**CA 614 – Data Mining and Analytics**

**Assignment – IV**

**Clustering**

1. **Read the Wholesale\_customers\_data.csv file.**

import pandas as pd

df = pd.read\_csv("C:\\Users\\student\\Downloads\\wholesale\_customers\_data.csv")

print(df)

df.info()

df.info()

print(df)

df.describe()

df.isnull()

1. **Perform all data pre-processing techniques.**

df.info()

df.isnull()

df.isnull().sum()

df.isnull().sum()

1. **Perform one-hot encoding and normalization.**

#Task : To categorise the countries using socio-economic and health factors

#that determine the overall development of the country.

import numpy as np

import pandas as pd

import statsmodels.api as sm

import random as rd

import matplotlib.pyplot as plt

import seaborn as sns

sns.set()

from sklearn.cluster import KMeans

data = pd.read\_csv('C:\\Users\\student\\Downloads\\wholesale\_customers\_data.csv')

data

#Apply one hot encoding to categorical column

data1 = pd.get\_dummies(data, columns = ['Channel','Region'])

data1

#Apply normalization to the dataframe as k-means is distance-based algorithm

from sklearn import preprocessing

data2 = preprocessing.normalize(data1)

data2

1. **Apply k-means clustering algorithm.**

kmeans = KMeans(5)

kmeans.fit(data2)

pred = kmeans.predict(data2)

pred

frame = pd.DataFrame(data2)

frame['cluster'] = pred

frame['cluster'].value\_counts()

from sklearn.preprocessing import OneHotEncoder

OneHotEncoder().fit\_transform(data2)

#you cannot do it directly if you have more than 3 columns. However, you can apply a Principal Component Analysis to reduce the space in 2 columns and visualize this instead.

from sklearn.decomposition import PCA

pca\_num\_components = 2

reduced\_data = PCA(n\_components=pca\_num\_components).fit\_transform(data2)

results = pd.DataFrame(reduced\_data,columns=['pca1','pca2'])

sns.scatterplot(x="pca1", y="pca2", hue=frame['cluster'], data=results)

plt.title('K-means Clustering with 2 dimensions')

plt.show()

1. **Apply hierarchical clustering algorithm.**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

#Our aim is to make clusters from this data that can segment similar clients together.

data = pd.read\_csv('C:\\Users\\91972\\Downloads\\Wholesale\_customers\_data.csv')

data.head()

#before applying Hierarchical Clustering,

#we have to normalize the data so that the scale of each variable is the same.

#if the scale of the variables is not the same,

#the model might become biased towards the variables

#with a higher magnitude like Fresh or Milk.

#So, let’s 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()

import scipy.cluster.hierarchy as shc

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

plt.title("Dendrograms")

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

#We have two clusters as this line cuts the dendrogram at two points.

#Let’s now apply hierarchical clustering for 2 clusters:

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['Frozen'], c=cluster.labels\_)