**Name – Souvik Samanta**

**Roll no – 19PGDM064**

**Problem 1:**

1. **R code –**

*# Setting working directory*

*setwd("C:/BA")*

*# Reading the data from csv file*

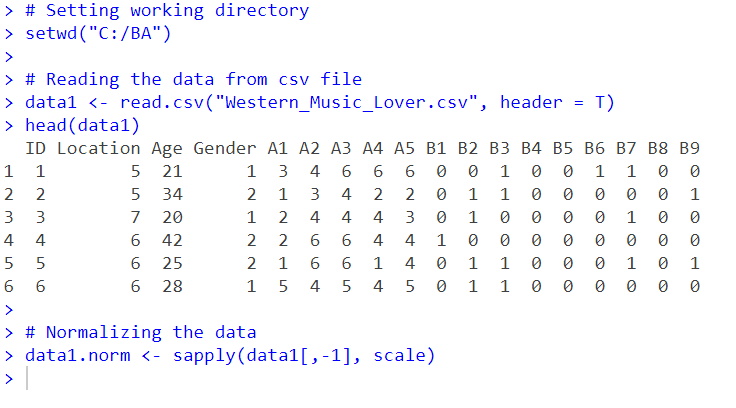
*data1 <- read.csv("Western\_Music\_Lover.csv", header = T)*

*head(data1)*

*# Normalizing the data*

*data1.norm <- sapply(data1[,-1], scale)*

**Output –**

****

Normalization is a very important step in clustering. Basically, normalization transforms the data within -1 to 1 without affecting the relative differences between the values. In our case various variables have different ranges. To quantify them within same range, we need to normalize them otherwise we will have partiality in our cluster as the ranges will differ otherwise.

