**Ex-7 Naive Bayes Classification**

**Python program:**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.naive\_bayes import GaussianNB

from sklearn.preprocessing import LabelEncoder

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

from sklearn.metrics import accuracy\_score

from sklearn import metrics

data=pd.read\_csv('naive.csv')

print("Before Transformation")

print(data)

**#data preprocessing convert the string into integer**

number = LabelEncoder()

data['age']=number.fit\_transform(data['age'])

data['income']=number.fit\_transform(data['income'])

data['student']=number.fit\_transform(data['student'])

data['credit']=number.fit\_transform(data['credit'])

data['buy']=number.fit\_transform(data['buy'])

print("After Transformation")

print(data)

**#selected the important features**

feature=['age','income','student','credit']

x=data[feature]

y=data.buy

xtr,xte,ytr,yte=train\_test\_split(x,y,test\_size=0.3,random\_state=0)

model=GaussianNB()

model.fit(xtr,ytr)

ypre=model.predict(xte)

acc=accuracy\_score(yte,ypre)

**#confusion matrix**

from sklearn.metrics import classification\_report,confusion\_matrix

cm=confusion\_matrix(y,model.predict(x))

**#classification report**

print(classification\_report(y,model.predict(x)))

**#visulaization of heatmap**

fig,ax=plt.subplots(figsize=(8,8))

ax.imshow(cm)

ax.xaxis.set(ticks=(0,1),ticklabels=('Predicted 0s','Predicted 1s'))

ax.yaxis.set(ticks=(0,1),ticklabels=('Actual 0s','Actual 1s'))

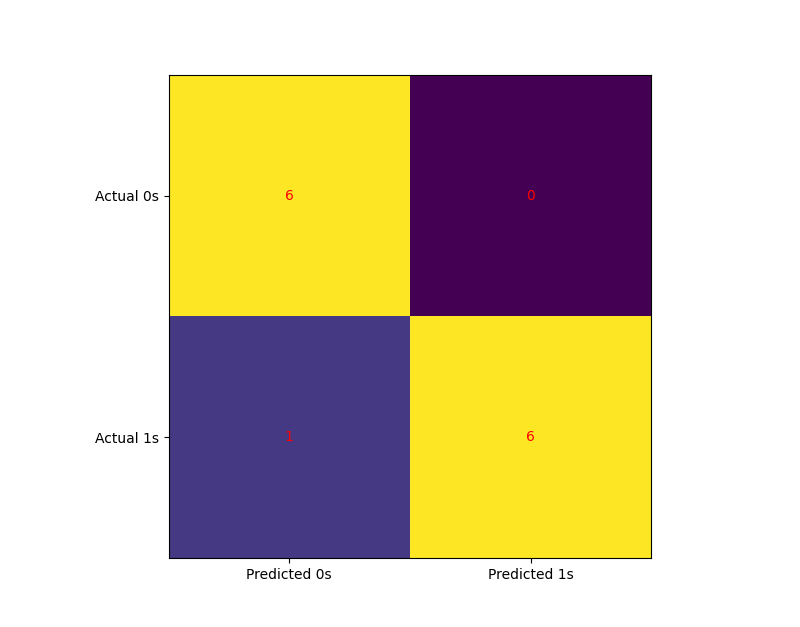
for i in range(2):

for j in range(2):

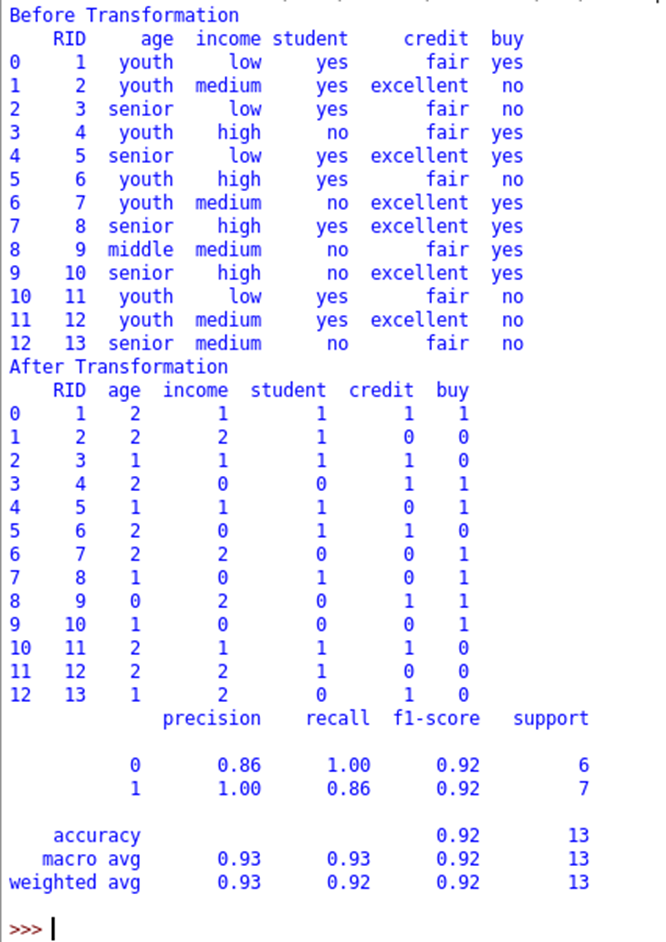
ax.text(j,i,cm[i,j],ha='center',va='center',color='red')

plt.show()

**Output figure:**



**Output:**

****

**Ex: 5**

