Full codes:

# -\*- coding: utf-8 -\*-

"""

Created on Mon Oct 12 18:56:11 2020

@author: sunil

"""

#-----------------HIERARICAL CLUSTERING--------------------#

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from scipy.spatial.distance import cdist

airline = pd.read\_csv("E://Excelr//excler//Hierachial and kmeans clustering//ewa.csv")

# Normalized data frame (considering the numerical part of data)

from sklearn.preprocessing import scale

#normalize data frame

df\_norm=pd.DataFrame(scale(airline.iloc[:,1:7]))

df\_norm.head(5)

from scipy.cluster.hierarchy import linkage

import scipy.cluster.hierarchy as sch # for creating dendrogram

type(df\_norm)

#p = np.array(df\_norm) # converting into numpy array format

help(linkage)

zz = linkage(df\_norm, method="complete",metric="euclidean")

help(linkage)

plt.figure(figsize=(15, 5));plt.title('Hierarchical Clustering Dendrogram');plt.xlabel('Index');plt.ylabel('Distance')

sch.dendrogram(

zz,

leaf\_rotation=0., # rotates the x axis labels

leaf\_font\_size=8., # font size for the x axis labels

)

plt.show()

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\_)

cluster\_labels.value\_counts()

#creating new column and assigning into it

airline['clust']=cluster\_labels

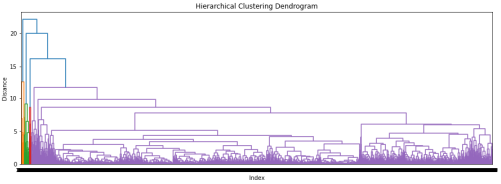
airline=airline.iloc[:,:]

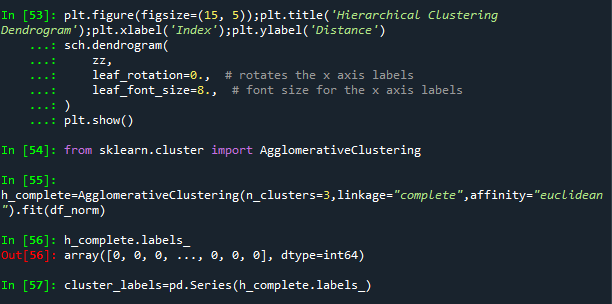
airline.head()

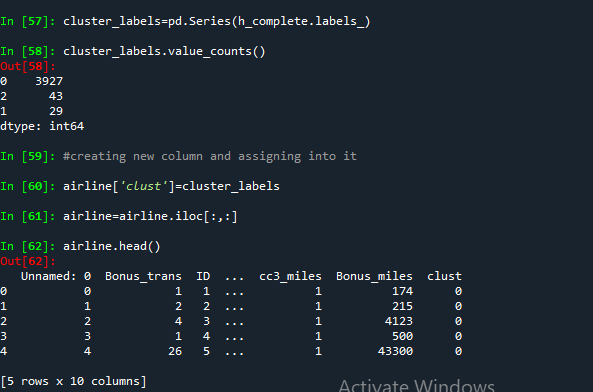
#getting aggregrate mean of each cluster

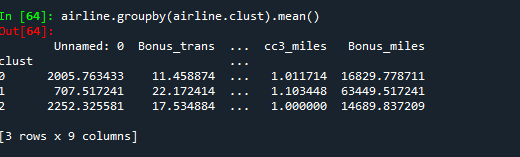
airline.groupby(airline.clust).mean()

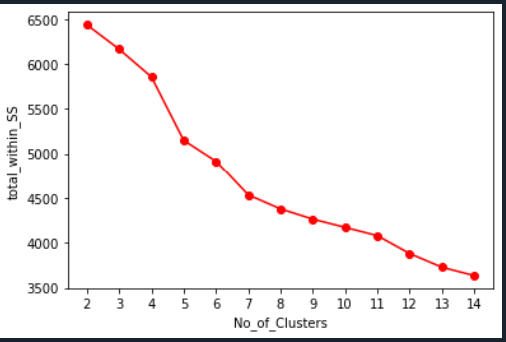
airline.to\_csv("E://Excelr//excler//Hierachial and kmeans clustering//ewa.csv",index=False) #,encoding="utf-8")

