# Experiment 03

# Code:

# 

# Naïve bayes

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset=pd.read\_csv('diabetes.csv')

X= dataset.iloc[:,:-1].values

y= dataset.iloc[:,-1].values

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,random\_state=0)

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score,confusion\_matrix

clf1 = GaussianNB()

clf1.fit(X\_train, y\_train)

predicted1 = clf1.predict(X\_test)

accuracy1 = accuracy\_score(predicted1,y\_test)

print('Accuracy\_score=',accuracy1

from sklearn.metrics.\_plot.confusion\_matrix import ConfusionMatrixDisplay

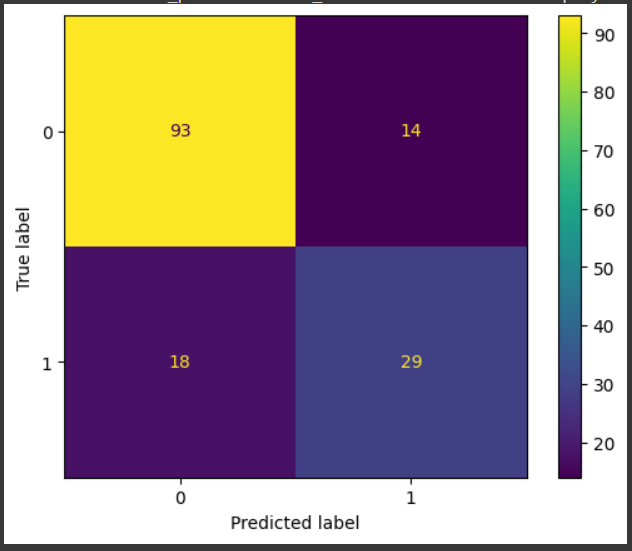
lables=[0,1]

cm1=confusion\_matrix(y\_test,predicted1,labels=lables)

disp1=ConfusionMatrixDisplay(confusion\_matrix=cm1,display\_labels=lables)

disp1.plot()

**Output:**



import numpy as np

import matplotlib.pyplot as plt

from sklearn.metrics import roc\_curve, roc\_auc\_score, auc

fpr, tpr, thresholds = roc\_curve(y\_test, predicted1)

auroc = roc\_auc\_score(y\_test, predicted1)

print(f"AUROC: {auroc}")



plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'AUROC = {auroc:.2f}')

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

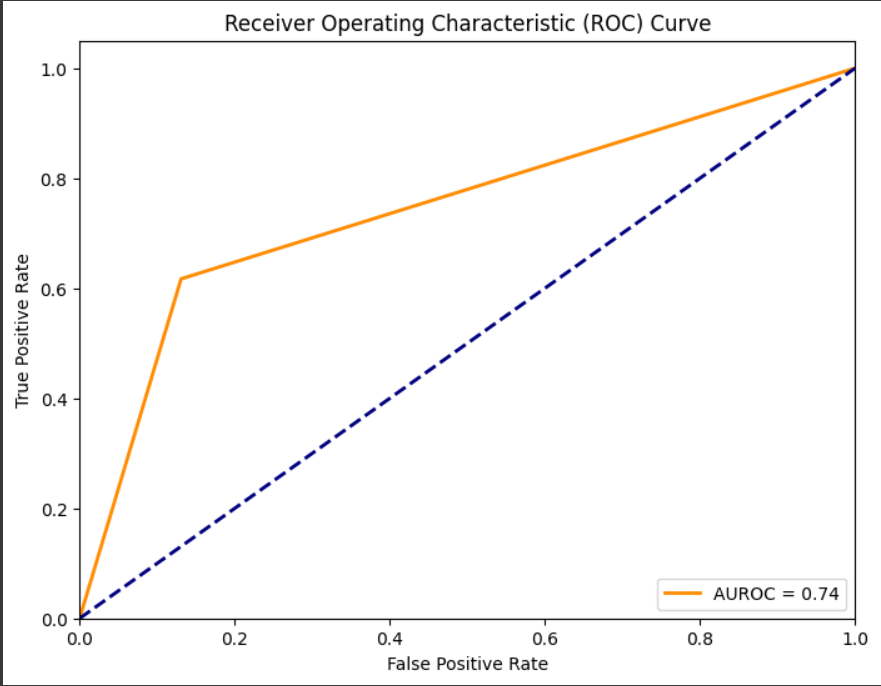
plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic (ROC) Curve')

plt.legend(loc='lower right')

plt.show()



**Decision Tree**

from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier()

clf.fit(X\_train,y\_train)

DT\_predicted=clf.predict(X\_test)

DT1 = accuracy\_score(DT\_predicted, y\_test)

print("Accuracy:", DT1)

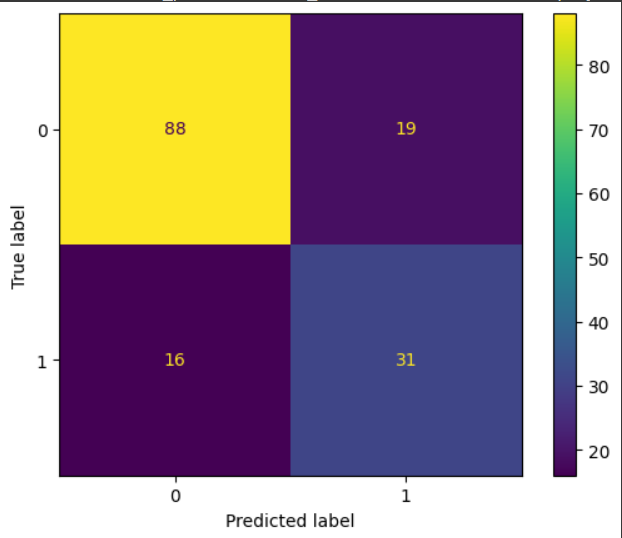


labels = [0,1]

cm = confusion\_matrix(y\_test, DT\_predicted, labels=labels)

disp = ConfusionMatrixDisplay(confusion\_matrix=cm, display\_labels=labels)

disp.plot()



fpr, tpr, thresholds = roc\_curve(y\_test, predicted1)

auroc = roc\_auc\_score(y\_test, DT\_predicted)

print(f"AUROC: {auroc}")



**K Fold**

from sklearn.model\_selection import KFold

from sklearn.model\_selection import cross\_val\_score

kfold = KFold(n\_splits = 10, shuffle = True , random\_state=42)

scores = cross\_val\_score(clf1,X,y,cv=kfold,scoring='accuracy')

for i in scores:

  print(i)

