Experiment No. 9

Aim : Mini-Project : Data Mining for Breast Cancer Identification

Code :

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from yellowbrick.classifier import ClassificationReport
```

```
bcd =
pd.read_csv("C:\\Users\\Jayesh\\Desktop\\breast-cancer-wisconsin-data\\data.csv")
```

```
print(bcd.head())
print(bcd.tail())
print(bcd.dtypes)
```

```
def b():

    sns.set(style="whitegrid", color_codes=True)

    sns.set(rc={'figure.figsize':(8,8)})

    sns.countplot('diagnosis',data=bcd,hue = 'diagnosis')

    sns.despine(offset=10, trim=True)

    plt.show()


b()



def v():

    sns.set(rc={'figure.figsize':(8,8)})
    sns.violinplot(x="diagnosis",y="radius_mean", hue="diagnosis", data=bcd)
    plt.show()

v()



data = bcd[['radius_mean']]
target = bcd['diagnosis']
train, test, train_labels, test_labels = train_test_split(data,
                                                          target,
                                                          test_size=0.15,
                                                          random_state=42)



print("\n")
print("1. By KNN algorithm")

print("\n")

knn = KNeighborsClassifier()
model2=knn.fit(train,train_labels)
p= knn.predict(test)

print("Accuracy of KNN classifier:"+str(accuracy_score(test_labels,p)))

y_actu = bcd['diagnosis'].head(n=86)
y_pred = p

print("\n")
df_confusion = pd.crosstab(y_actu, y_pred, rownames=['Actual'],
colnames=['Predicted'], margins=True)
print(df_confusion)
print("\n")

visualizer = ClassificationReport(knn, classes=['M','B'],size=(600,600))
visualizer.fit(train, train_labels)
visualizer.score(test,test_labels)
b = visualizer.poof()




print("\n")
print("2. By Naive Bayes algorithm")
print("\n")
gnb = GaussianNB()
model = gnb.fit(train, train_labels)
preds = gnb.predict(test)
print("Accuracy of Naive Bayes classifier:"+str(accuracy_score(test_labels, preds)))

y_actu = bcd['diagnosis'].head(n=86)
y_pred = preds
print("\n")
```

```
df_confusion = pd.crosstab(y_actu, y_pred, rownames=['Actual'],
colnames=['Predicted'], margins=True)
print(df_confusion)
print("\n")

visualizer = ClassificationReport(gnb, classes=['M','B'],size=(600,600))
visualizer.fit(train, train_labels)
visualizer.score(test,test_labels)
a = visualizer.poof()








print("\n")


print("3. By Random Forest algorithm")
print("\n")
rfor = RandomForestClassifier()
model3=rfor.fit(train,train_labels)
ro=rfor.predict(test)
print("Accuracy of Random Forest classifier:"+str(accuracy_score(test_labels,ro)))


y_actu = bcd['diagnosis'].head(n=86)
y_pred = ro
print("\n")
df_confusion = pd.crosstab(y_actu, y_pred, rownames=['Actual'],
colnames=['Predicted'], margins=True)
print(df_confusion)
print("\n")

visualizer = ClassificationReport(rfor, classes=['M','B'],size=(600,600))
visualizer.fit(train, train_labels)
visualizer.score(test,test_labels)
c = visualizer.poof()
```

```
print("\n")

print("4. By Decision Tree algorithm")
print("\n")
clf = DecisionTreeClassifier()
model1=clf.fit(train,train_labels)
pred=clf.predict(test)

print("Accuracy of Decision Tree classifier:"+str(accuracy_score(test_labels,pred)))


y_actu = bcd['diagnosis'].head(n=86)
y_pred = pred
print("\n")
df_confusion = pd.crosstab(y_actu, y_pred, rownames=['Actual'],
colnames=['Predicted'], margins=True)
print(df_confusion)
print("\n")

visualizer = ClassificationReport(clf, classes=['M','B'],size=(600,600))
visualizer.fit(train, train_labels)
visualizer.score(test,test_labels)
c = visualizer.poof()
```

Output :

Dataset : breast-cancer-wisconsin-data

1. First Five Entries of Dataset :

```
Sr.no      id          diagnosis     radius_mean   ….   fractal_dimension_worst

