Perform clustering (hierarchical,K means clustering and DBSCAN) for the airlines data to obtain optimum number of clusters.

Draw the inferences from the clusters obtained.

Data Description:

The file EastWestAirlinescontains information on passengers who belong to an airline’s frequent flier program. For each passenger the data include information on their mileage history and on different ways they accrued or spent miles in the last year. The goal is to try to identify clusters of passengers that have similar characteristics for the purpose of targeting different segments for different types of mileage offers

ID --Unique ID

Balance--Number of miles eligible for award travel

Qual\_mile--Number of miles counted as qualifying for Topflight status

cc1\_miles -- Number of miles earned with freq. flyer credit card in the past 12 months:

cc2\_miles -- Number of miles earned with Rewards credit card in the past 12 months:

cc3\_miles -- Number of miles earned with Small Business credit card in the past 12 months:

1 = under 5,000

2 = 5,000 - 10,000

3 = 10,001 - 25,000

4 = 25,001 - 50,000

5 = over 50,000

Bonus\_miles--Number of miles earned from non-flight bonus transactions in the past 12 months

Bonus\_trans--Number of non-flight bonus transactions in the past 12 months

Flight\_miles\_12mo--Number of flight miles in the past 12 months

Flight\_trans\_12--Number of flight transactions in the past 12 months

Days\_since\_enrolled--Number of days since enrolled in flier program

Award--whether that person had award flight (free flight) or not

ANS: import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import scipy.cluster.hierarchy as sch

from sklearn.cluster import AgglomerativeClustering

from sklearn.preprocessing import normalize

Air=pd.read\_csv("E:\\EastWestAirlines.csv")

Air.head()

Air.info()

Air2=Air.drop(['ID#'],axis=1)

Air2

Air2\_norm=pd.DataFrame(normalize(Air2),columns=Air2.columns)

Air2\_norm

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

dendograms=sch.dendrogram(sch.linkage(Air2\_norm,'complete'))

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

hclusters

y=pd.DataFrame(hclusters.fit\_predict(Air2\_norm),columns=['clustersid'])

y['clustersid'].value\_counts()

Air2['clustersid']=hclusters.labels\_

Air2

Air2.groupby('clustersid').agg(['mean']).reset\_index()

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

plt.scatter(Air2['clustersid'],Air2['Balance'], c=hclusters.labels\_)