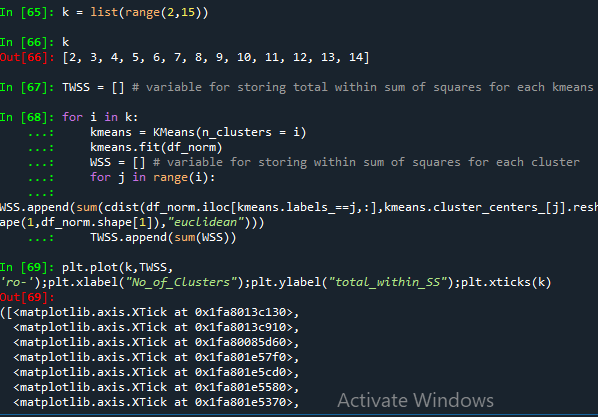


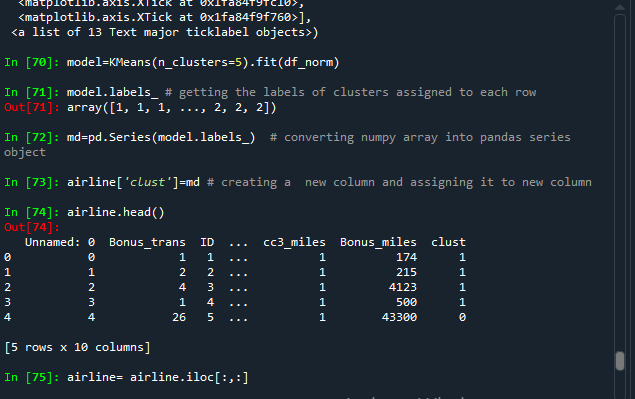


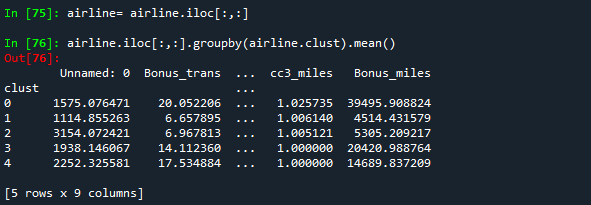




KMEANS CLUSTERING  








#-------------------K-MEANS CLUSTERING----------------------#

k = list(range(2,15))

k

TWSS = [] # variable for storing total within sum of squares for each kmeans

for i in k:

kmeans = KMeans(n\_clusters = i)

kmeans.fit(df\_norm)

WSS = [] # variable for storing within sum of squares for each cluster

for j in range(i):

WSS.append(sum(cdist(df\_norm.iloc[kmeans.labels\_==j,:],kmeans.cluster\_centers\_[j].reshape(1,df\_norm.shape[1]),"euclidean")))

TWSS.append(sum(WSS))

# Scree plot

plt.plot(k,TWSS, 'ro-');plt.xlabel("No\_of\_Clusters");plt.ylabel("total\_within\_SS");plt.xticks(k)

# Selecting 5 clusters from the above scree plot which is the optimum number of clusters

model=KMeans(n\_clusters=5).fit(df\_norm)

model.labels\_ # getting the labels of clusters assigned to each row

md=pd.Series(model.labels\_) # converting numpy array into pandas series object

airline['clust']=md # creating a new column and assigning it to new column

airline.head()

airline= airline.iloc[:,:]

airline.iloc[:,:].groupby(airline.clust).mean()

airline.to\_csv("E://Excelr//excler//Hierachial and kmeans clustering//ewa.csv") # .to\_csv() is like write.csv() in r