 1       842302           M              17.99      ….         0.11890
 2       842517           M              20.57      ….         0.08902
 3       84300903         M              19.69      ….         0.08758
 4       84348301         M              11.42      ….         0.17300
 5       84358402         M              20.29      ….         0.07678
```

2.Last Five Entries of Dataset :

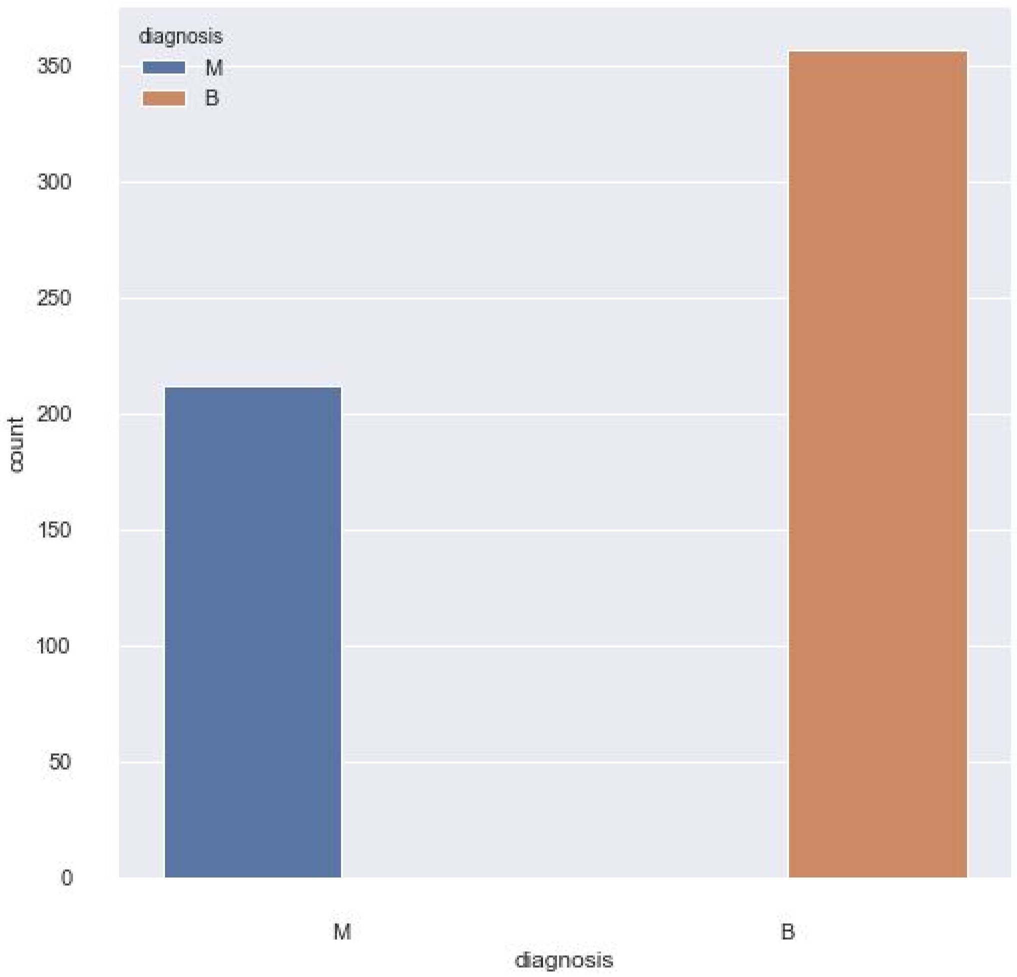
```
Sr.no      id          diagnosis     radius_mean   ….   fractal_dimension_worst

