**Worksheet-2.2**

**Student Name:-** Pushpraj Roy **UID:-** 20BCS9866

**Branch:-** BE- CSE **Section/Group:-** WM\_617 “A”

**Subjetct Code:-** 20CSP-317 **Semester:-** 5th

**Subject Name:-** Machine Learning Lab

1. **Aim/Overview of the practical: -**

To implement Naive Bayes algorithm.

1. **Task to be done/ Which logistics used:-**

To prepare a model with Naive Bayes algorithm.

1. **Source Code:-**

# importing the libraries import numpy as np import matplotlib.pyplot as plt import pandas as pd

import seaborn as sns

# importing the dataset dataset = pd.read\_csv('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))

# assigning features and label variables weather = ['Sunny','Sunny','Overcast','Rainy','Rainy','Rainy','Overcast','Sunny','Sun ny', 'Rainy','Sunny','Overcast','Overcast','Rainy']

# output class

play = ['No','No','Yes','Yes','Yes','No','Yes','No','Yes','Yes','Yes','Yes','Yes','No ']

|  |
| --- |
| # Import LabelEncoder from sklearn import preprocessing    # creating LabelEncoder labelCode = preprocessing.LabelEncoder()    # Converting string labels into numbers.  wheather\_encoded=labelCode.fit\_transform(weather) |

#Printing the encoded values

print(wheather\_encoded)

|  |
| --- |
| # import LabelEncoder from sklearn import preprocessing    # creating LabelEncoder labelCode = preprocessing.LabelEncoder()    # converting string labels into numbers.  label=labelCode.fit\_transform(play) |

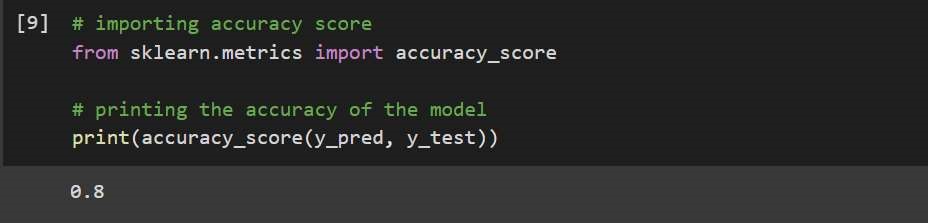
|  |
| --- |
| # importing numpy module import numpy as np    # converting 1D array to 2D weather\_2d = np.reshape(wheather\_encoded, (-1, 1)) |

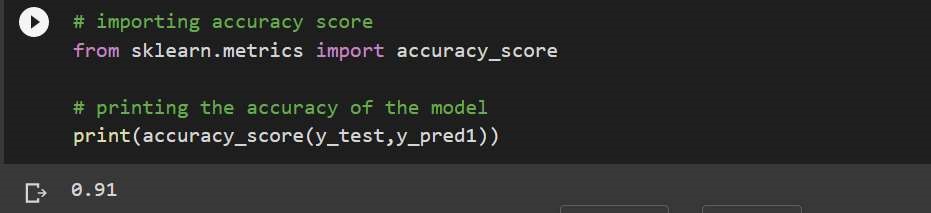
|  |
| --- |
| # import Gaussian Naive Bayes model from sklearn.naive\_bayes import GaussianNB    # create a Gaussian Classifier model = GaussianNB()    # train the model using the training sets model.fit(weather\_2d, label) |

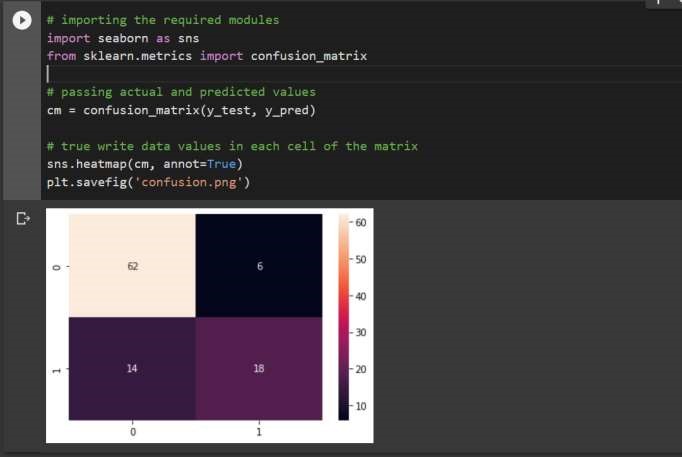
# predicting the odel predicted= model.predict([[0]]) # 0:Overcast

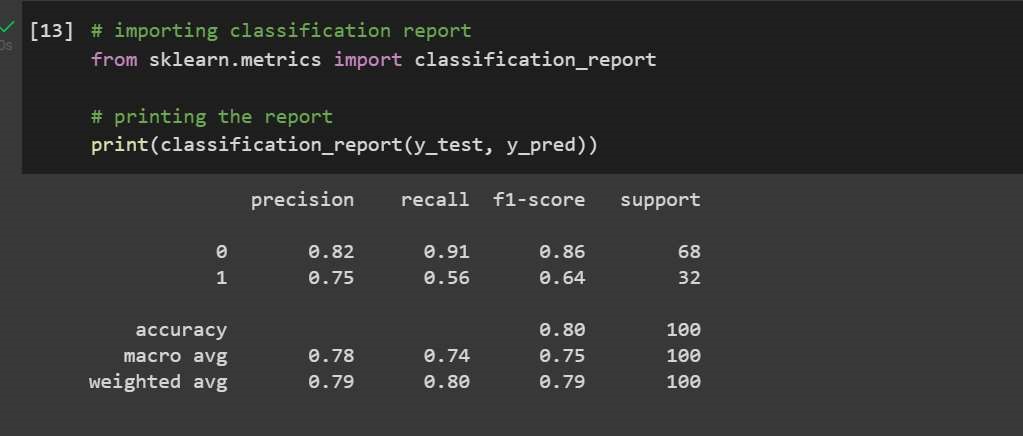
# printing predicted value print(predicted)

