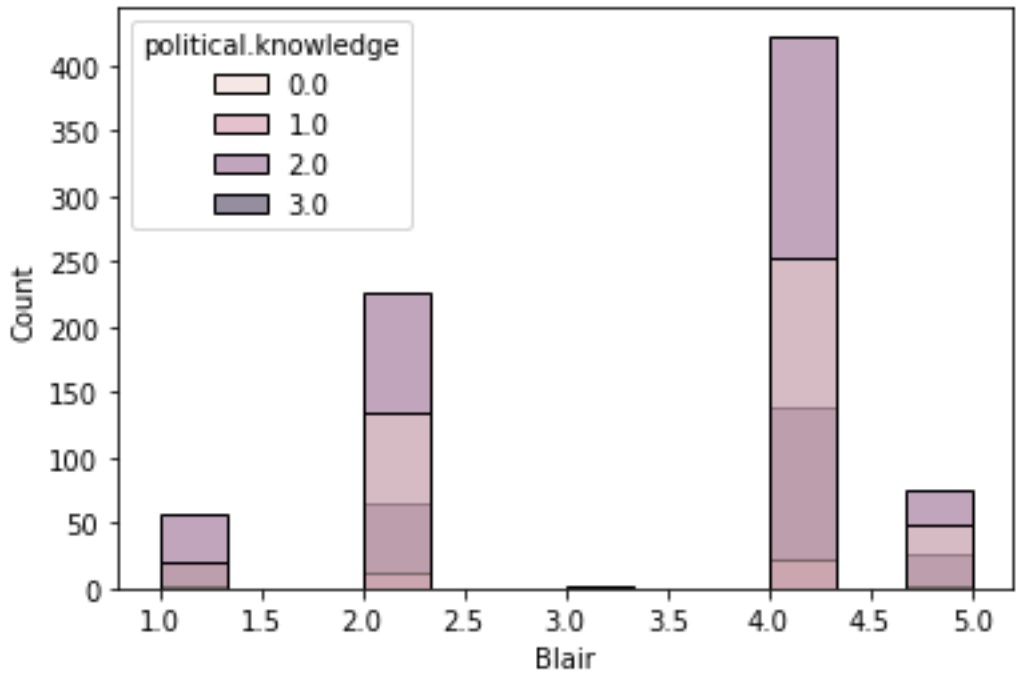


Image:13

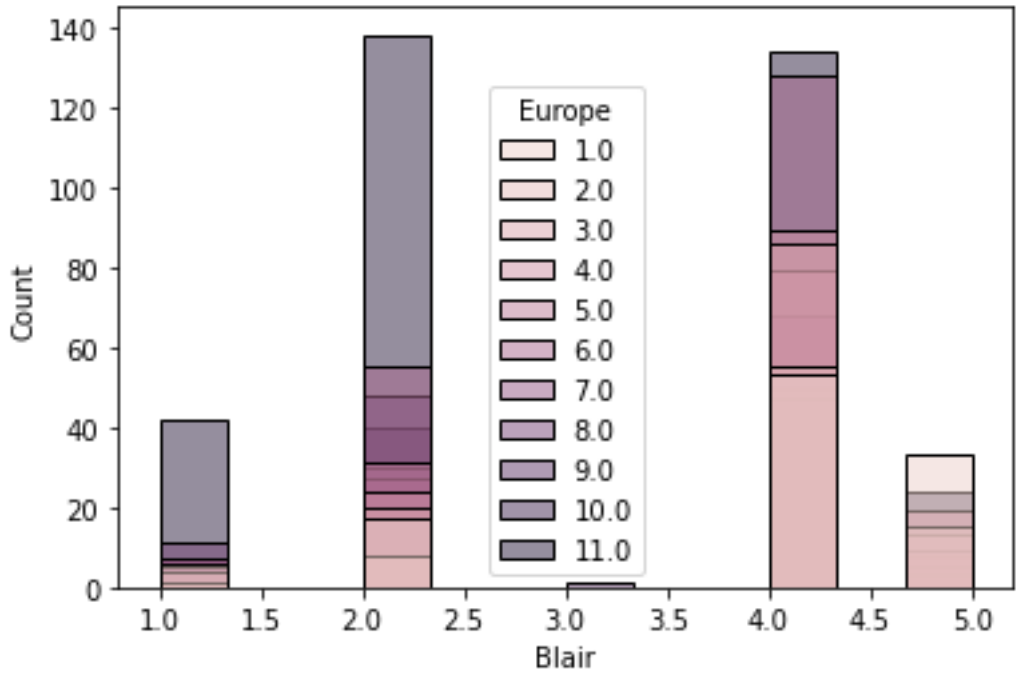


Image: 14

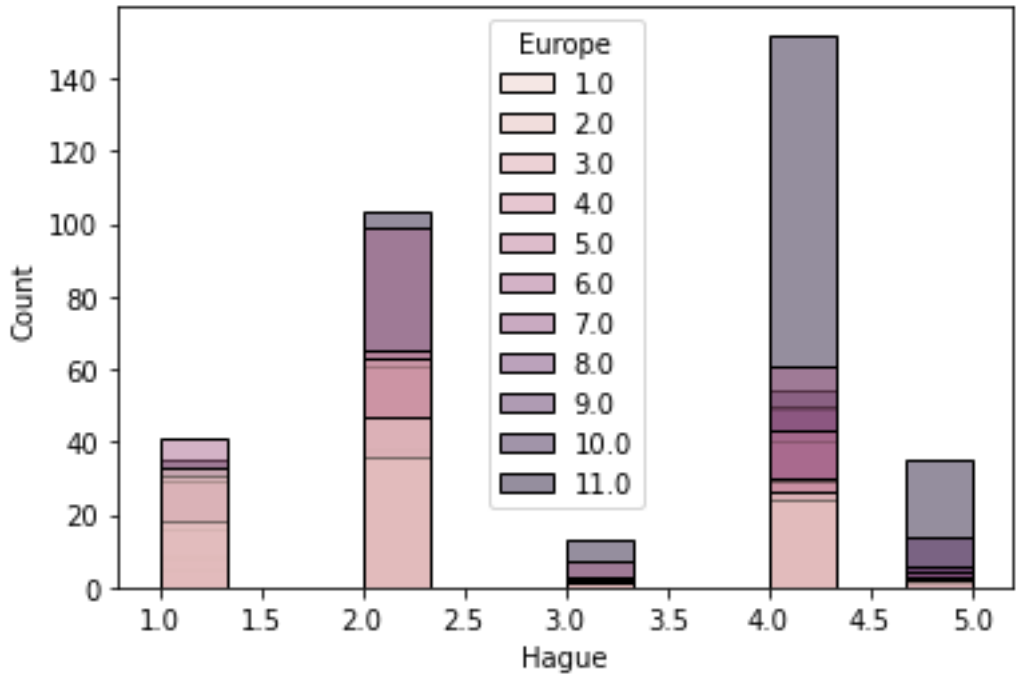


Image :15

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) . The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for example std dev, variance, etc. Should justify whether there is a necessity for scaling. Object data should be converted into categorical/numerical data to fit in the models. (pd.categorical().codes(), pd.get_dummies (drop_first=True)) Data split, ratio defined for the split, train-test split should be discussed.

We need to do Scaling only when the data points are far from each other,In Our Case , Only “Age” variable is distance from Others, So it doesn’t /Impact Much , even when we don’t do Scaling.

So , we are not doing Scaling.

After changing variables which are object datatype, into Categorical/Integer type . We are doing this so that, the data can fit into Machine Learning Models.

vote category
age float64
economic.cond.national float64
economic.cond.household float64
Blair float64
Hague float64
Europe float64
political.knowledge float64
gender int32

Headset of the Dataset:after changing the varable datatypes:

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

Table 5
After creating Dummy : Head of the Dataset.

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender	vote_Labour
1	43.0	3.0	3.0	4.0	1.0	2.0	2.0	0	1
2	36.0	4.0	4.0	4.0	4.0	5.0	2.0	1	1
3	35.0	4.0	4.0	5.0	2.0	3.0	2.0	1	1
4	24.0	4.0	2.0	2.0	1.0	4.0	0.0	0	1
5	41.0	2.0	2.0	1.0	1.0	6.0	2.0	1	1

Target Variable: Vote_ Value counts (for training dataset)

Labour 735
Conservative 332

Target Variable: Vote_ Value counts (for testing dataset)

Labour 328
Conservative 130

1.4) Apply Logistic Regression and LDA (Linear Discriminant Analysis) (2 pts). Interpret the inferences of both model s (2 pts). Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)

	0	1
0	0.921946	0.078054
1	0.690526	0.309474
2	0.346669	0.653331
3	0.488887	0.511113
4	0.158897	0.841103

Table 6

Confusion Matrix:

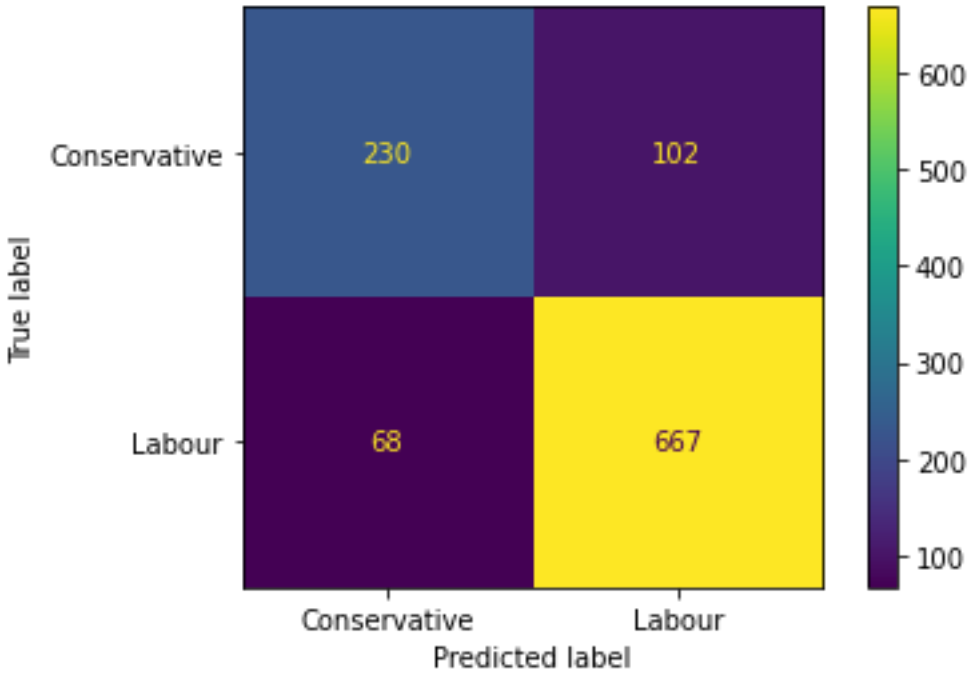


Image 16

Classification Report: Train data

	precision	recall	f1-score	support
0	0.87	0.91	0.89	735
1	0.77	0.69	0.73	332
accuracy			0.84	1067
macro avg	0.82	0.80	0.81	1067
weighted avg	0.84	0.84	0.84	1067

Classification Report: Test data

	precision	recall	f1-score	support
0	0.87	0.89	0.88	328
1	0.70	0.65	0.68	130
accuracy			0.82	458
macro avg	0.78	0.77	0.78	458
weighted avg	0.82	0.82	0.82	458

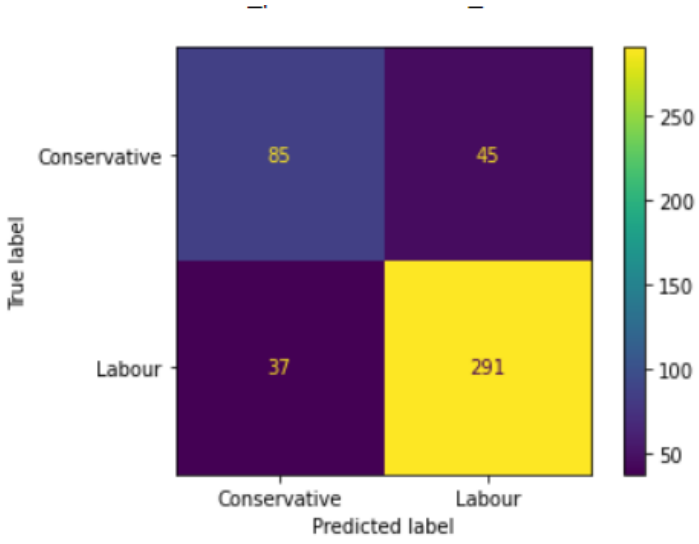


Image 17

Logistic Regression : Train Dataset:

Accuracy : 0.889
Precision : 0.87

Recall: 0.91

F1 Score: 0.89

Test Data:

Accuracy: 0.883

Precision : 0.87

Recall: 0.89

F1 Score: 0.88

The model is nor over fit or underfit.

