**Assignment No 6**

**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.datasets import load\_iris

from sklearn.preprocessing import StandardScaler

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

from sklearn.naive\_bayes import GaussianNB

from mlxtend.plotting import plot\_confusion\_matrix

from sklearn.metrics import confusion\_matrix, accuracy\_score, classification\_report, precision\_score, recall\_score, f1\_score

import warnings

warnings.filterwarnings("ignore")

#iris = pd.read\_csv('iris.csv')

iris = load\_iris()

iris.keys()

x = pd.DataFrame(iris['data'], columns=iris['feature\_names'])

y = pd.DataFrame(iris['target'], columns=['target'])

print("------------------------Dataset Info---------------------")

print(x.head())

print(x.shape, y.shape)

print(x.info())

print(y.info())

print(x.describe())

scaler = StandardScaler()

x = scaler.fit\_transform(x.values)

print("-------------------------Train and Test Split--------------------")

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y.values, test\_size=0.2, random\_state=42)

print(x\_train.shape, x\_test.shape, y\_train.shape, y\_test.shape)

print("------------------Gaussian Naive Bayes----------------------------")

model = GaussianNB()

model.fit(x\_train, y\_train)

y\_pred = model.predict(x\_test)

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

plot\_confusion\_matrix(conf\_mat=cm, figsize=(5,5), show\_normed=True)

plt.show()

print(f"TP value is {cm[0,0]}")

print(f"TN value is {cm[1,1] + cm[2,2]}")

print(f"FP value is {cm[0,1] + cm[0,2]}")

print(f"FN value is {cm[1,0] + cm[2,0]}")

print(f"Accuracy score is {accuracy\_score(y\_test, y\_pred)}")

print(f"Error rate is {1 - accuracy\_score(y\_test, y\_pred)}")

print(f"Precision score is {precision\_score(y\_test, y\_pred, average='macro')}")

print(f"Recall score is {recall\_score(y\_test, y\_pred, average='macro')}")

print("F1 Score:",(2\*P\*R)/(P+R))

**Output:**

------------------------Dataset Info---------------------

   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)

0                5.1               3.5                1.4               0.2

1                4.9               3.0                1.4               0.2

2                4.7               3.2                1.3               0.2

3                4.6               3.1                1.5               0.2

4                5.0               3.6                1.4               0.2

(150, 4) (150, 1)

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 150 entries, 0 to 149

Data columns (total 4 columns):

 #   Column             Non-Null Count  Dtype

---  ------             --------------  -----

 0   sepal length (cm)  150 non-null    float64

 1   sepal width (cm)   150 non-null    float64

 2   petal length (cm)  150 non-null    float64

 3   petal width (cm)   150 non-null    float64

dtypes: float64(4)

memory usage: 4.8 KB

None

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 150 entries, 0 to 149

Data columns (total 1 columns):

 #   Column  Non-Null Count  Dtype

---  ------  --------------  -----

 0   target  150 non-null    int32

dtypes: int32(1)

memory usage: 732.0 bytes

None

       sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)

count         150.000000        150.000000         150.000000        150.000000

mean            5.843333          3.057333           3.758000          1.199333

std             0.828066          0.435866           1.765298          0.762238

min             4.300000          2.000000           1.000000          0.100000

25%             5.100000          2.800000           1.600000          0.300000

50%             5.800000          3.000000           4.350000          1.300000

75%             6.400000          3.300000           5.100000          1.800000

max             7.900000          4.400000           6.900000          2.500000

-------------------------Train and Test Split--------------------

(120, 4) (30, 4) (120, 1) (30, 1)

------------------Gaussian Naive Bayes----------------------------

[[10  0  0]

 [ 0  9  0]

 [ 0  0 11]]

TP value is 10

TN value is 20

FP value is 0

FN value is 0

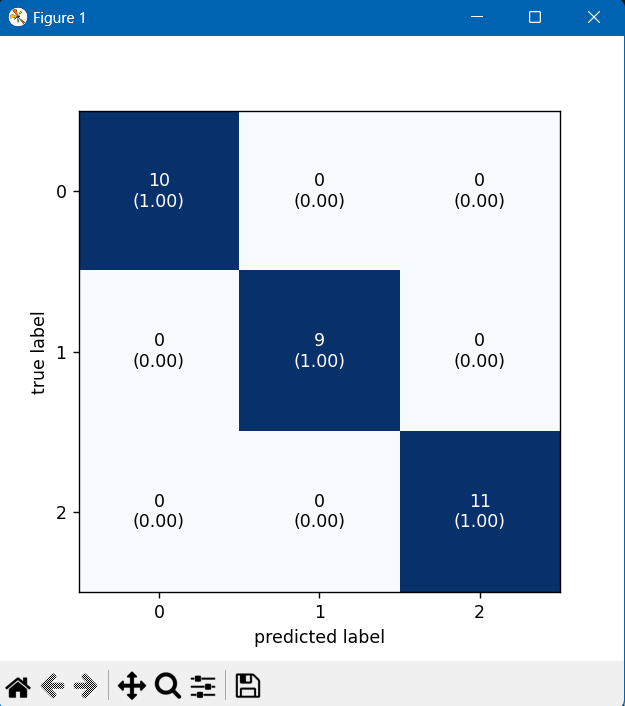
Accuracy score is 1.0

Error rate is 0.0

Precision score is 1.0

Recall score is 1.0

F1 Score: 1.0

****