10. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use

the same data set for clustering using the K-Means algorithm.

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

from sklearn import preprocessing

from sklearn import datasets

from sklearn.cluster import KMeans

from sklearn.mixture import GaussianMixture

# Load Iris dataset

iris = datasets.load\_iris()

x = pd.DataFrame(iris.data, columns=["sepal\_length", "sepal\_width",

"petal\_length", "petal\_width"])

# K-Means clustering

km = KMeans(n\_clusters=3)

km.fit(x)

km\_labels = km.labels\_

# Gaussian Mixture Model (GMM) clustering

scaler = preprocessing.StandardScaler()

xs = pd.DataFrame(scaler.fit\_transform(x), columns=x.columns)

gm = GaussianMixture(n\_components=3)

gm.fit(xs)

gmm\_y = gm.predict(xs)

# Plotting

15

plt.figure(figsize=(14, 5))

plt.subplot(1, 3, 1)

plt.scatter(x.petal\_length, x.petal\_width, c=iris.target, cmap='viridis', s=40)

plt.title("Real Plot")

plt.subplot(1, 3, 2)

plt.scatter(x.petal\_length, x.petal\_width, c=km\_labels, cmap='viridis', s=40)

plt.title("K-Means Plot")

plt.subplot(1, 3, 3)

plt.scatter(x.petal\_length, x.petal\_width, c=gmm\_y, cmap='viridis', s=40)

plt.title("GMM Plot")

plt.tight\_layout()

plt.show()