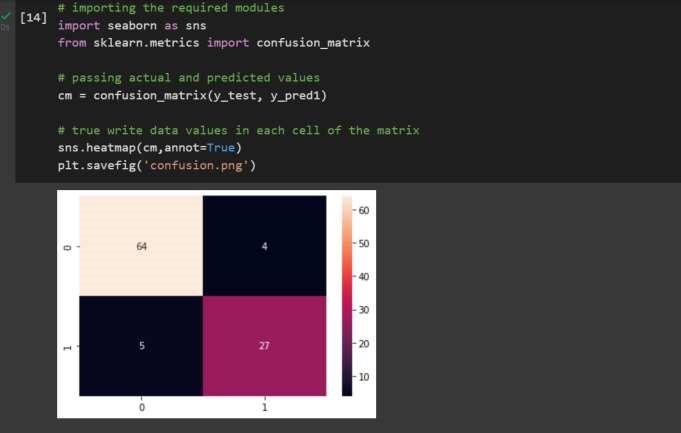
**3. Result/Output**

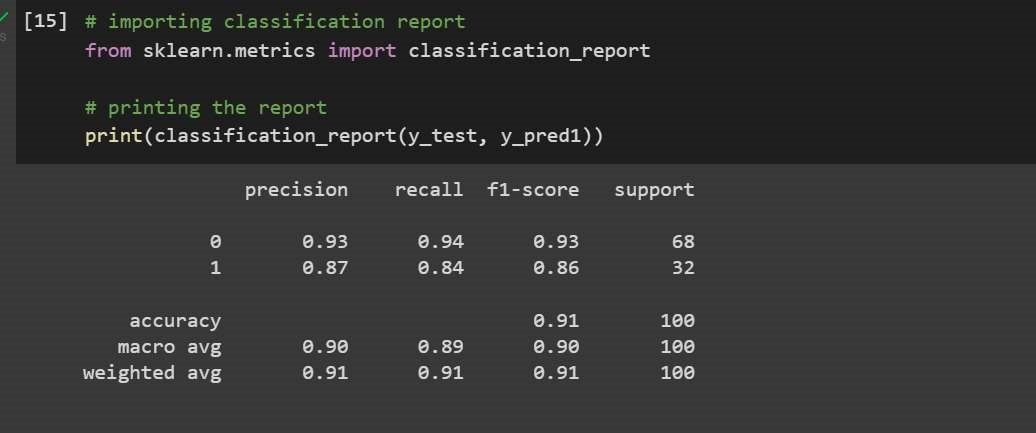


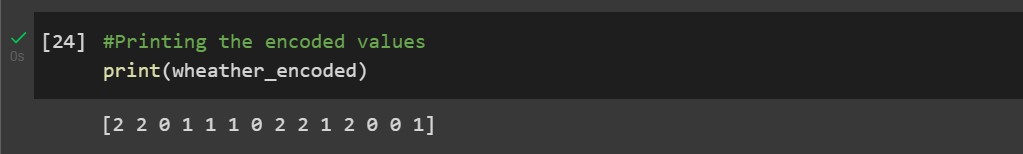


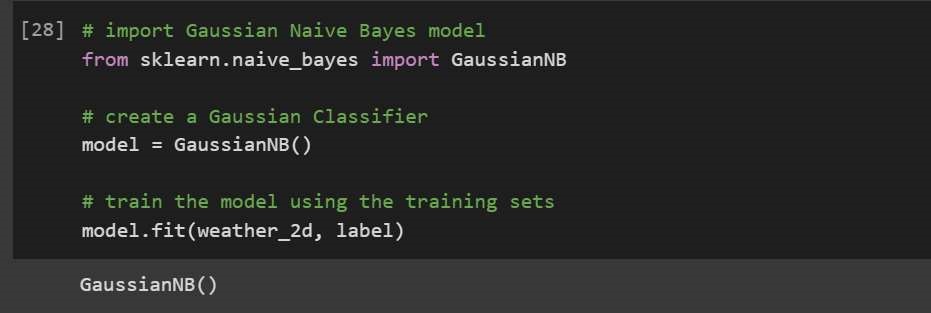


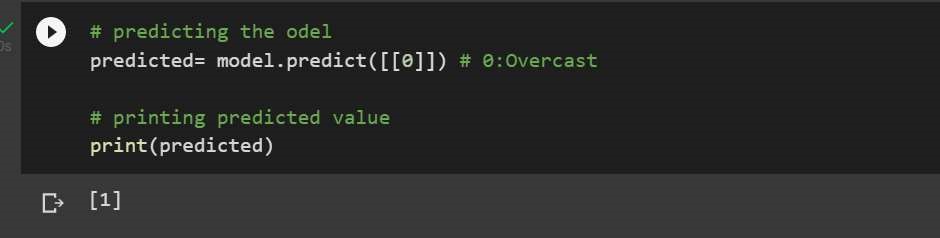












**Learning outcomes (What I have learnt):**

1. Understanding of Exploratory Navies Byes.
2. Able to analyze different Navies Byes with the help of python and pandas library.
3. Learning about different library/packages of python.
4. Learning about the different methods, that are needed to analyze the given dataset.
5. Learning of different Machine Learning Functions