Title : Data Analytics III

1. Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset.

2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision,Recall on

the given dataset.

import pandas as pd

import numpy as np

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

from sklearn.naive\_bayes import GaussianNB

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

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

X = data.drop('Species', axis=1)

y = data['Species']

print("Value of X=",X)

print( "Value of Y=",y)

Value of X= Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm

0 1 5.1 3.5 1.4 0.2

1 2 4.9 3.0 1.4 0.2

2 3 4.7 3.2 1.4 0.2

3 4 4.6 3.1 1.5 0.2

4 5 5.0 3.6 1.4 0.2

.. ... ... ... ... ...

145 146 6.7 3.0 5.2 2.3

146 147 6.3 2.5 5.0 1.9

147 148 6.5 3.0 5.2 2.0

148 149 6.2 3.4 5.4 2.3

149 150 5.9 3.0 5.1 1.8

[150 rows x 5 columns]

Value of Y= 0 Iris-setosa

1 Iris-setosa

2 Iris-setosa

3 Iris-setosa

4 Iris-setosa

...

145 Iris-virginica

146 Iris-virginica

147 Iris-virginica

148 Iris-virginica

149 Iris-virginica

Name: Species, Length: 150, dtype: object

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

print("X\_train shape:", X\_train.shape)

print("y\_train shape:", y\_train.shape)

X\_train shape: (120, 5)

y\_train shape: (120,)

model= GaussianNB()

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

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

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

print("Confusion Matrix:\n", cm)

Confusion Matrix:

[[10 0 0]

[ 0 9 0]

[ 0 0 11]]

TP = cm[0, 0] # True positives (class 0)

FP = cm[0, 1] # False positives (class 1)

TN = cm[1, 1] # True negatives (class 1)

FN = cm[1, 0]

accuracy = accuracy\_score(y\_test, y\_pred)

error\_rate = 1 - accuracy

precision = precision\_score(y\_test, y\_pred, average='macro')

recall = recall\_score(y\_test, y\_pred, average='macro')

f1 = f1\_score(y\_test, y\_pred, average='macro')

print(f"Accuracy: {accuracy:.4f}")

print(f"Error Rate: {error\_rate:.4f}")

print(f"Precision: {precision:.4f}")

print(f"Recall: {recall:.4f}")

print(f"F1 Score: {f1:.4f}")

Accuracy: 1.0000

Error Rate: 0.0000

Precision: 1.0000

Recall: 1.0000

F1 Score: 1.0000