## sample student data proximity matrix

## final dendrogramdendrogram threshold



## Implementing Hierarchical Clustering in Python

import pandas as pd

import matplotlib.pyplot as plt

data = pd.read\_csv('C:/Users/Admin/Desktop/Wholesale customers data.csv')

data.head()

#first normalize the data and bring all the variables to the same scale:

from sklearn.preprocessing import normalize

data\_scaled = normalize(data)

data\_scaled = pd.DataFrame(data\_scaled, columns=data.columns)

data\_scaled.head()

#Draw the dendrogram to help us decide the number of clusters for this particular problem:

import scipy.cluster.hierarchy as shc

plt.figure(figsize=(10, 7))

plt.title("Dendrograms")

dend = shc.dendrogram(shc.linkage(data\_scaled, method='ward'))

plt.figure(figsize=(10, 7))

plt.title("Dendrograms")

dend = shc.dendrogram(shc.linkage(data\_scaled, method='ward'))

plt.axhline(y=6, color='r', linestyle='--')

from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n\_clusters=2, affinity='euclidean', linkage='ward')

cluster.fit\_predict(data\_scaled)

plt.figure(figsize=(10, 7))

plt.scatter(data\_scaled['Milk'], data\_scaled['Grocery'], c=cluster.labels\_)

## Implement Hierarchical clustering

import matplotlib.pyplot as plt

import pandas as pd

customer\_data = pd.read\_csv('C:/Users/Admin/Desktop/shopping-data.csv')

customer\_data.shape

customer\_data.head()

data = customer\_data.iloc[:, 3:5].values

import scipy.cluster.hierarchy as shc

plt.figure(figsize=(10, 7))

plt.title("Customer Dendograms")

plt.xlabel('Customers')

plt.ylabel('Euclidean distances')

nd = shc.dendrogram(shc.linkage(data, method='ward'))

from sklearn.cluster import AgglomerativeClustering

cluster=AgglomerativeClustering(n\_clusters=5, affinity='euclidean',linkage='ward')

cluster.fit\_predict(data)

plt.figure(figsize=(10, 7))

plt.scatter(data[:,0], data[:,1], c=cluster.labels\_, cmap='rainbow')

## Implementing Gaussian Mixture Models in Python

import pandas as pd

import matplotlib.pyplot as plt

data = pd.read\_csv('C:/Users/Admin/Desktop/Clustering\_gmm.csv')

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

plt.scatter(data["Weight"],data["Height"])

plt.xlabel('Weight')

plt.ylabel('Height')

plt.title('Data Distribution')

plt.show()

#build a k-means model

**#training k-means model**

from sklearn.cluster import KMeans

kmeans = KMeans(n\_clusters=4)

kmeans.fit(data)

**#predictions from kmeans**

pred = kmeans.predict(data)

frame = pd.DataFrame(data)

frame['cluster'] = pred

frame.columns = ['Weight', 'Height', 'cluster']

**#plotting results**

color=['blue','green','cyan', 'black']

for k in range(0,4):

data = frame[frame["cluster"]==k]

plt.scatter(data["Weight"],data["Height"],c=color[k])

plt.show()

#build a Gaussian Mixture Model

import pandas as pd

data = pd.read\_csv('C:/Users/Admin/Desktop/Clustering\_gmm.csv')

**# training gaussian mixture model**

from sklearn.mixture import GaussianMixture

gmm = GaussianMixture(n\_components=4)

gmm.fit(data)

**#predictions from gmm**

labels = gmm.predict(data)

frame = pd.DataFrame(data)

frame['cluster'] = labels

frame.columns = ['Weight', 'Height', 'cluster']

color=['blue','green','cyan', 'black']

for k in range(0,4):

data = frame[frame["cluster"]==k]

plt.scatter(data["Weight"],data["Height"],c=color[k])

plt.show()