565      926424           M             21.56       ….           0.07115
566      926682           M             20.13       ….           0.06637
567      926954           M             16.6        ….           0.07820
568      927241           M             20.6        ….           0.12400
569      92751            B              7.76        ….           0.07039
```

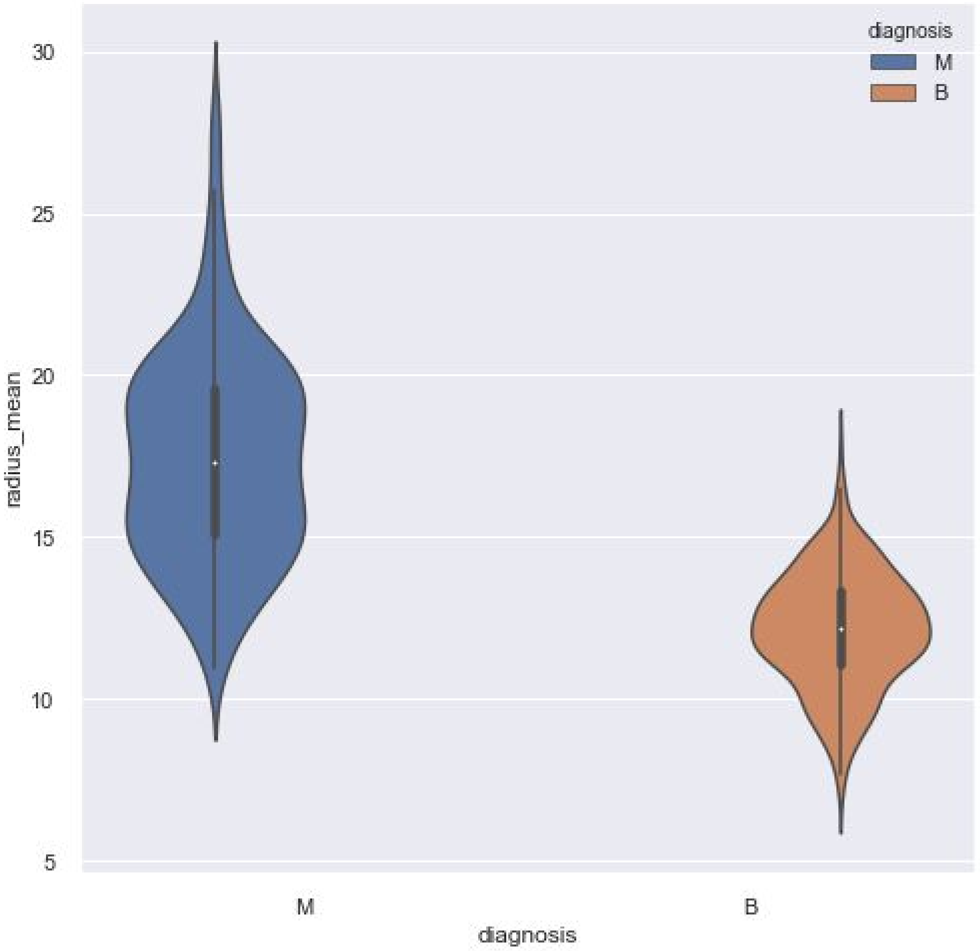
3.To display the different datatypes of each column in the dataset :

Columns	Datatype
id	int64
diagnosis	object
radius_mean	float64
texture_mean	float64
perimeter_mean	float64
area_mean	float64
smoothness_mean	float64
compactness_mean	float64
concavity_mean	float64
concave points_mean	float64
symmetry_mean	float64
fractal_dimension_mean	float64
radius_se	float64
texture_se	float64
perimeter_se	float64
area_se	float64
smoothness_se	float64
compactness_se	float64
concavity_se	float64
concave points_se	float64
