import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import seaborn as sns

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

df.head()

X = df.iloc[:,:4].values

Y = df['Species'].values

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

from sklearn.preprocessing import StandardScaler

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 0)

sc\_X = StandardScaler()

X\_train = sc\_X.fit\_transform(X\_train)

X\_test = sc\_X.transform(X\_test)

print(f'Train Dataset Size - X: {X\_train.shape}, Y: {Y\_train.shape}')

print(f'Test Dataset Size - X: {X\_test.shape}, Y: {Y\_test.shape}')

from sklearn.naive\_bayes import GaussianNB

nvclassifier=GaussianNB()

nvclassifier.fit(X\_train,Y\_train)

Y\_pred=nvclassifier.predict(X\_test)

Y\_compare=np.vstack((Y\_test,Y\_pred)).T

Y\_compare[:4,:]

predictions=Y\_pred

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import classification\_report

cm = confusion\_matrix(Y\_test, predictions)

print(f'''Confusion matrix :\n

| Positive Prediction\t| Negative Prediction

---------------+------------------------+----------------------

Positive Class | True Positive (TP) {cm[0, 0]}\t| False Negative (FN) {cm[0, 1]}

---------------+------------------------+----------------------

Negative Class | False Positive (FP) {cm[1, 0]}\t| True Negative (TN) {cm[1, 1]}\n\n''')

cm = confusion\_matrix(Y\_test, predictions)

TP = cm[0, 0]

TN = cm[1, 1]

FP = cm[1, 0]

FN = cm[0, 1]

print(f"Accuracy : {(TP+TN)/(TP+FP+TN+FN)} ")

print(f'Error Rate: {(FP+FN)/(TP+TN+FN+FP)}')

print(f'Recall: {TN/(FP+TN)}')

print(f'Precision : {TP/(TP+FP)}')

print(f"True positive rate : {TP/(TP+FN)}")

print(f'False Positive Rate : {FP/(TN+FP)}')