Linear Discriminant Analysis

Trained Data

	precision	recall	f1-score	support
0	0.90	0.87	0.88	759
1	0.70	0.76	0.73	308
accuracy			0.84	1067
macro avg	0.80	0.81	0.81	1067
weighted avg	0.84	0.84	0.84	1067

Test Data

	precision	recall	f1-score	support
0	0.88	0.87	0.87	333
1	0.66	0.69	0.67	125
accuracy			0.82	458
macro avg	0.77	0.78	0.77	458

weighted avg	0.82	0.82	0.82	458

Linear Discriminant Analysis

Train data:

- Accuracy: 89%
- Precision: 90%
- Recall: 87%
- F1-Score: 88%

Test data:

- Accuracy: 88%
- Precision: 88%
- Recall: 87%
- F1-Score: 87%

Validation:

The model is not overfit or Underfit

.

1.5) Apply KNN Model and Naïve Bayes Model Interpret the inferences of each model Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Calculate Train and Test Accuracies for each model. Comment on the validness of models (over fitting or under fitting)

Classification Repot of Naïve Base Model:

Train Data

	precision	recall	f1-score	support
0	0.88	0.88	0.88	741
1	0.72	0.74	0.73	326
accuracy			0.83	1067
macro avg	0.80	0.81	0.80	1067
weighted avg	0.83	0.83	0.83	1067

Test Data

	precision	recall	f1-score	support
0	0.88	0.88	0.88	741
1	0.72	0.74	0.73	326
accuracy			0.83	1067
macro avg	0.80	0.81	0.80	1067
weighted avg	0.83	0.83	0.83	1067

Naïve Base:

Train Dataset:

Accuracy : 89%
Precision : 87 %

Recall: 88%

F1 Score: 88%

Test Data:

Accuracy: 88%

Precision : 80%

Recall: 81%

F1 Score: 80%

Validation:

The model is nor over fit or underfit.

KNN Model:

Classification report on Train Data

	precision	recall	f1-score	support
0	0.92	0.89	0.90	759
1	0.74	0.80	0.77	308
accuracy			0.86	1067
macro avg	0.83	0.84	0.83	1067
weighted avg	0.87	0.86	0.86	1067

Classification report on TestData

	precision	recall	f1-score	support
0	0.85	0.85	0.85	328
1	0.62	0.62	0.62	130
accuracy			0.79	458
macro avg	0.74	0.74	0.74	458
weighted avg	0.79	0.79	0.79	458

KNN : Train Dataset:

Accuracy : 92%
Precision : 92%

Recall: 89%

F1 Score: 90%

Test Data:

Accuracy: 83%

Precision : 85%

Recall: 85%

F1 Score: 85%

Validation:

The model is nor over fit or underfit.

1.6) Model Tuning , Bagging and Boosting .Apply grid search on each model (include all models) and make models on best_params. Define a logic behind choosing particular values for different hyper-parameters for grid search. Compare and comment on performances of all. Comment on feature importance if applicable. Successful implementation of both algorithms along with inferences and comments on the model performances.

```
GridSearch CV(cv=10, estimator=RandomForest Classifier(),
        param_grid={'max_depth': [5, 10, 15, 20],
                'min_samples_leaf': [15, 25, 35, 50],
                'min_samples_split': [30, 50, 70, 100],
                'random_state': [0]})

GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
        param_grid={'max_depth': [5, 10, 15, 20],
                'min_samples_leaf': [15, 25, 35, 50],
                'min_samples_split': [30, 50, 70, 100],
                'random_state': [0]})

GridSearchCV(estimator=AdaBoostClassifier(),
        param_grid={'algorithm': ['SAMME', 'SAMME.R'],
                'learning_rate': [0.1, 0.01, 0.001],
                'n_estimators': [100, 500, 1000]})
```

Logistic Regression

Classification Report
Training Dataset

	precision	recall	f1-score	support
0	0.87	0.91	0.89	735
1	0.77	0.69	0.73	332
accuracy			0.84	1067
macro avg	0.82	0.80	0.81	1067
weighted avg	0.84	0.84	0.84	1067

Classification Report
Testing Dataset

	precision	recall	f1-score	support
0	0.87	0.89	0.88	328
1	0.70	0.65	0.68	130
accuracy			0.82	458
macro avg	0.78	0.77	0.78	458
weighted avg	0.82	0.82	0.82	458

Accuracy for Training dataset 89%

Accuracy for Test dataset 88%

This model is valid, because this model is not underfit/overfit.

Linear Discriminant Analysis

Classification report for Training dataset:

	precision	recall	f1-score	support
0	0.90	0.87	0.88	759
1	0.70	0.76	0.73	308
accuracy			0.84	1067
macro avg	0.80	0.81	0.81	1067
weighted avg	0.84	0.84	0.84	1067

Classification report for Testing dataset:

	precision	recall	f1-score	support
0	0.88	0.87	0.87	333
1	0.66	0.69	0.67	125
accuracy			0.82	458
macro avg	0.77	0.78	0.77	458
weighted avg	0.82	0.82	0.82	458

Accuracy for Training Dataset:88.9%

Accuracy for Testing Dataset:88.4%

Naïve_Baye's

Naïve_Bayes Classification Report
Training Dataset

	precision	recall	f1-score	support
0	0.88	0.88	0.88	741
1	0.72	0.74	0.73	326
accuracy			0.83	1067
macro avg	0.80	0.81	0.80	1067
weighted avg	0.83	0.83	0.83	1067

Naïve_Bayes Classification Report
Testing Dataset

	precision	recall	f1-score	support
0	0.87	0.89	0.88	320
1	0.72	0.68	0.70	138
accuracy			0.83	458
macro avg	0.79	0.78	0.79	458
weighted avg	0.82	0.83	0.82	458

Accuracy for Training Dataset:88.6%

Accuracy for Testing Dataset:88.5%

KNN

Classification Report for Training Dataset:

	precision	recall	f1-score	support
0	0.92	0.89	0.90	759
1	0.74	0.80	0.77	308
accuracy			0.86	1067
macro avg	0.83	0.84	0.83	1067
weighted avg	0.87	0.86	0.86	1067

Classification Report for Test Dataset:

	precision	recall	f1-score	support
0	0.85	0.85	0.85	328
1	0.62	0.62	0.62	130
accuracy			0.79	458
macro avg	0.74	0.74	0.74	458
weighted avg	0.79	0.79	0.79	458

Accuracy for Training Dataset:93%

Accuracy for Testing Dataset:83%

Validation: This is a valid model, as the model is not overfit nor underfit

AdaBoost Classifier

Classification Report for Trained Set

	precision	recall	f1-score	support
0	0.85	0.92	0.88	735
1	0.78	0.65	0.71	332
accuracy			0.84	1067
macro avg	0.82	0.79	0.80	1067
weighted avg	0.83	0.84	0.83	1067

Classification Report for Test Set

	precision	recall	f1-score	support
0	0.87	0.90	0.88	328
1	0.73	0.65	0.69	130
accuracy			0.83	458
macro avg	0.80	0.78	0.79	458
weighted avg	0.83	0.83	0.83	458

Accuracy for Training dataset 90%

Accuracy for Test dataset 89%

This model is valid, because this model is not underfit/overfit.

Decision Tree:

Classification Report for Train Data:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	736
1	1.00	1.00	1.00	331
accuracy			1.00	1067
macro avg	1.00	1.00	1.00	1067
weighted avg	1.00	1.00	1.00	1067

Classification Report for Test Data:

	precision	recall	f1-score	support
0	0.81	0.84	0.83	314
1	0.62	0.56	0.59	144
accuracy			0.76	458
macro avg	0.72	0.70	0.71	458
weighted avg	0.75	0.76	0.75	458

Accuracy for Train Dataset 100%

Accuracy for Test Data 71%

Validation: This model is Overfit, so we are tuning the model for better results.

DecisionTree with Tuning

After Tuning the Model with Hyper Parameters:

Classification Report for Train Data

	precision	recall	f1-score	support
0	1.00	1.00	1.00	735
1	1.00	1.00	1.00	332
accuracy			1.00	1067
macro avg	1.00	1.00	1.00	1067
weighted avg	1.00	1.00	1.00	1067

Classification Report for Test Data

	precision	recall	f1-score	support
0	0.87	0.87	0.87	328
1	0.67	0.68	0.67	130
accuracy			0.81	458
macro avg	0.77	0.77	0.77	458
weighted avg	0.81	0.81	0.81	458

Accuracy for Train Dataset: 89%

Accuracy for Test Dataset: 88%

RandomForest:

Classification Report for Train Data

	precision	recall	f1-score	support
0	1.00	1.00	1.00	736
1	1.00	1.00	1.00	331
accuracy			1.00	1067
macro avg	1.00	1.00	1.00	1067
weighted avg	1.00	1.00	1.00	1067

Classification Report for Test Data

	precision	recall	f1-score	support
0	0.88	0.88	0.88	331
1	0.68	0.70	0.69	127
accuracy			0.83	458
macro avg	0.78	0.79	0.79	458
weighted avg	0.83	0.83	0.83	458

Accuracy for Train dataset: 100%

Accuracy for TestDataset : 88%

Validation: The model is Overfit , so we are tuning the model with Hyper Parameters.

Random Forest with Tuning /Hyper Parameters

Classification Report for Train Data

	precision	recall	f1-score	support
0	0.85	0.93	0.89	735
1	0.81	0.62	0.71	332
accuracy			0.84	1067
macro avg	0.83	0.78	0.80	1067
weighted avg	0.84	0.84	0.83	1067

Classification Report for Test Data

	precision	recall	f1-score	support
0	0.86	0.91	0.89	328
1	0.75	0.63	0.68	130
accuracy			0.83	458
macro avg	0.80	0.77	0.79	458
weighted avg	0.83	0.83	0.83	458

Accuracy for Training Dataset: 91%

Accuracy for Test Dataset:89%

Validation: The model is not Overfit/Underfit. It's a perfect model.

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, classification report . Final Model - Compare and comment on all models on the basis of the performance metrics in a structured tabular manner. Describe on which model is best/optimized, After comparison which model suits the best for the problem in hand on the basis of different measures. Comment on the final model.

