Q7.KNN

**Write a program to implement K-Nearest neighbour algorithm to classify IRIS dataset. Printboth correct and wrong predictions using python ML libraries .classes can be used for this program**

CODE:

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

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import classification\_report, confusion\_matrix

from sklearn import datasets

iris = datasets.load\_iris()

iris\_data = iris.data

iris\_labels = iris.target

print(iris\_data)

print(iris\_labels)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(iris\_data, iris\_labels, test\_size=0.30)

clsify = KNeighborsClassifier(n\_neighbors=5)

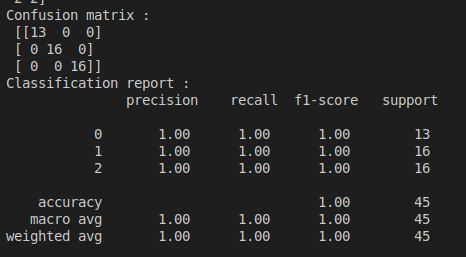
clsify.fit(x\_train, y\_train)

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

print("Confusion matrix : \n", confusion\_matrix(y\_test, y\_pred))

print("Classification report : \n", classification\_report(y\_test, y\_pred))

OUTPUT:



Q8.K Means

**Apply Em algorithm to cluster a set of data stored in a .csv file. Use the same data for clustering using the K-means algorithms. Compare the results of these two algorithms and comment on the quality of clustering. you can add python ML library classes/API in the program.**

CODE:

from sklearn.cluster import KMeans

from sklearn import preprocessing

from sklearn.mixture import GaussianMixture

from sklearn.datasets import load\_iris

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

datasets = load\_iris()

x = pd.DataFrame(datasets.data, columns=['Sepal\_length', 'Sepal\_width', 'Petal\_length', 'Petal\_width'])

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

colourmap = np.array(['red', 'green', 'blue'])

plt.figure(figsize=(14, 7))

plt.subplot(1, 3, 1)

plt.scatter(x.Petal\_length, x.Petal\_width, c=colourmap[y.Targets], s=40)

plt.title("Real")

kmeans = KMeans(n\_clusters=3)

kmeans.fit(x)

predy = np.choose(kmeans.labels\_, [0, 1, 2]).astype(np.int64)

plt.subplot(1, 3, 2)

plt.scatter(x.Petal\_length, x.Petal\_width, c=colourmap[predy], s=40)

plt.title('KMeans')

scaler = preprocessing.StandardScaler()

scaler.fit(x)

xsa = scaler.transform(x)

xs = pd.DataFrame(xsa, columns=x.columns)

gmm = GaussianMixture(n\_components=3)

gmm.fit(xs)

y\_cluster\_gmm = gmm.predict(xs)

plt.subplot(1, 3, 3)

plt.scatter(x.Petal\_length, x.Petal\_width, c=colourmap[y\_cluster\_gmm], s=40)

plt.title('GMM Classification')

plt.tight\_layout()

plt.show()

OUTPUT:

