EXPERIMENT -7

**Aim:** Given a case study with data set(Remove the class label if any). You are expected to perform Clustering using Python/R/Java.

Quote your observations after the Clustering.

**Observations:**

**Using Python/R/Java:**

from sklearn.cluster import KMeans

import pandas as pd

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

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

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

from sklearn.decomposition import PCA

file = '/home/lavina/Desktop/pima-indians-diabetes-database/diabetes.csv'

dataFrame = pd.read\_csv(file, names=['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'])

X = dataFrame.loc[:, ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']]

y = dataFrame.loc[:, ['Outcome']]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30)

kmeans = KMeans(n\_clusters=2, random\_state=0, max\_iter=200).fit(X\_train)

print(kmeans.labels\_)

predictions = kmeans.predict(X\_test)

print(kmeans.cluster\_centers\_)

print(confusion\_matrix(y\_test, predictions))

print(classification\_report(y\_test, predictions))

x = StandardScaler().fit\_transform(X)

pca1 = PCA(n\_components=2)

principalComponents = pca1.fit\_transform(X)

principalDf = pd.DataFrame(data=principalComponents, columns=['pc1', 'pc2'])

X\_train1, X\_test1, y\_train1, y\_test1 = train\_test\_split(principalDf, y, test\_size=0.30)

kmeans = KMeans(n\_clusters=2, random\_state=0, max\_iter=200).fit(X\_train1)

print(kmeans.labels\_)

predictions = kmeans.predict(X\_test1)

print(kmeans.cluster\_centers\_)

print(confusion\_matrix(y\_test, predictions))

print(classification\_report(y\_test, predictions))

plt.scatter(principalDf.loc[:, 'pc1'], principalDf.loc[:, 'pc2'],s=50, cmap='viridis')

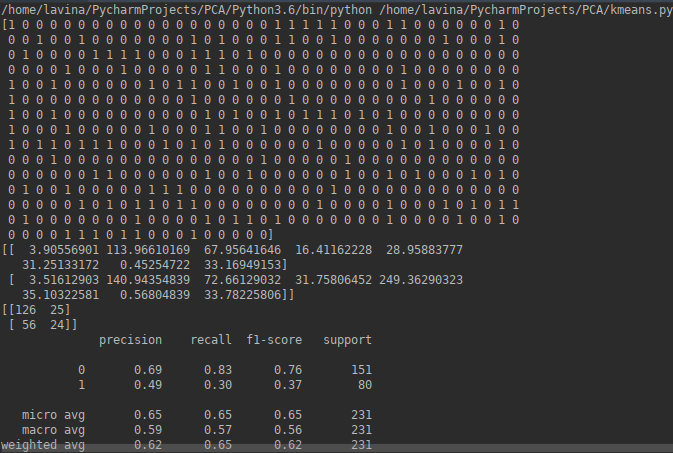
centers = kmeans.cluster\_centers\_

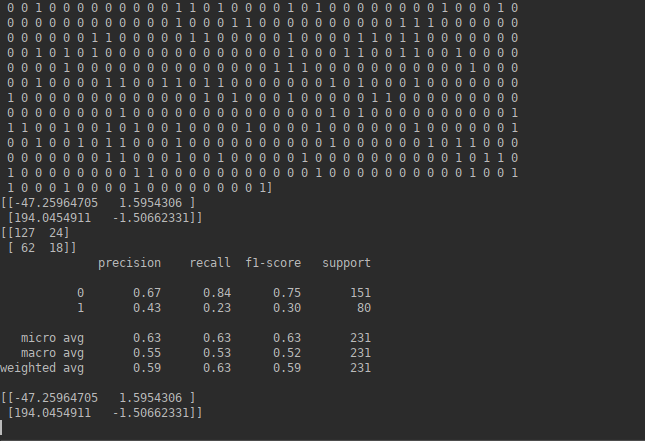
print(centers)

plt.scatter([centers[0][0], centers[1][0]], [centers[0][1], centers[1][1]], c='black', s=200, alpha=0.5)

plt.show()

**Output:**

****

****

**Conclusion:**Hence we have successfully demonstrated unsupervised learning with k means clustering.