Logistic Regression:

Confusion Matrix:

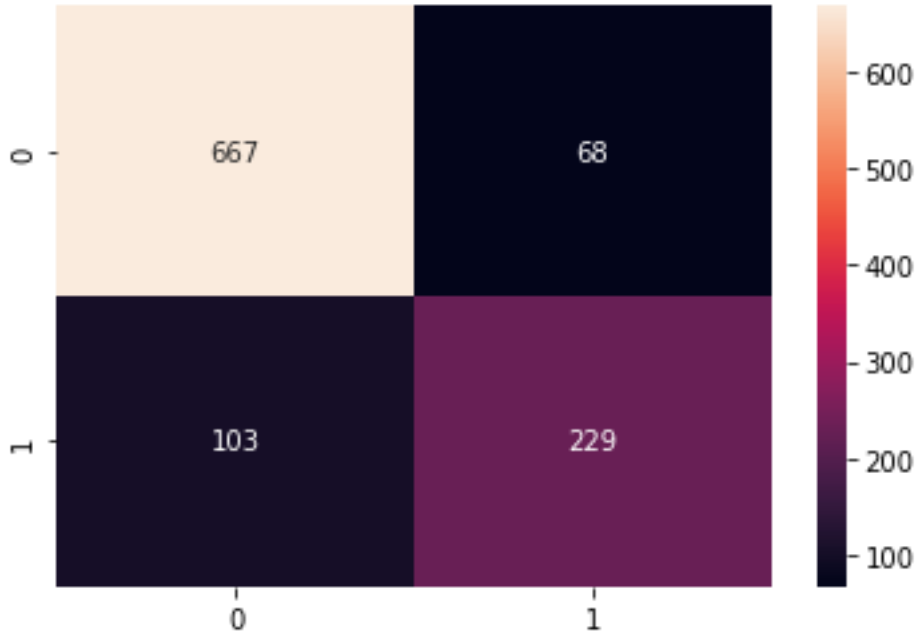


Image-18

Accuracy for Training dataset 89%

Accuracy for Test dataset 88%

Auc &Roc Curve for Train Dataset

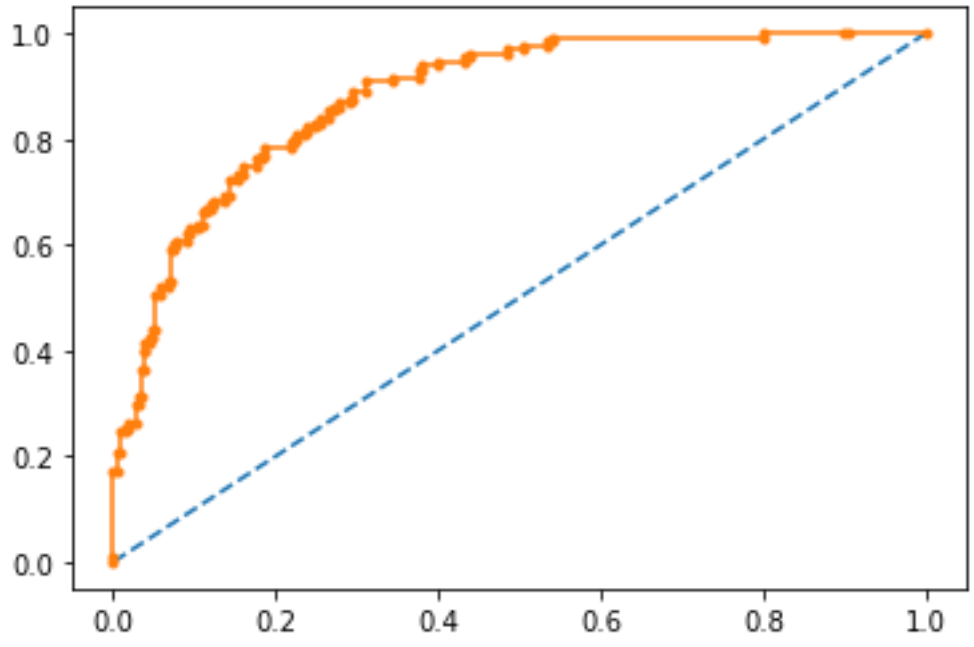


Image 19

Auc &Roc Curve for Test Dataset

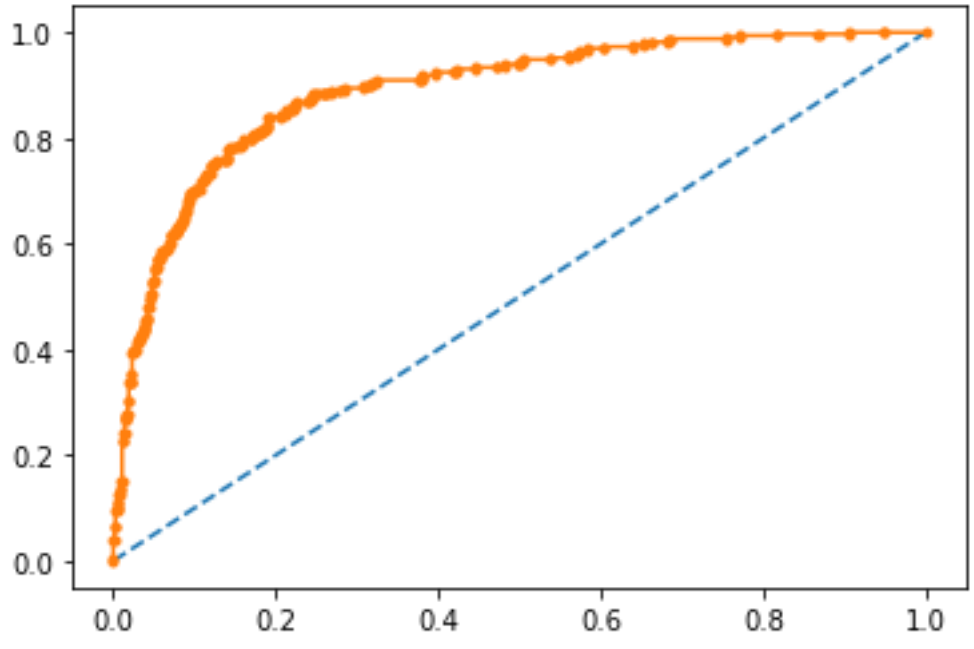


Image 20

Logistic Regression : Train Dataset:

Accuracy : 0.889
Precision : 0.87

Recall: 0.91

F1 Score: 0.89

Test Data:

Accuracy: 0.883

Precision : 0.87

Recall: 0.89

F1 Score: 0.88

The model is nor over fit or underfit.

LDA

Confusion Matrix

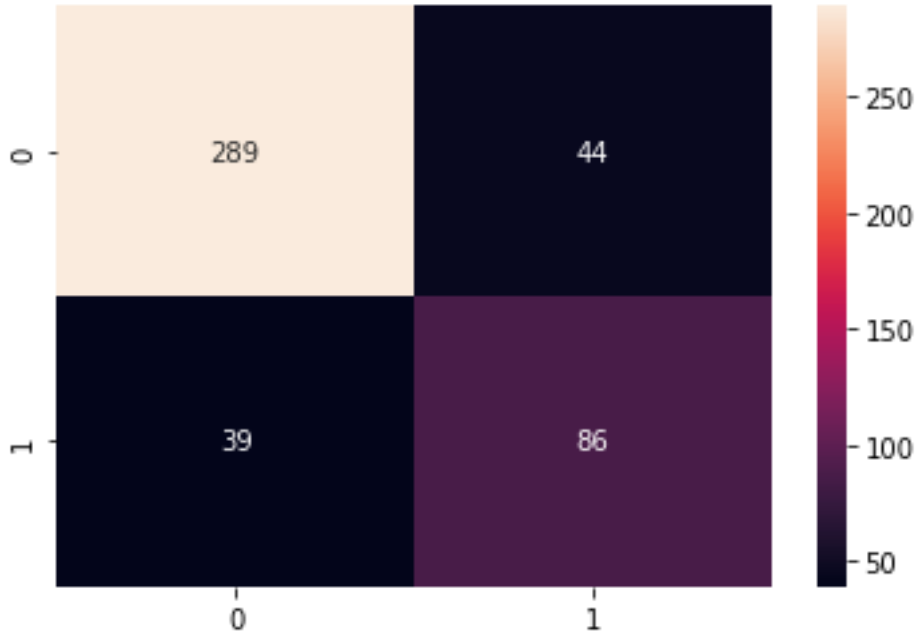


Image 21

Linear Discriminant Analysis

Train data:

- Accuracy: 89%
- Precision: 90%
- Recall: 87%
- F1-Score: 88%

Test data:

- Accuracy: 88%
- Precision: 88%
- Recall: 87%
- F1-Score: 87%

Validation:

The model is not overfit or Underfit

AUC & ROC Curve for Train Dataset:

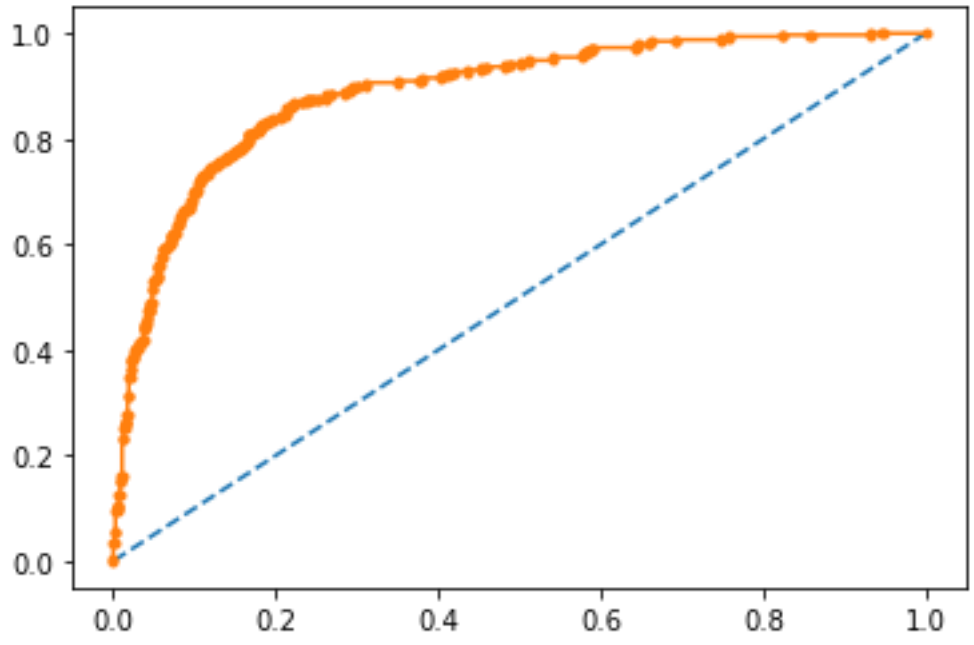


Image 22

AUC & ROC Curve for Test Dataset:

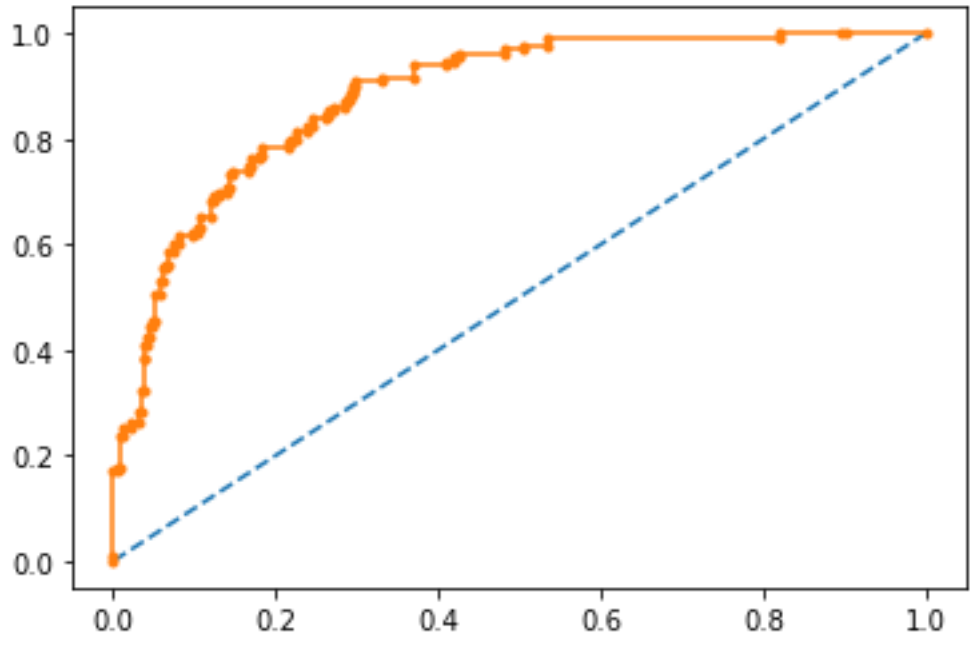


Image 23

Naïve-Baye's :

