**Business objective:** To identify customer segments via **clustering.Inference**: The main objective is to segment the customers based on their flying patterns,frequently use of rewards and use of airline credit.This is a large data set, this contains 4000 rows and 12 columns. After importing the data and normalizing all the columns except ‘ID’ and after applying ‘manhattan’ distance and to find the distance ,I have experimented various combinations “ centroid ”linkage is showing me some good results .Hence I have created 3 clusters cluster(0,1,2).

**Group.1 Balance Qual\_miles cc1\_miles cc2\_miles**

**1 1 73134.42 143.9474 2.059339 1.014522**

**2 2 1704838.00 0.0000 1.000000 1.000000**

**3 3 131999.50 347.0000 2.500000 1.000000**

So,here the group 3 people are the people who travel most frequently Based on number of miles counted as qualifying for Topflight status,group 3 customers also earn moe cc1 miles .Then comes group 2 where the people are least travelled.They earn very less cc1 miles.Then comes group customers where are more frequentlu travelling customers , have a average earned cc1,2 miles.

**#R code for agglomerative clustering**

library(readr)

library(readxl)

library(readr)

air\_data<- read\_excel("E:/Assignments/Assignments week 2/m4/EastWestAirlines (2).xlsx",2)

View(air\_data)

summary(air\_data)

normalize<-scale(air\_data[,2:12])

View(normalize)

summary(normalize)

d<- dist(normalize,method='manhattan')

fit<-hclust(d,method="centroid")

plot(fit)

plot(fit,hang=-1)

groups<-cutree(fit,k=3)

rect.hclust(fit,k=3,border='red')

customers<-as.matrix(groups)

final\_segmentation<-data.frame(customers,air\_data)

View(air\_data)

View(final\_segmentation)

aggregate(air\_data[,2:5],by=list(final\_segmentation$customers),FUN = mean)

library(readr)

write\_csv(final\_segmentation,"customersegmentationcent.csv")

getwd()

**clustering.Inference**: The main objective is to segment the customers based on their flying patterns,frequently use of rewards and use of airline credit.This is a large data set, this contains 4000 rows and 12 columns. After importing the data and normalizing all the columns except ‘ID’ and after applying ‘manhattan’ distance and to find the distance ,I have experimented various combinations “ centroid ”linkage is showing me some good results .Hence I have created 3 clusters cluster(0,1,2).

Qual\_miles cc1\_miles cc2\_miles Days\_since\_enroll ranking

239.728387 2.670524 1.007488 4628.761743 0

347.000000 2.500000 1.000000 2200.250000 1

88.188836 1.703484 1.018606 3824.887965 2

So, by the given data we can say that the rank 1 people are the most frequently travelled customers as they have earned more qual\_miles .Next are the rank 0 customers who also travel frequently. Then comes the rank 2 customers where they travel the least based on the qualmiles.

**#Python code for agglomerative clustering**

import pandas as pd

from pandas import ExcelWriter

from pandas import ExcelFile

import matplotlib.pylab as plt

air\_data=pd.read\_excel('E:/data/EastWestAirlines.xlsx')

air\_data.describe()

#Normalize function

def norm\_func(i):

x=(i-i.min()) / (i.max()-i.min())

return(x)

df\_norm=norm\_func(air\_data.iloc[:,1:])

df\_norm.describe()

#for creating dendogram

from scipy.cluster.hierarchy import linkage

import scipy.cluster.hierarchy as sch

k=linkage(df\_norm,method="complete",metric="Euclidean")

#dendogram

plt.figure(figsize=(15,8));plt.title('Agglomerative clustering');plt.xlabel('customers');plt.ylabel('air\_data')

sch.dendrogram(k,leaf\_rotation=0,leaf\_font\_size=10)

plt.show()

#apply agglomerative clustering

from sklearn.cluster import AgglomerativeClustering

h\_complete=AgglomerativeClustering(n\_clusters=3,linkage='complete',affinity="euclidean").fit(df\_norm)

h\_complete.labels\_

cluster\_labels=pd.Series(h\_complete.labels\_)

air\_data['ranking']=cluster\_labels

final\_data=air\_data.iloc[:, [12,0,1,2,3,4,5,6,7,8,9,10,11]]

air\_data.head()

#aggregate mean of each cluster

air\_data.iloc[:, 2:].groupby(final\_data.ranking).mean()

#creating a csv file

air\_data.to\_csv("airdatapython2.csv",encoding="utf-8")

import os

os.getcwd()