symmetry_se	float64
fractal_dimension_se	float64
radius_worst	float64
texture_worst	float64
perimeter_worst	float64
area_worst	float64
smoothness_worst	float64
compactness_worst	float64
concavity_worst	float64
concave points_worst	float64
symmetry_worst	float64
fractal_dimension_worst	float64

4.To Create a Countplot for diagnosis data :



5. To create a Violinplot for radius_mean vs diagnosis data :



Comparison of Algorithms :

1.K-Nearest Neighbors (KNN) :

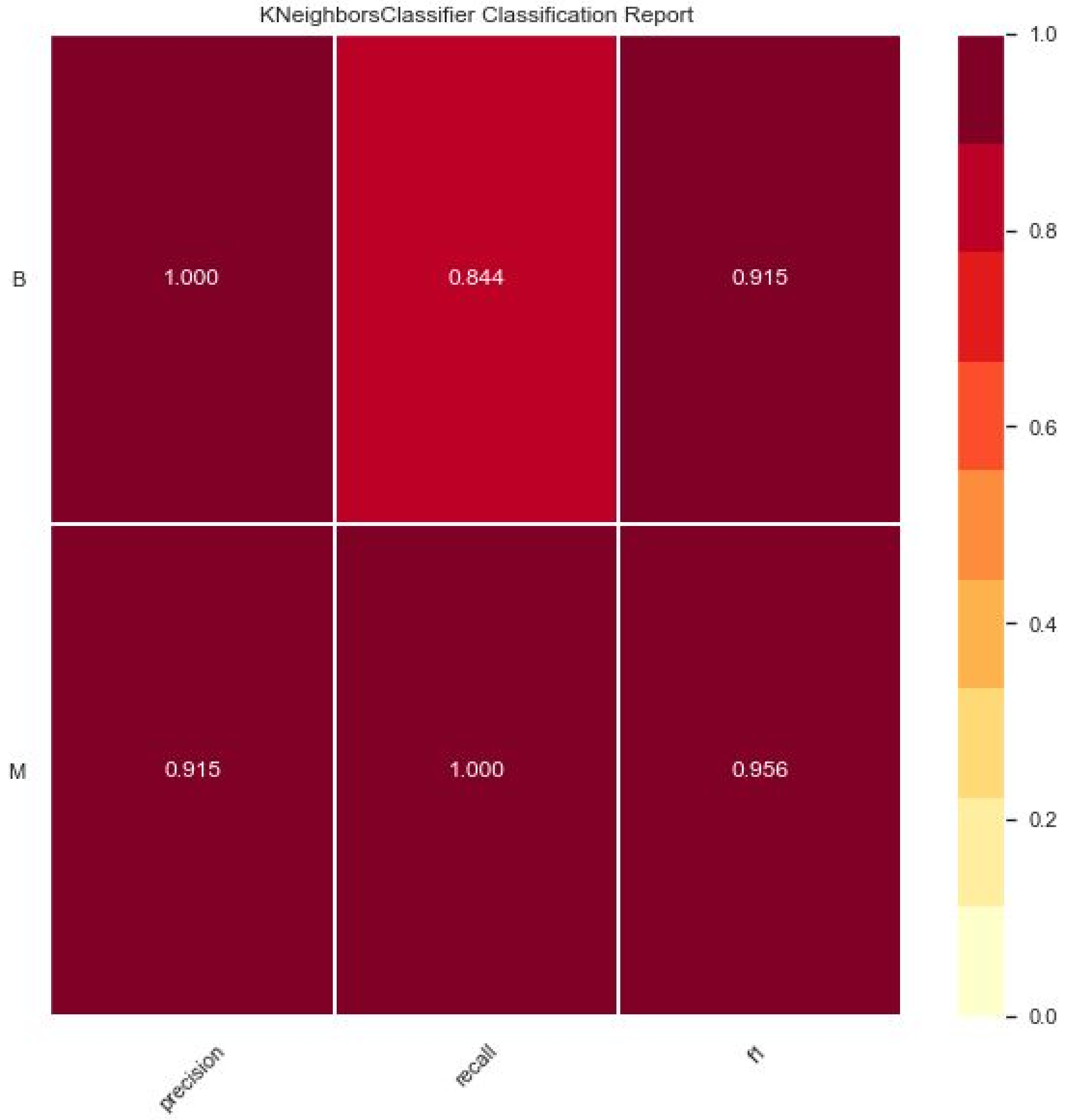
Definition : KNN algorithms use data and classify new data points based on similarity measures (e.g. distance function). Classification is done by a majority vote to its neighbors. The data is assigned to the class which has the nearest neighbors. As you increase the number of nearest neighbors, the value of k, accuracy might increase.

Accuracy of KNN classifier : 0.9418604651162791

Confusion matrix of KNN classifier :