Confusion Matrix for Trained Data

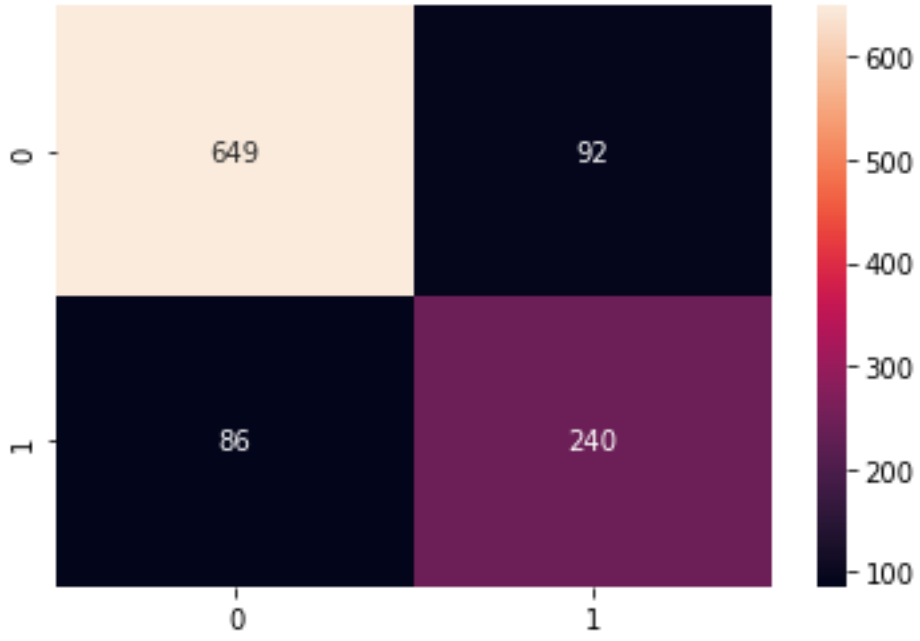


Image 24

Confusion Matrix for Test Dataset:

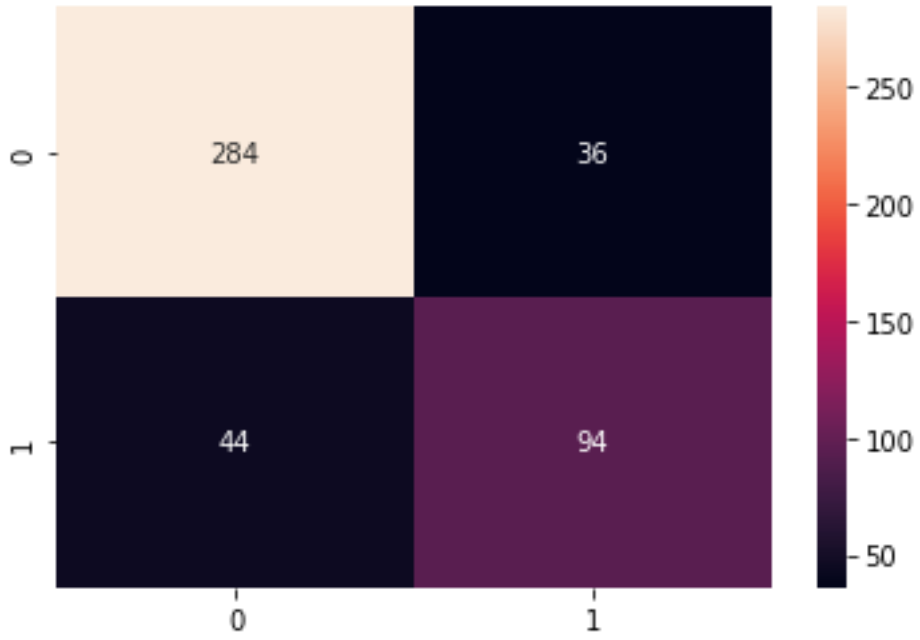


Image 25

AUC_ROC Curve: Train Dataset

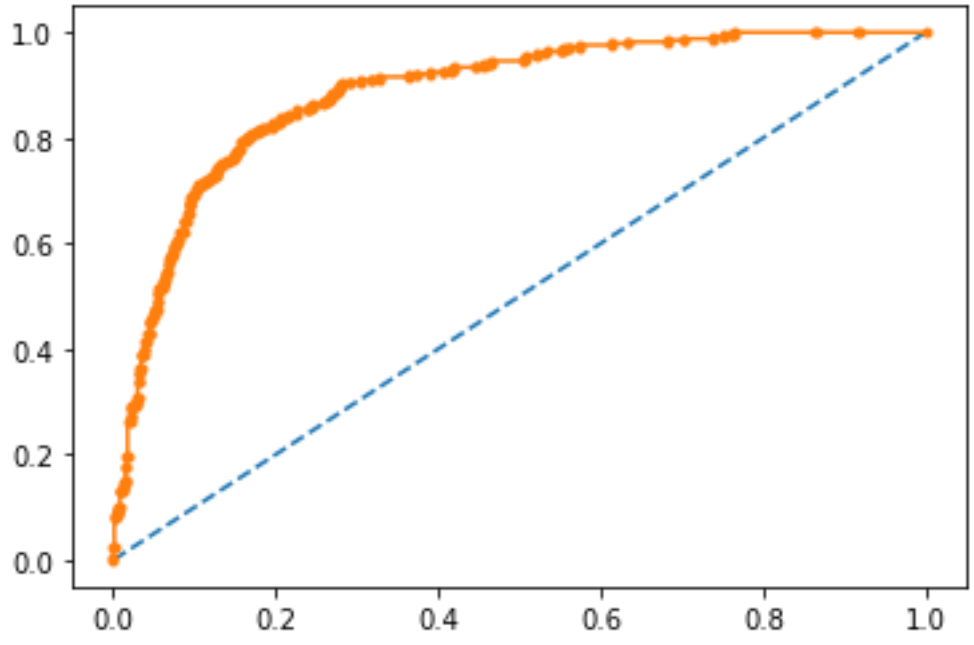


Image 26

AUC_ROC Curve for Test Dataset

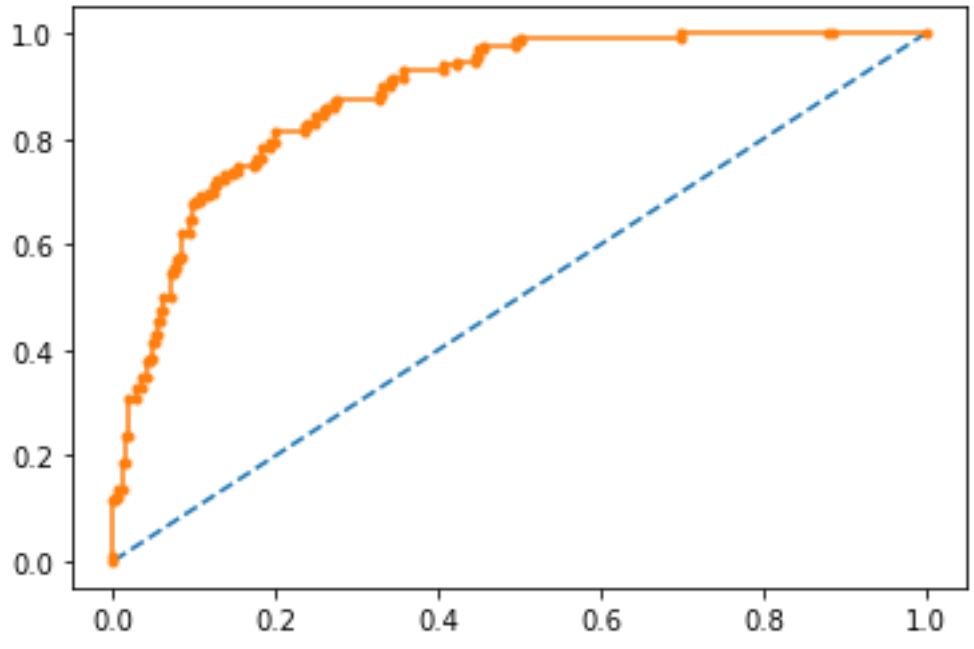


Image 27

Train Dataset:

Accuracy : 89%
Precision : 87 %

Recall: 88%

F1 Score: 88%

Test Data:

Accuracy: 88%

Precision : 80%

Recall: 81%

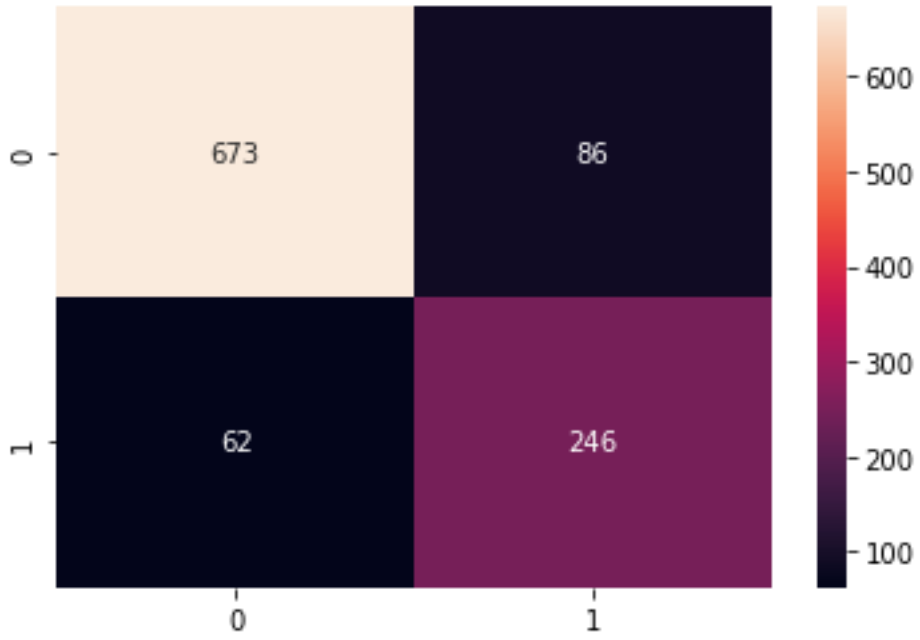
F1 Score: 80%

Validation:

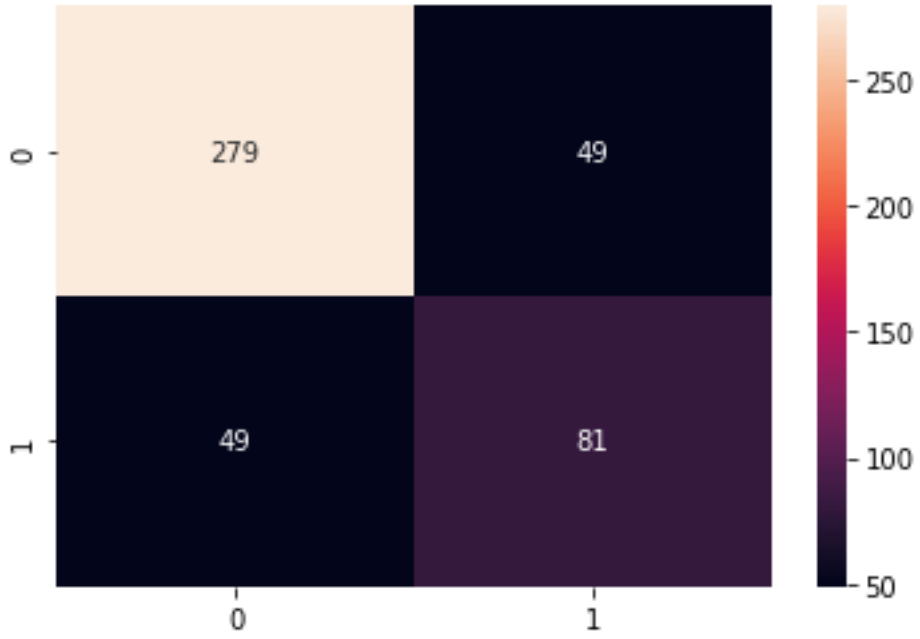
The model is nor over fit or underfit.

