import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd  
import seaborn as sns  
# importing the dataset  
dataset = pd.read\_csv ('/Users/mac/Desktop/NaiveBayes.csv')  
# split the data into inputs and outputs  
X = dataset.iloc[:, [0,1]].values  
y = dataset.iloc[:, 2].values  
# training and testing data  
from sklearn.model\_selection import train\_test\_split  
  
# assign test data size 25%  
X\_train, X\_test, y\_train, y\_test =train\_test\_split(X,y,test\_size= 0.25, random\_state=0)  
# importing standard scaler  
from sklearn.preprocessing import StandardScaler  
  
# scalling the input data  
sc\_X = StandardScaler()  
X\_train = sc\_X.fit\_transform(X\_train)  
X\_test = sc\_X.fit\_transform(X\_test)  
# importing classifier  
from sklearn.naive\_bayes import BernoulliNB  
  
# initializaing the NB  
classifer = BernoulliNB()  
  
# training the model  
classifer.fit(X\_train, y\_train)  
  
# testing the model  
y\_pred = classifer.predict(X\_test)  
# importing accuracy score  
from sklearn.metrics import accuracy\_score  
  
# printing the accuracy of the model  
print(accuracy\_score(y\_pred, y\_test))  
# import Gaussian Naive Bayes classifier  
from sklearn.naive\_bayes import GaussianNB  
  
# create a Gaussian Classifier  
classifer1 = GaussianNB()  
  
# training the model  
classifer1.fit(X\_train, y\_train)  
  
# testing the model  
y\_pred1 = classifer1.predict(X\_test)  
# importing accuracy score  
from sklearn.metrics import accuracy\_score  
  
# printing the accuracy of the model  
print(accuracy\_score(y\_test,y\_pred1))  
  
# importing the required modules  
import seaborn as sns  
from sklearn.metrics import confusion\_matrix  
  
# passing actual and predicted values  
cm = confusion\_matrix(y\_test, y\_pred)  
  
# true write data values in each cell of the matrix  
sns.heatmap(cm, annot=True)  
plt.savefig('confusion.png')  
# importing classification report  
from sklearn.metrics import classification\_report  
  
# printing the report  
print(classification\_report(y\_test, y\_pred))  
# importing the required modules  
import seaborn as sns  
from sklearn.metrics import confusion\_matrix  
  
# passing actual and predicted values  
cm = confusion\_matrix(y\_test, y\_pred1)  
  
# true write data values in each cell of the matrix  
sns.heatmap(cm,annot=True)  
plt.savefig('confusion.png')  
# importing classification report  
from sklearn.metrics import classification\_report  
  
# printing the report  
print(classification\_report(y\_test, y\_pred1))