```
             Predicted
              B    M    All
Actual
B            18    9    27
M            41   18    59
All          59   27    86
```

Presion , Recall , F1-Score of KNN classifier :



2. Naive Bayes :

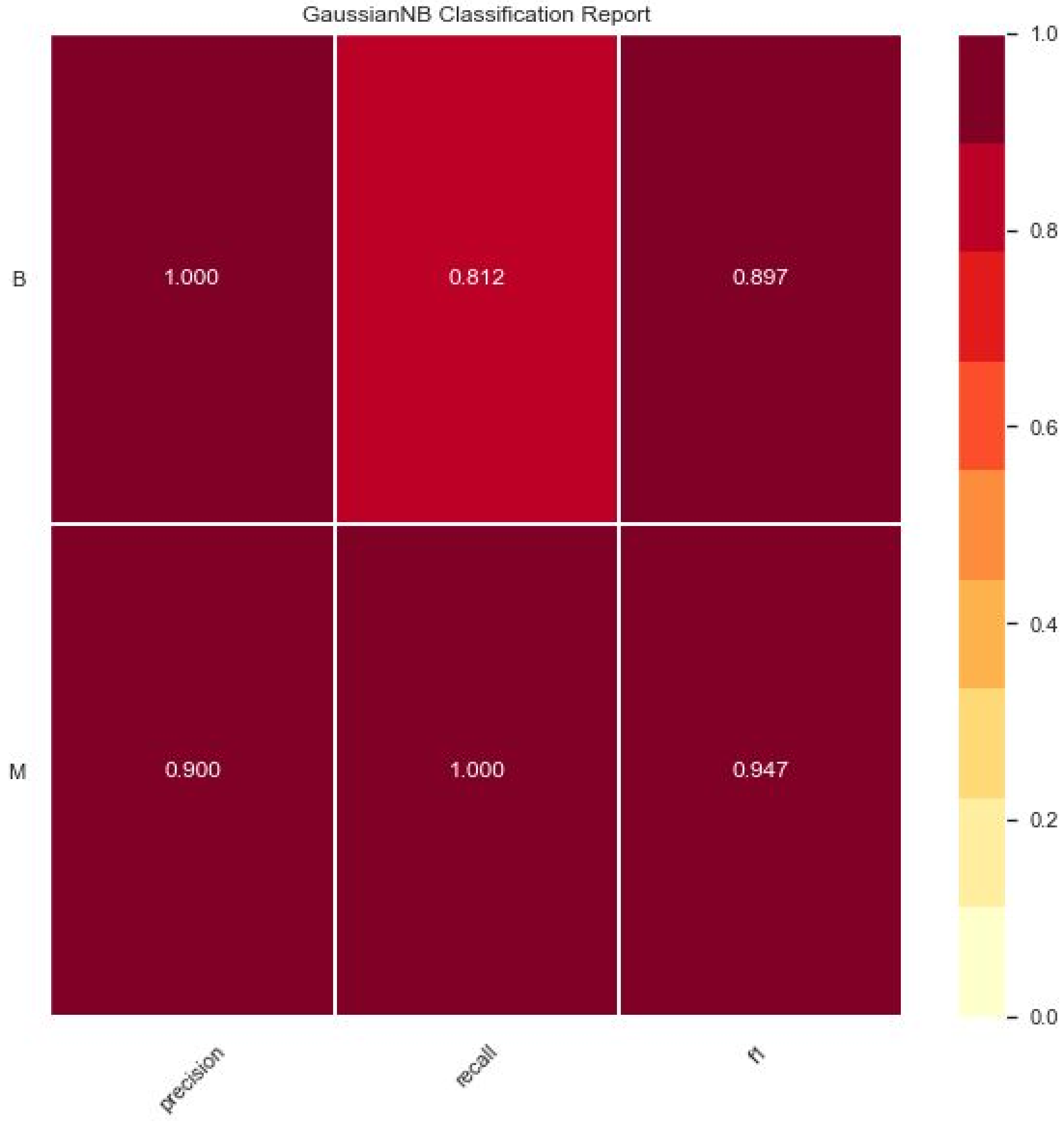
Definition : Naive Bayes classifiers are a collection of classification algorithms based on Bayes’ Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Accuracy of Naive Bayes classifier : 0.9302325581395349

Confusion matrix of Naive Bayes classifier :