KNN_Model

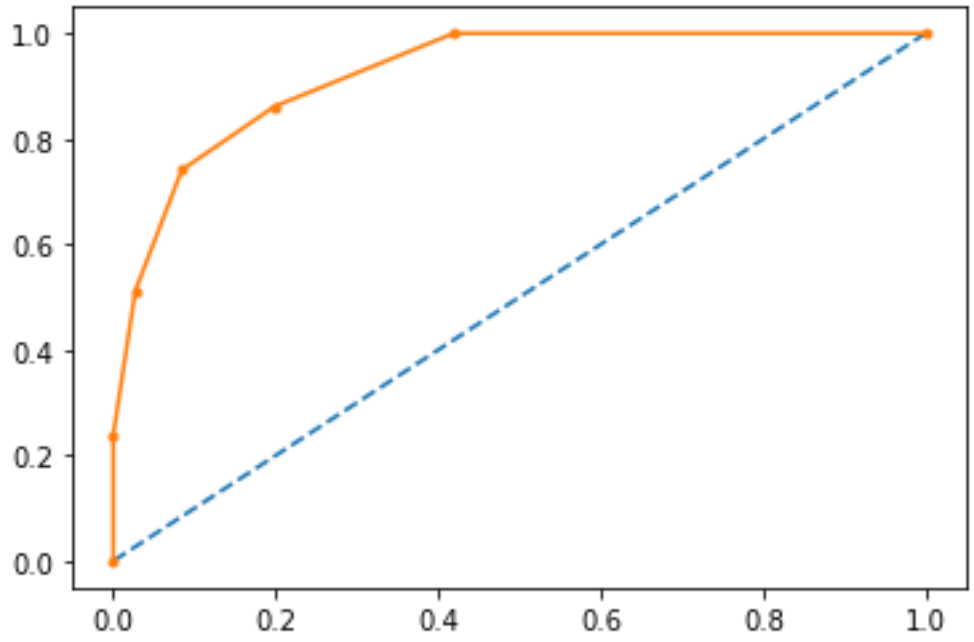
Confusion Matrix for Trained Dataset



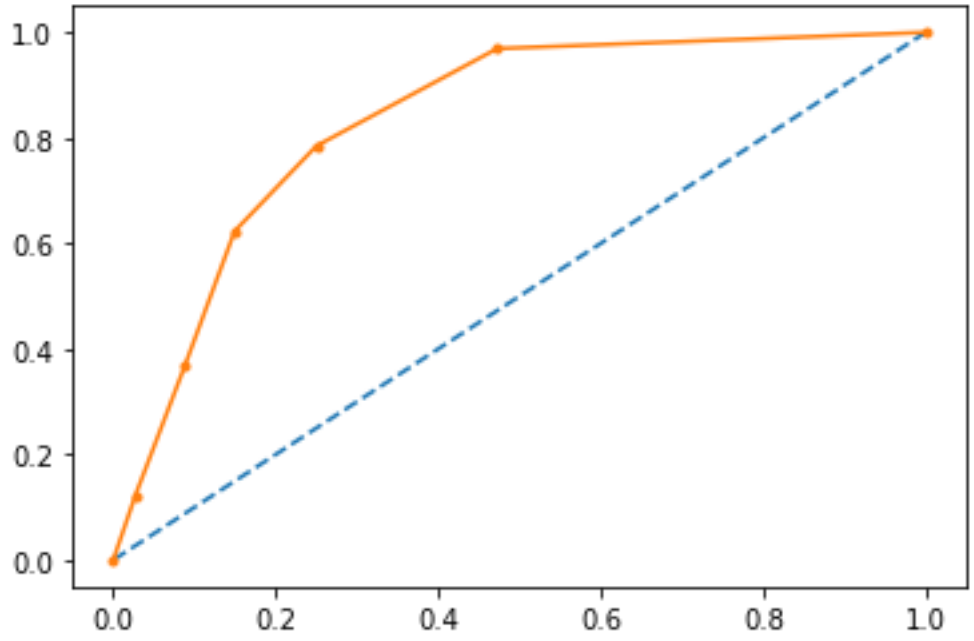
Confusion Matrix for Test Dataset



AUC_Roc Curve for Training Dataset



AUC_Roc Curve for Test Dataset



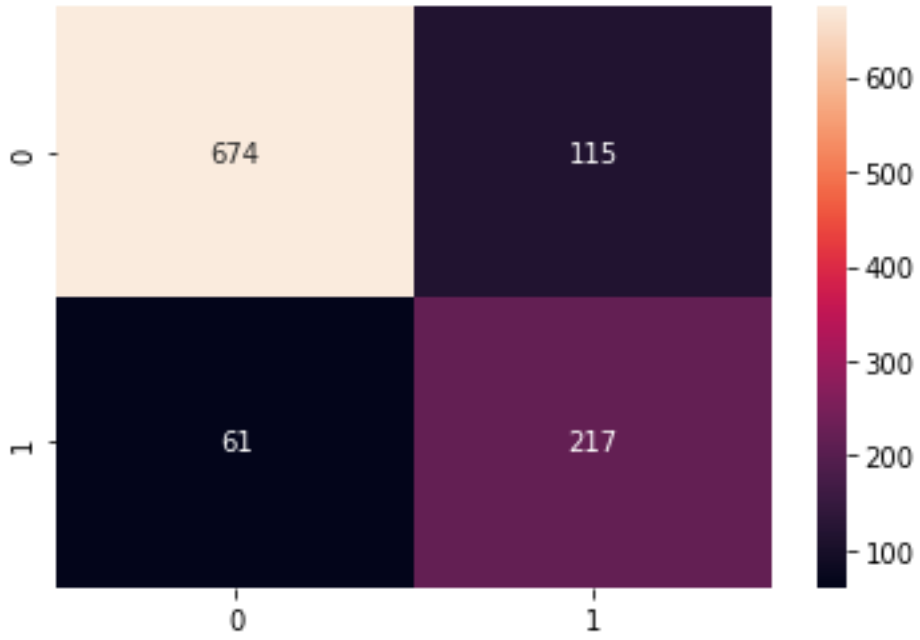
Accuracy for TrainDataset 93%

Accuracy for Test Dataset 82%

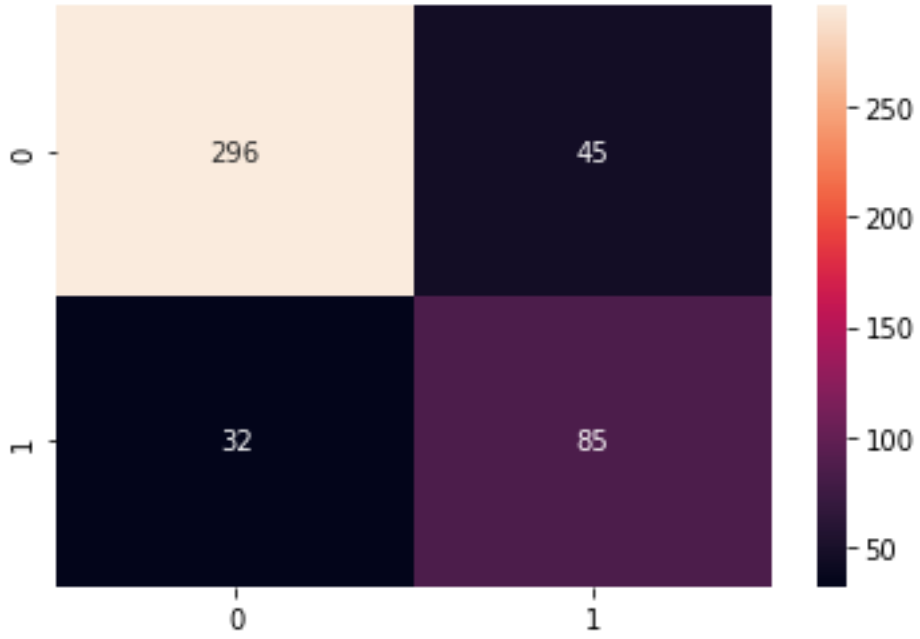
Validation: The model is not Overfit nor underfit.

ADABoost Classifier

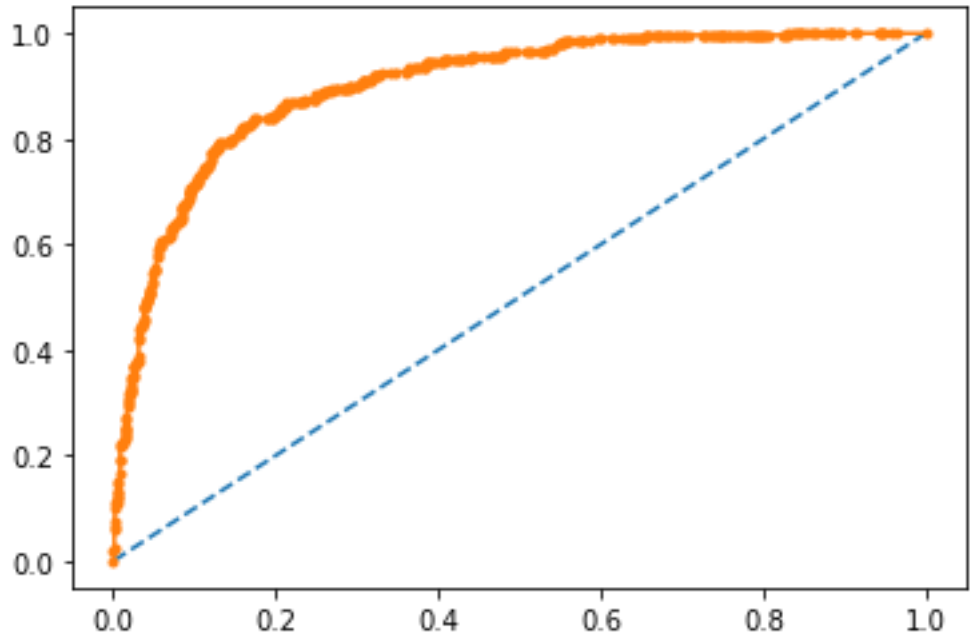
Confusion Matrix for Trained Dataset:



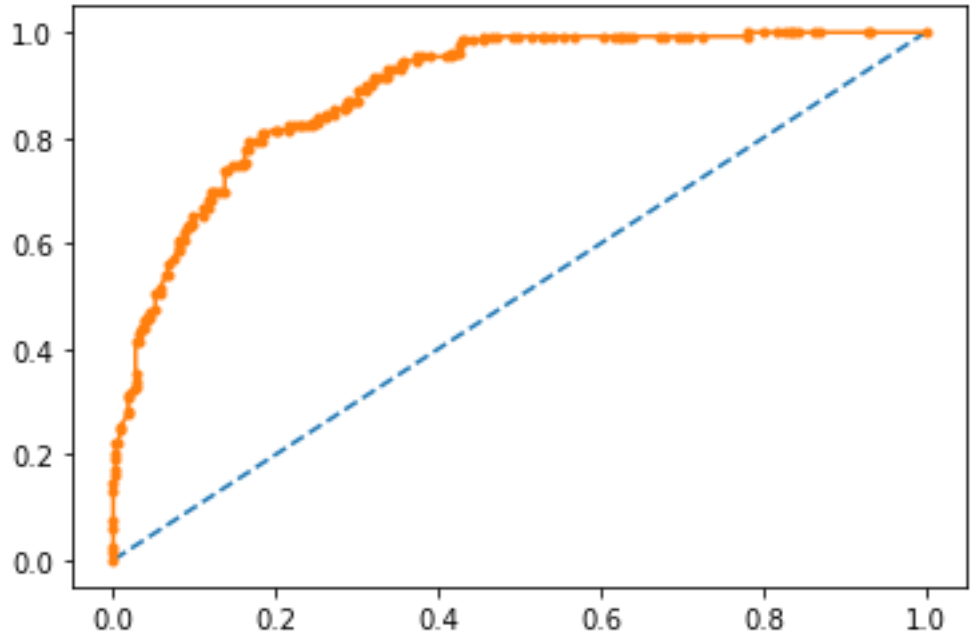
Confusion Matrix for Test Dataset:



