Assignment 22.1

Name : Y Vasudev

Batch : DA with R , Excel and Tableau

1. Use the below given data set

DataSet

2. Perform the below given activities:

a. apply K-means clustering to identify similar recipies

b. apply K-means clustering to identify similar attributes

c. how many unique recipies that people order often

d. what are their typical profiles

#1- HIERARCHICAL CLUSTERING (Agglomorative and Divisive)

#2- PARTITION BASED (kmeans, k-median, model based, graph based)

#3- BAYSIAN CLUSTERING (decision rule based, non paramteric)

df = read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data",sep=',',header=F)

#problem: find out natual grouping or classes of wine quality

head(df)

str(df)

set.seed(1234)

ind = sample(1:nrow(df),0.8\*nrow(df),replace = F)

df\_train =df[ind,-1]

df\_test = df[-ind,-1]

summary(df)

dim(df)

# outlier definition

# x > Q3+1.5\*IQR - positive side outlier

# x < Q1-1.5\*IQR - negative or lower side outlier

par(mfrow=c(2,3))

(boxplot(df$V2)$out);(boxplot(df$V3)$out);(boxplot(df$V4)$out);(boxplot(df$V5)$out);(boxplot(df$V6)$out)

(boxplot(df$V7)$out);(boxplot(df$V8)$out);(boxplot(df$V9)$out);(boxplot(df$V10)$out)

apply(df,2,range)

apply(df,2,summary)

# KMeans - comes from Rcmdr library

# Kmeans- from amap library

# kmeans- from stats library

# steps in k-means clustering

#1- preprocessing the data (impute missing values, remove outliers, feature trasnformation)

#2- scaling or standardization of data set

#3- decide the number of clusters (value of K)

#4- iterate over the samples to create clusters

#5- decide the distance measure

#6- calculate the group accuracy

# scaling of data

df\_train1 <- scale(df\_train)

head(df\_train1)

class(df\_train1)

# screeplot approach to decide the number of clusters

km = kmeans(df\_train1,1)

km$withinss

km$tot.withinss

km = kmeans(df\_train1,2)

km$withinss

km$tot.withinss

km = kmeans(df\_train1,3)

km$withinss

km$tot.withinss

km = kmeans(df\_train1,4)

km$withinss

km$tot.withinss

km = kmeans(df\_train1,5)

km$withinss

km$tot.withinss

km = kmeans(df\_train1,6)

km$withinss

km$tot.withinss

km = kmeans(df\_train1,7)

km$withinss

km$tot.withinss

km = kmeans(df\_train1,8)

km$withinss

km$tot.withinss

km = kmeans(df\_train1,9)

km$withinss

km$tot.withinss

km = kmeans(df\_train1,10)

km$withinss

km$tot.withinss

dev.off()

sumsq=NULL

for (i in 1:15)

sumsq[i] = sum(kmeans(df\_train,centers=i,

iter.max = 1000,

nstart=i,

algorithm='Forgy')$withinss)

plot(1:15,sumsq,type='b', main='Screeplot showing within group sum of squares')

km = kmeans(df\_train1,3)

km$withinss

km$tot.withinss

class(km$cluster)

summary(km)

km$centers

as.numeric(km$cluster)

length(km$cluster)

dim(df\_train)

class(df\_train)

df\_train$cl <- km$cluster

head(df\_train)

# profiles of clusters

aggregate(df\_train[,1:13],list(df\_train[,14]),mean)

table(df$V1)

library(cluster)

clusplot(df\_train,df\_train$cl,cex=0.9,color=T,shade=T, labels=4,lines=0)

#HC clustering or Hierarchical Clustering

# distance (euclidean, manhattan, cosine distance)

# Divisive method (top down)

# Agglomorative method (bottom up)

df\_train = df\_train[,-14]

head(df\_train)

# compute the distance metrix

d1 <- dist(df\_train,method='euclidean')

summary(d1)

# HC

fit <- hclust(d1,method = 'ward.D2')

plot(fit)

# single, double, average, ward, ward.D2

# agglomorative method

fit <- agnes(d1,metric='euclidean',method = 'ward')

plot(fit)

# divisive method

fit <- diana(d1,metric='euclidean')

plot(fit)