```
           Predicted
           B    M    All
Actual
B          18   9    27
M          42   17   59
All        60   26   86
```

Precision , Recall , F1-Score of Naive Bayes classifier :



3.Random Forest :

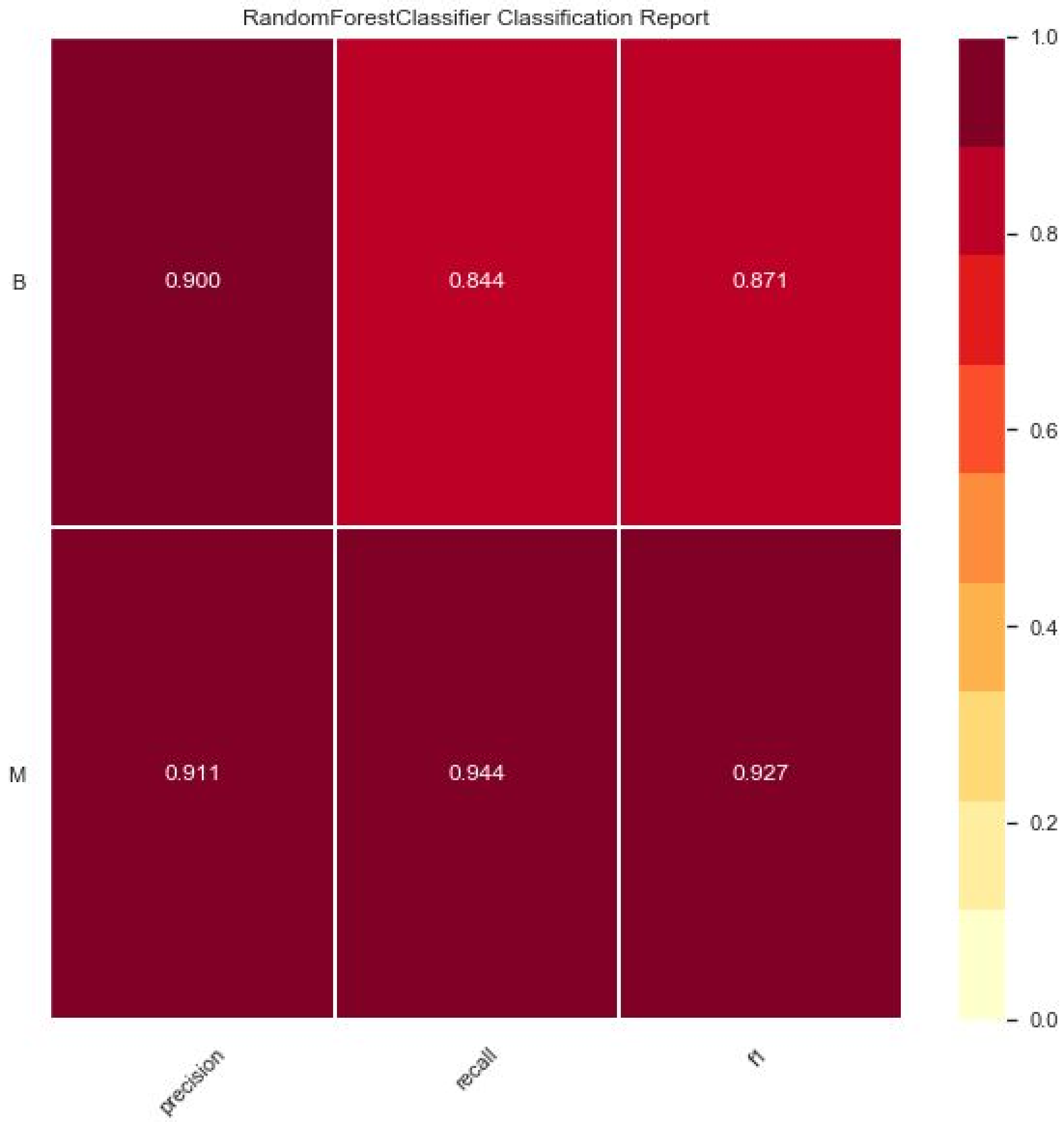
Definition : The random forest is a supervised learning algorithm that randomly creates and merges multiple decision trees into one “forest.” The goal is not to rely on a single learning model, but rather a collection of decision models to improve accuracy. The primary difference between this approach and the standard decision tree algorithms is that the root nodes feature splitting nodes are generated randomly.

Accuracy of Random Forest classifier : 0.9186046511627907

Confusion matrix of Random Forest classifier :