AUC_ROC Curve for Trained Dataset



AUC_ROC Curve for Test Dataset



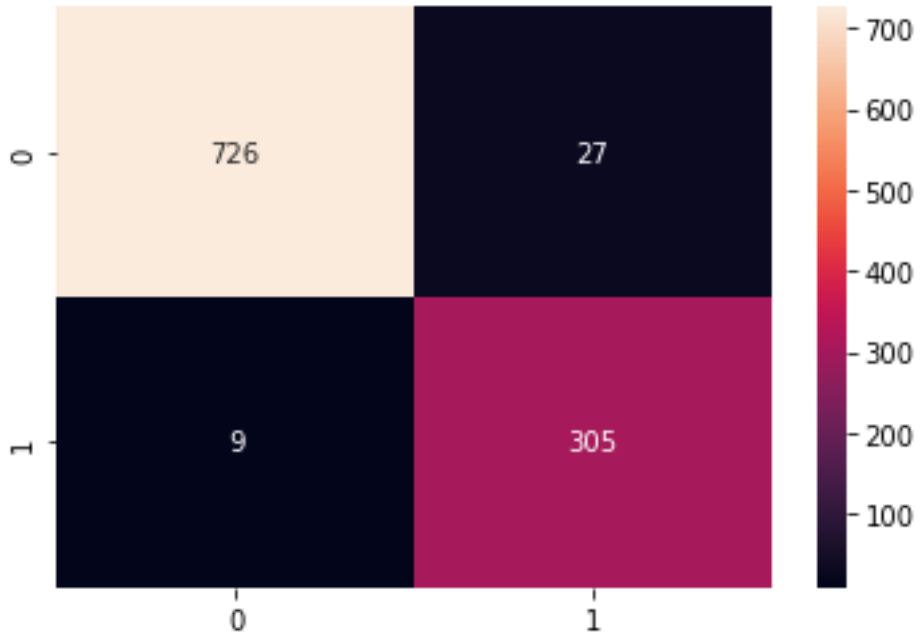
Accuracy for TrainDataset 90%

Accuracy for Test Dataset 89%

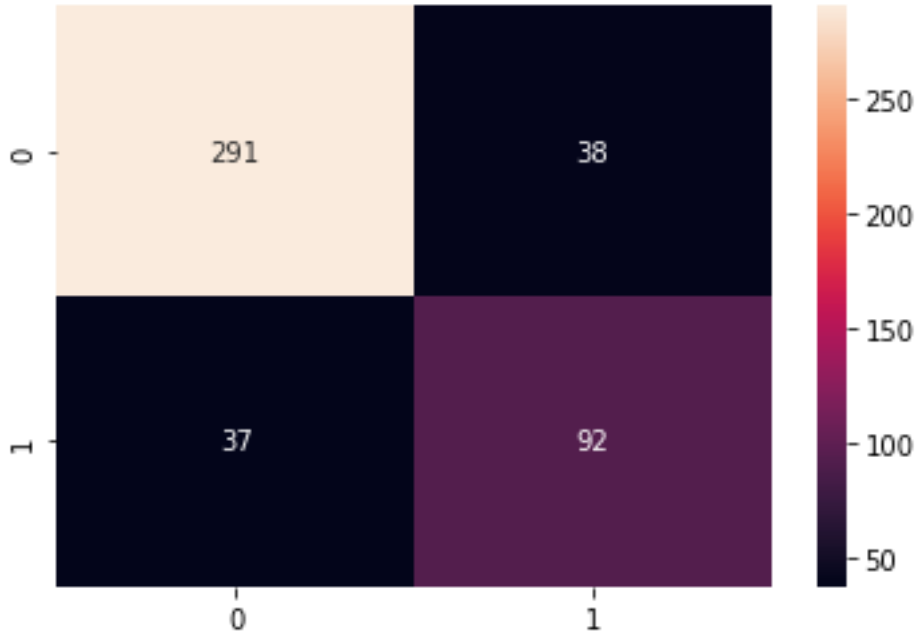
Validation: The model is not Overfit nor underfit.

Random Forest :

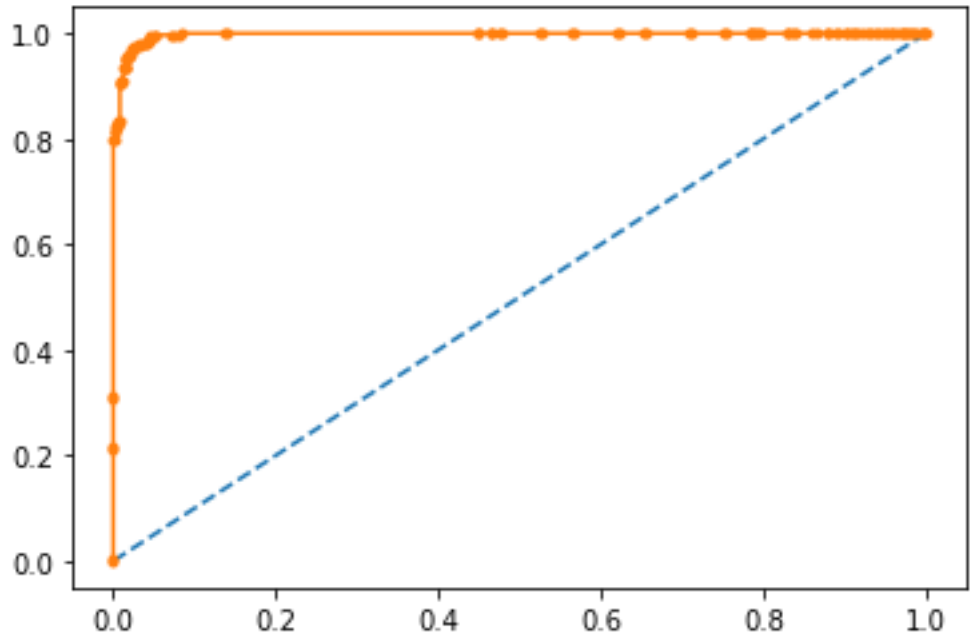
Confusion Matrix with Trained Dataset



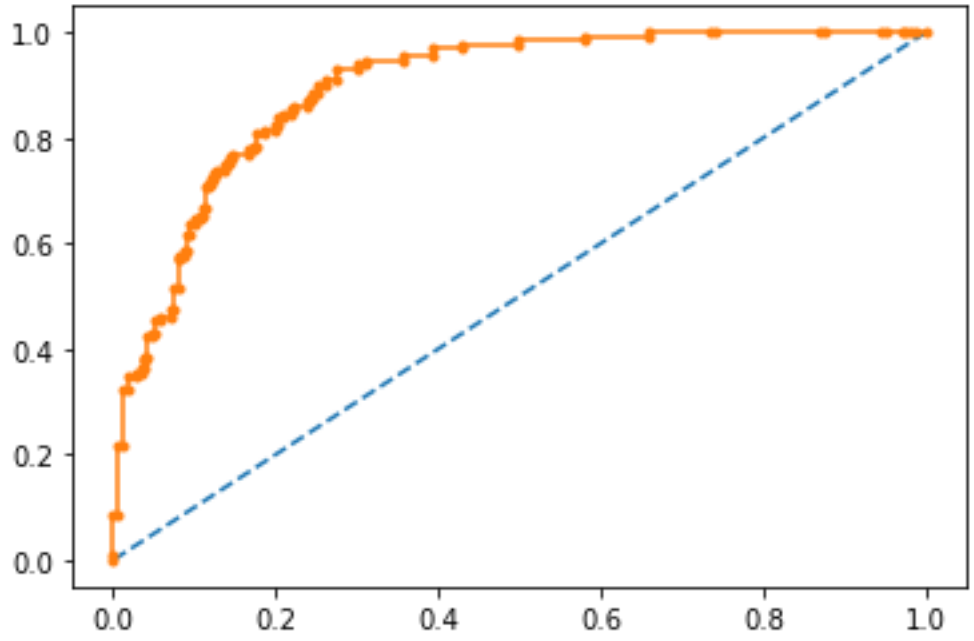
Confusion Matrix with Test Dataset



AUC-ROC Curve for Trained Dataset:



AUC-ROC Curve Test Dataset



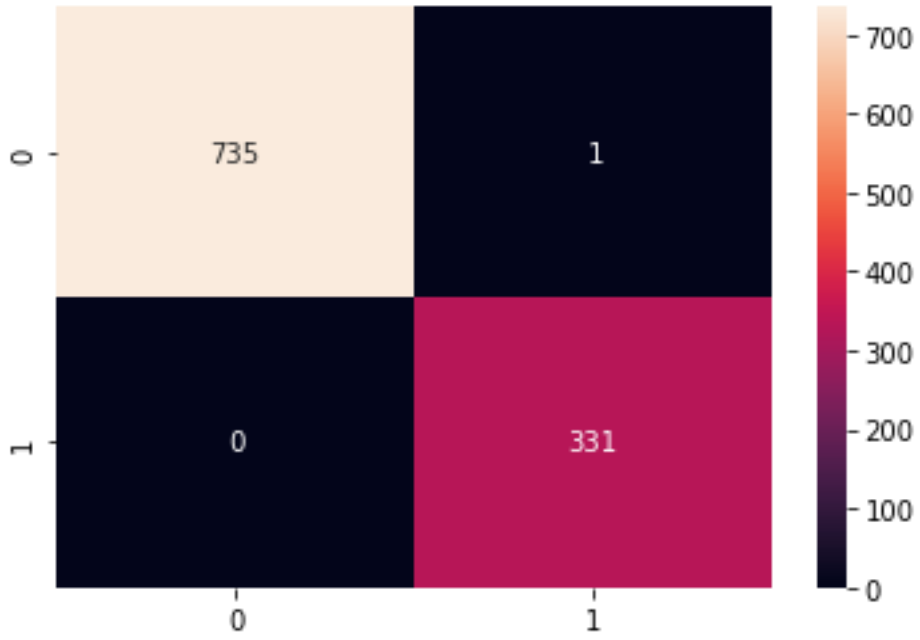
Accuracy for Train Dataset: 99%

Accuracy for Test Dataset: 89%

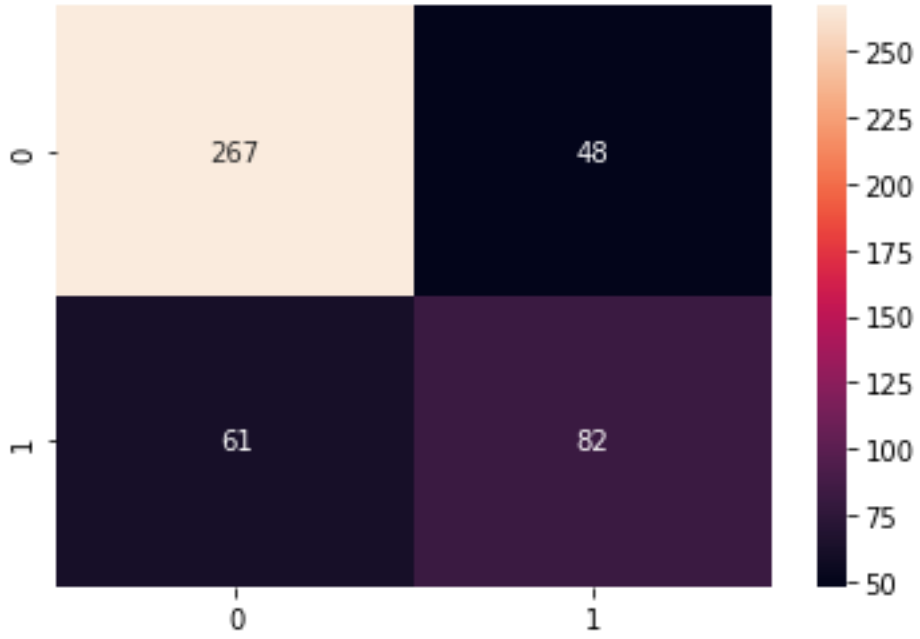
Validation: The model is not overfit nor underfit.