```
          Predicted
           B   M   All
Actual
B         17  10   27
M         40  19   59
All       57  29   86
```

Precision , Recall , F1-Score of Random Forest Classifier :



4.Decision Tree :

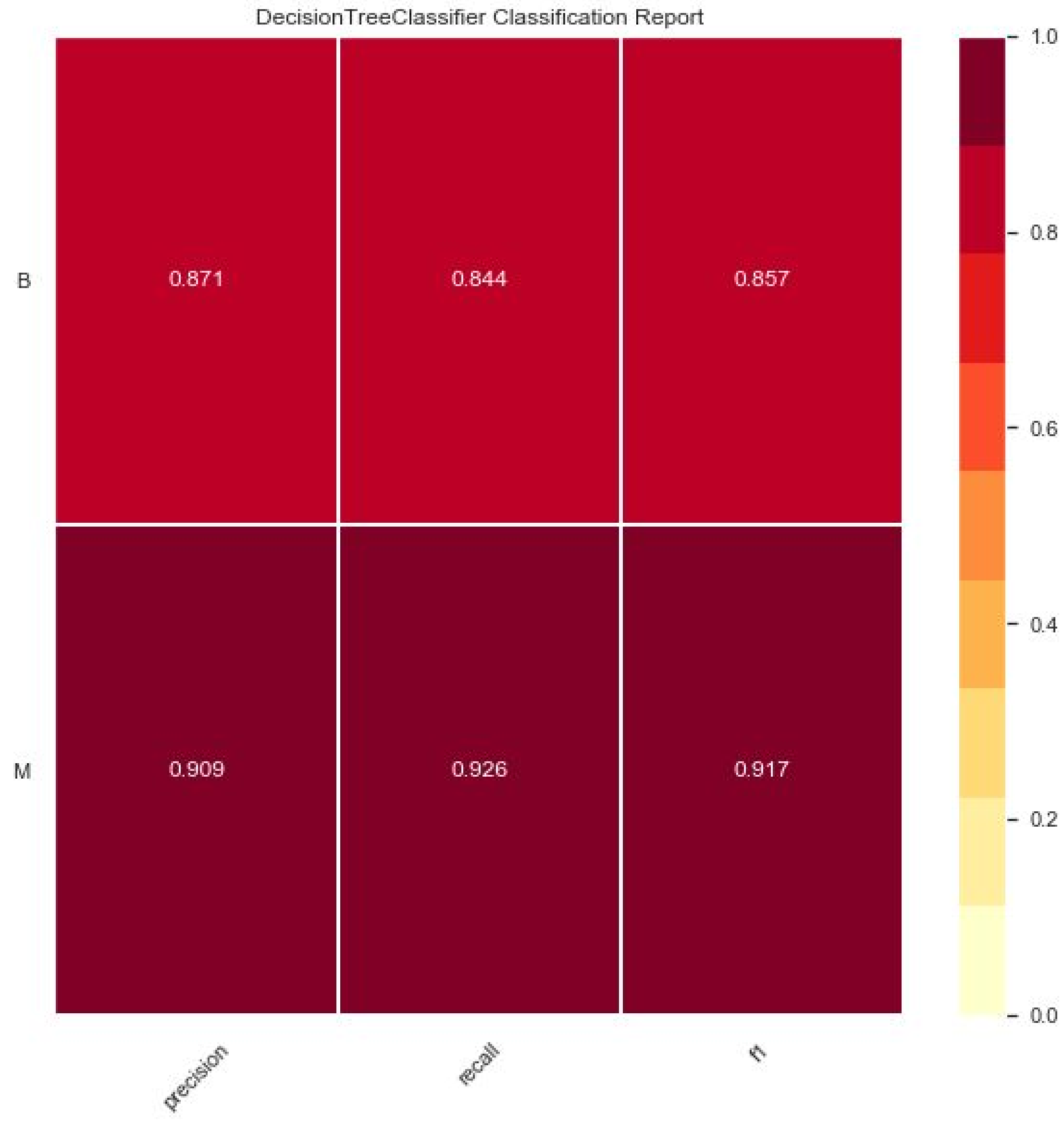
Definition : Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.Decision Tree is used to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

Accuracy of Decision Tree classifier : 0.8953488372093024

Confusion matrix of Decision Tree classifier :

```
           Predicted
           B   M  All
Actual
B          16  11  27
M          39  20  59
All        55  31  86
```

Precision , Recall , F1-Score of Decision Tree Classifier :


DecisionTreeClassifier Classification Report
B
M
precision
recall
f1