Decision_Tree:

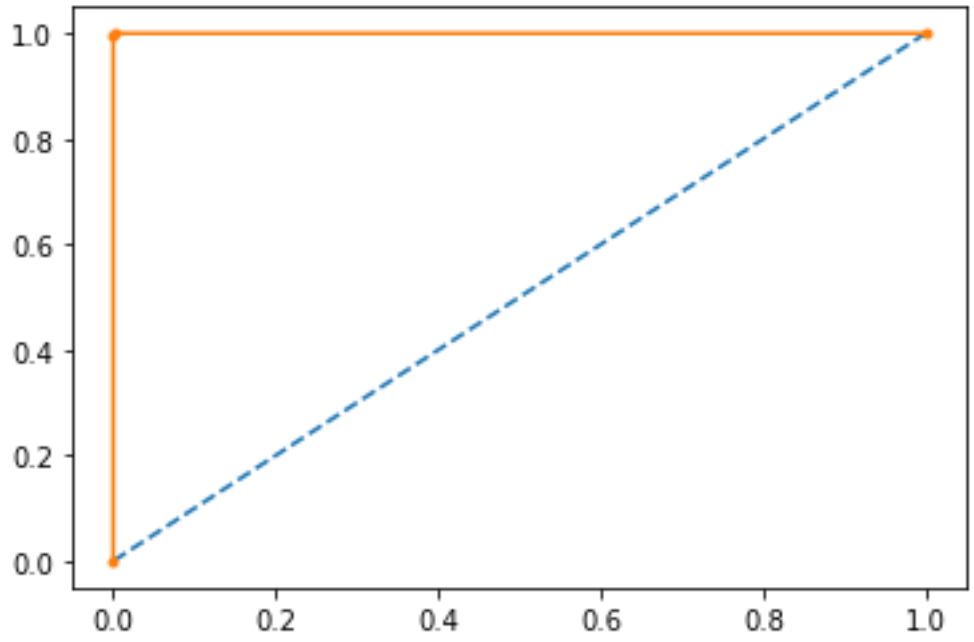
Confusion Matrix for Train Dataset:



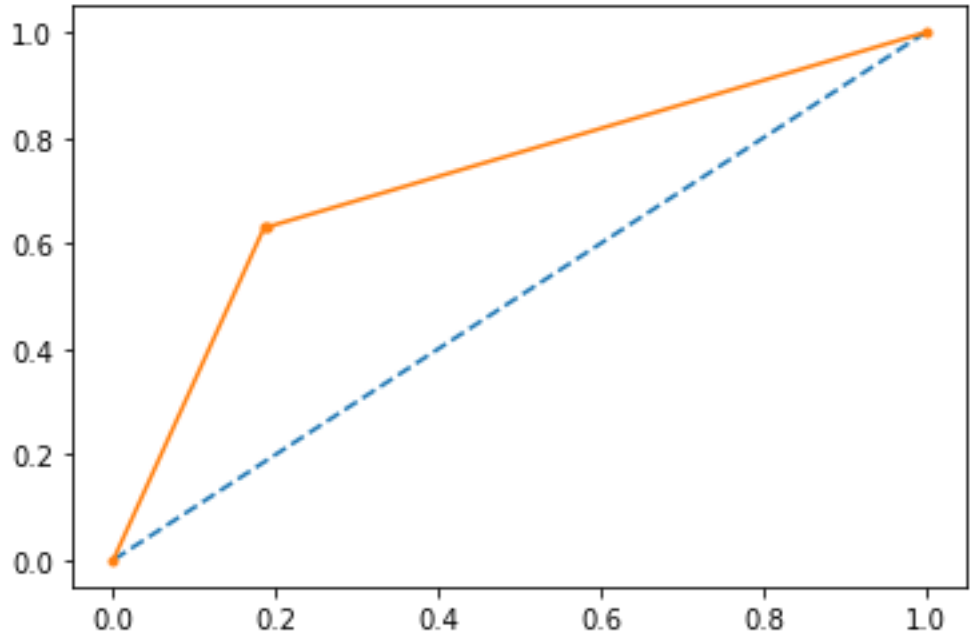
Confusion Matrix for Test Dataset:



AUC-ROC Curve: Train Dataset



AUC-ROC Curve: Test Dataset



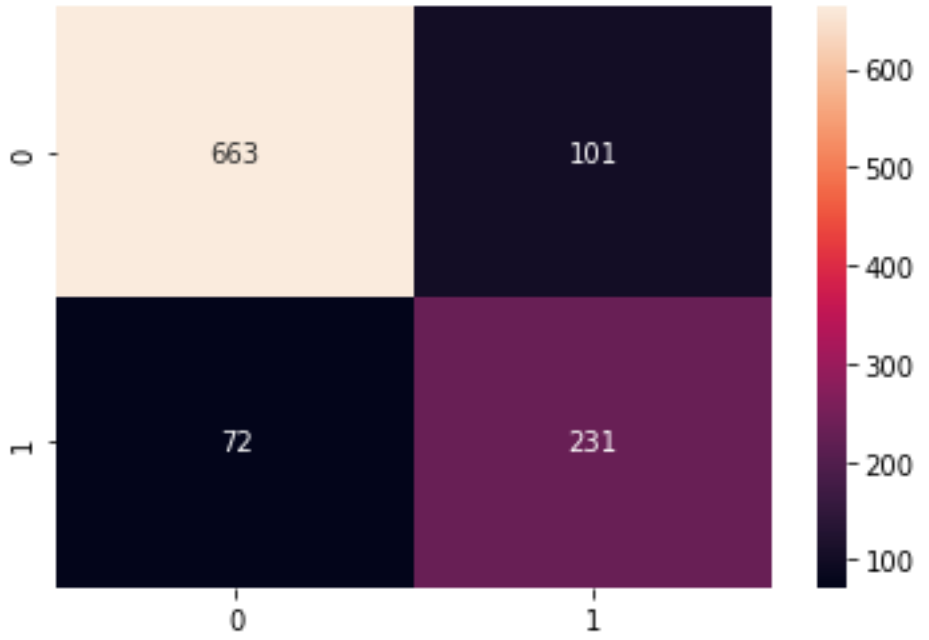
Accuracy for Trained DataSet: 100%

Accuracy for Test Dataset:72%

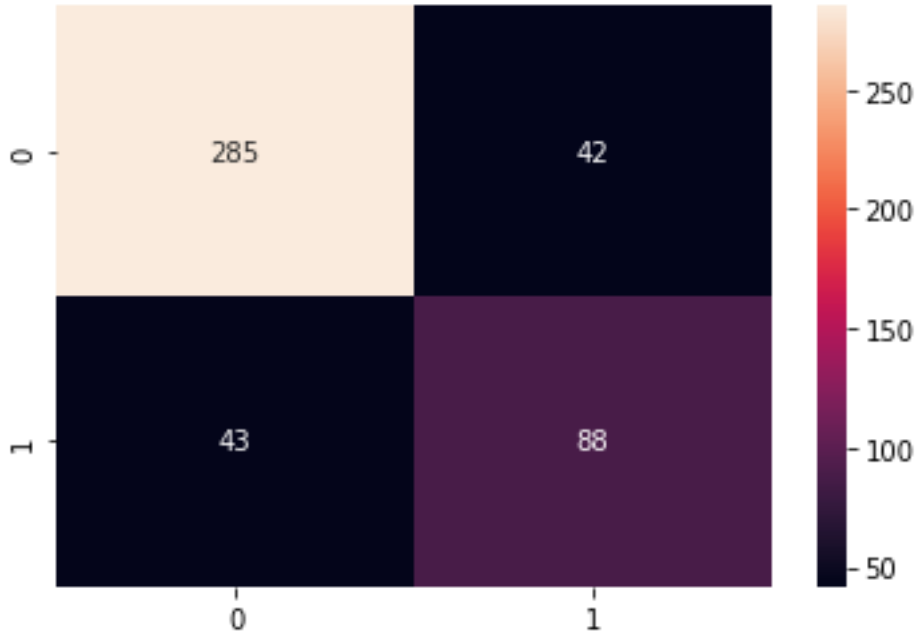
Validation: Model is Overfit

DecisionTree with hyper Parameters:

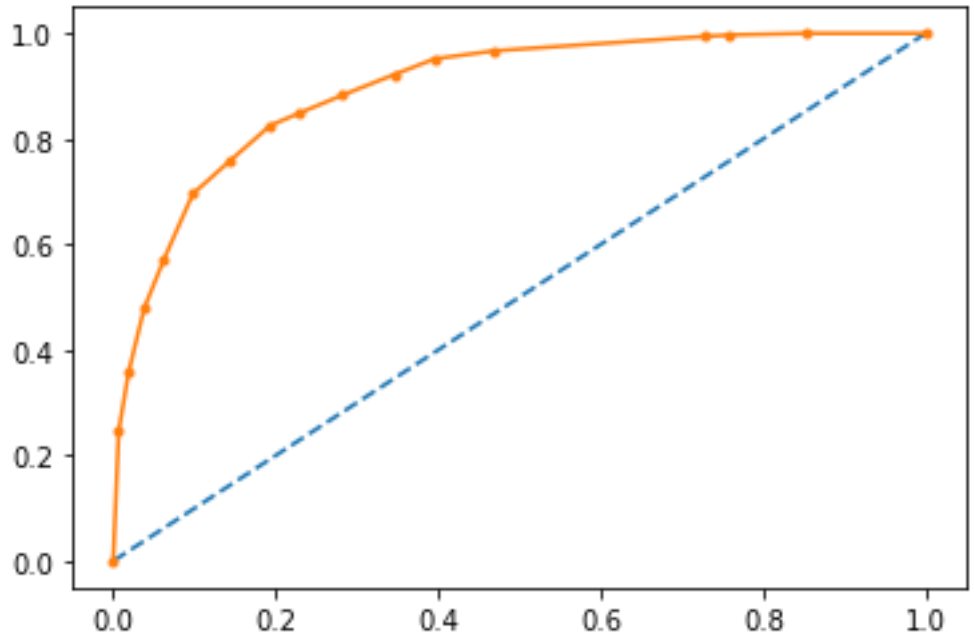
Confusion Matrix for Trained Dataset



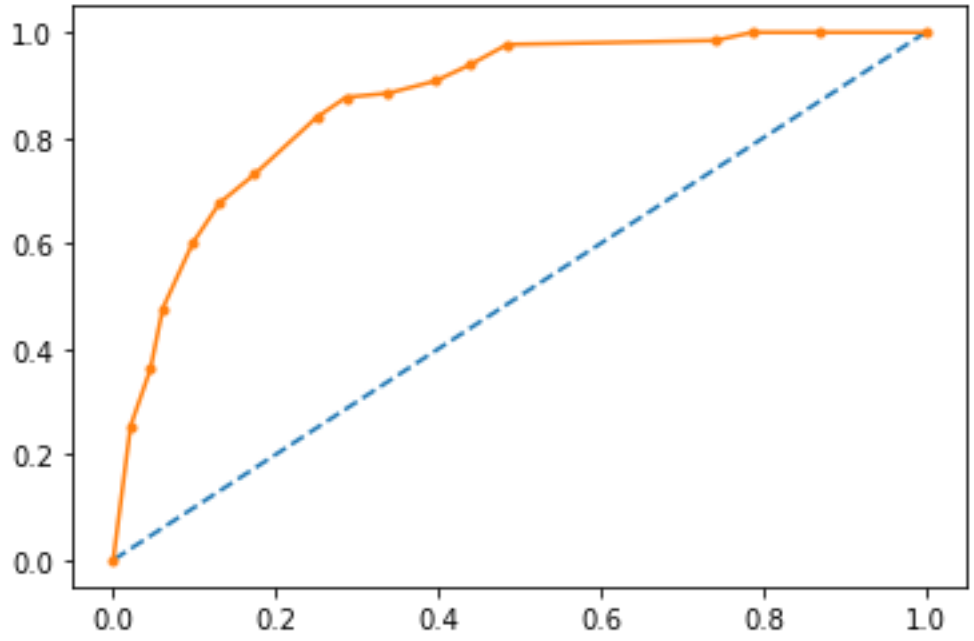
Confusion Matrix for Test Dataset



AUC-ROC Curve for Trained Dataset:



AUC-ROC Curve for Test Dataset:



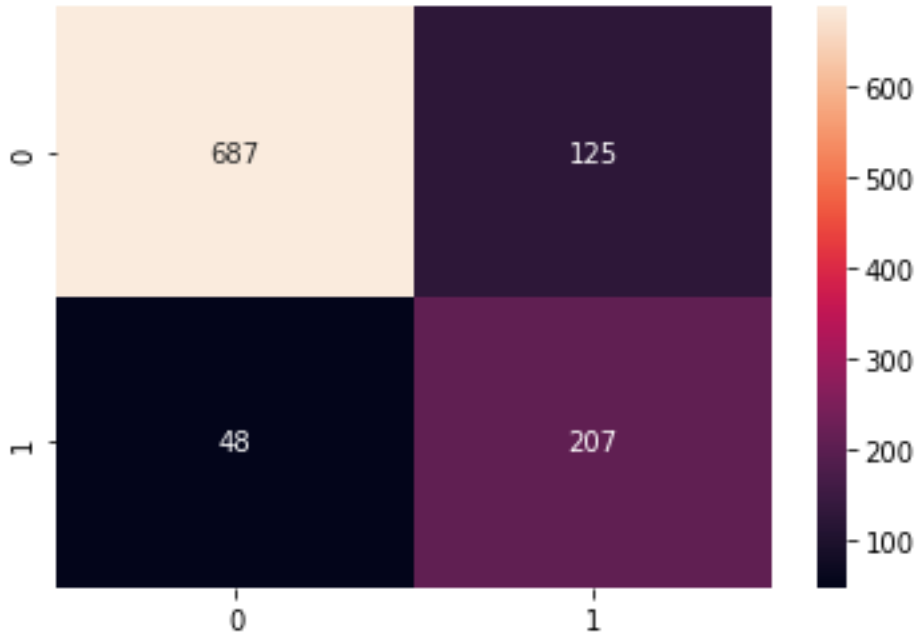
Accuracy for Trained Dataset 89%

Accuracy for Test Dataset 87%

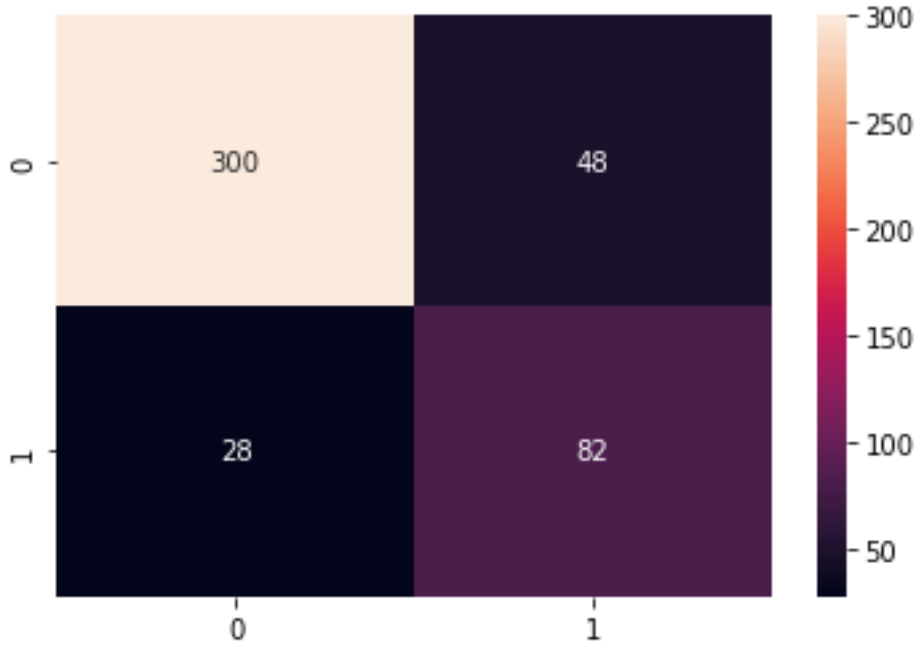
Validation: the model not Overfit nor Underfit.

Random Forest with Tuning(Hyper Parameters)

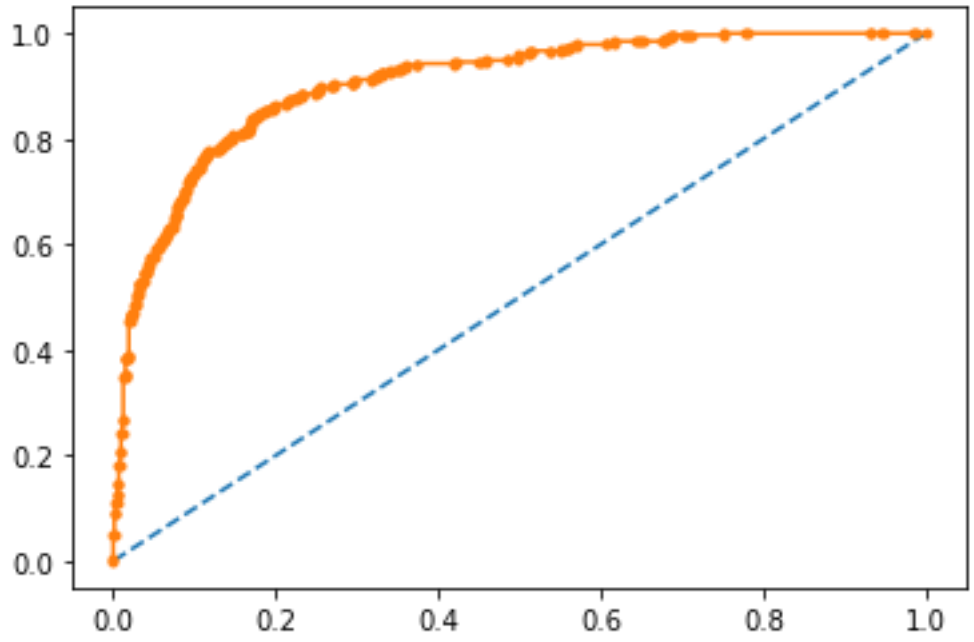
Confusion Matrix with TrainDataset:



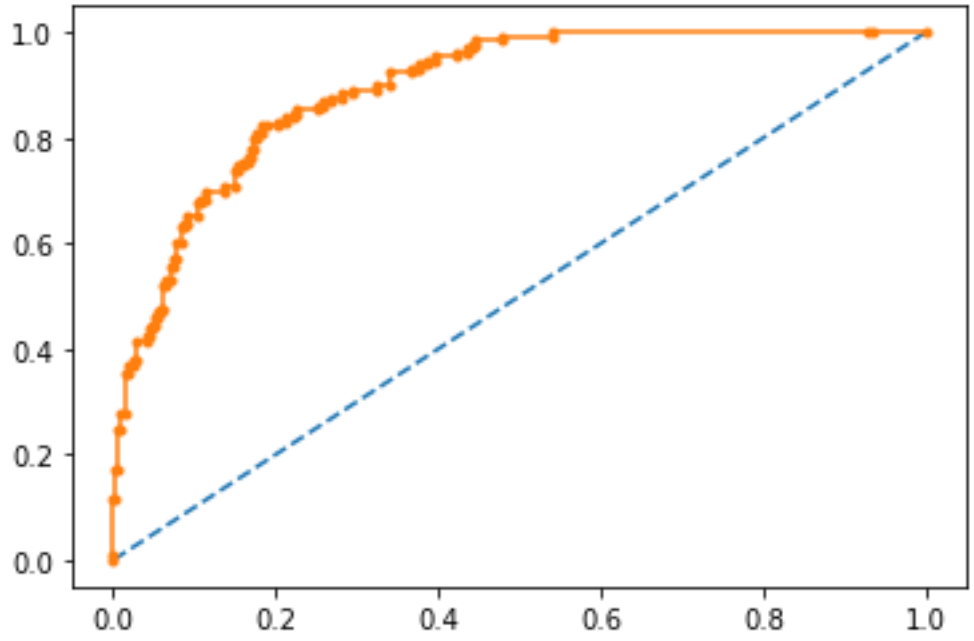
Confusion with Test Dataset:



AUC-ROC Curve for Trained Dataset



AUC-ROC Curve for Test Dataset:



Accuracy for Train Dataset: 90%

Accuracy for Test Dataset: 89%

Validation: Model is not Overfit nor Underfit.

Final Model: Compare the models and write inference which model is best/optimized.

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. There should be at least 3-4 Recommendations and insights in total. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks should only be allotted if the recommendations are correct and business specific.

In all the models, tuned ones are better than the regular models.

Conclusion:

• There is no under-fitting or over-fitting in any of the tuned models.

• All the tuned models have high values and every model is good. But as we can see, the most consistent tuned model in both train and test data is the Gradient Boost model.

• The tuned gradient boost model performs the best with 88.31% accuracy score in train and 87.28% accuracy score in test. Also it has the best AUC score of 94% inboth train and test data which is the highest of all the models.

• It also has a precision score of 88% and recall of 94% which is also the highest of all the models. So, we conclude that Gradient Boost Tuned model is the best/optimized model.

Problem -2

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

• President Franklin D. Roosevelt's speech have 7571 characters .

• President John F. Kennedy's speech have 7618 characters

• President Richard Nixon's speech have 9991 characters Number of words:

• There are 1360 words in President Franklin D. Roosevelt's speech.

• There are 1390 words in President John F. Kennedy's speech.

• There are 1819 words in President Richard Nixon's speech.

Number of sentences:

• There are 68 sentences in President Franklin D. Roosevelt's speech.

• There are 52 sentences in President John F. Kennedy's speech.

• There are 68 sentences in President Richard Nixon's speech

2.2 Remove all the stop words from all three speeches

Word count before the removal of stop-words,

• President Franklin D. Roosevelt's speech have 1360 words.

• President John F. Kennedy's speech have 1390 words.

• President Richard Nixon's speech have 1819 words. Word count after the removal of stop-words:

• President Franklin D. Roosevelt's speech have 871 words.

• President John F. Kennedy's speech have 904 words.

• President Richard Nixon's speech have 1094 words.

.

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 stop words)

Top 3 words in Roosevelt's speech:

- nation – 11
- know – 10
- spirit – 9

Top 3 words in Kennedy’s speech:

- let – 11
- us – 10
- sides – 9

Top 3 words in Nixon’s speech:

- us - 26
- let – 22
- peace – 19

2.4 Plot the word cloud of each of the speeches of the variable. (after removing the stop words) Word cloud of Roosevelt's speech:

Word Cloud for Roosevelt_word_cloud

Word cloud of Kennedy's speech:

Nixon_word_cloud