1. **R code –**

*# Applying k-means clustering for 3 clusters*

*library(caret)*

*km <- kmeans(data1.norm, 3)*

*km*

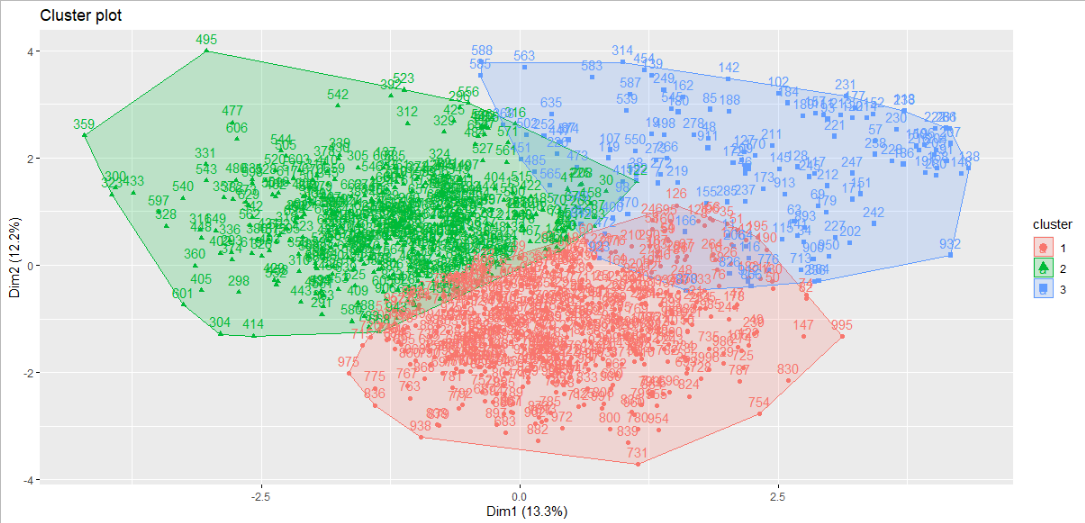
*table(km$cluster)*

*library(factoextra)*

*library(ggpubr)*

*fviz\_cluster(km, data = data1.norm)*

**Output –**



In the plot we can see the 3 clusters

1. **R code –**

*plot(c(0), xaxt = 'n', xlab = "", ylab = "", type = "l",*

*ylim = c(min(km$centers), max(km$centers)), xlim = c(0,17))*

*axis(1, at = c(1:17), labels = names(data1[,-1]))*

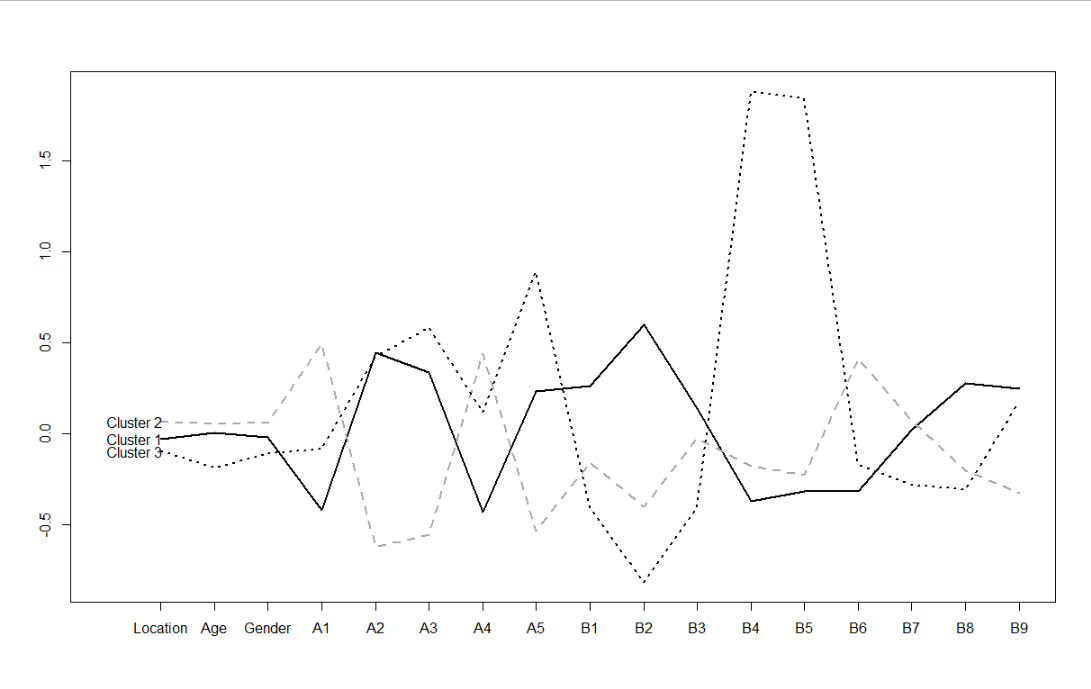
*for (i in c(1:3)){*

*lines(km$centers[i,], lty = i, lwd = 2,*

*col = ifelse(i %in% c(1,3,5), "black", "dark grey"))}*

*text(x = 0.5, y = km$centers[, 1], labels = paste("Cluster", c(1:3)))*

**Output –**



From the above plot we can see the variances in different parameters for the clusters. Followings are the characteristics of the clusters –

* **Cluster 1 –** For this cluster, people feel more energised and relaxed. They generally do not take the western music as a matter of entertainment; they enjoy the music by their heart and feel relaxed and motivated. We can label this cluster as *“Western music lovers”.*
* **Cluster 2 –** This cluster depicts that; they just prefer western music for nothing. They do not enjoy that music or feel anything special by listening to these kinds of music. They just find it cool to listen to them where they may not enjoy that for sure. We can label them as ”Casual *listeners”.*
* **Cluster 3 –** This cluster clearly shows a higher peak in B4 and B5 which simply means that they feel bored and tired while they listen to western music. A sharp downward centroid at b2 also infers that they never feel energized when they listen to it. So we can label them as *“Western music haters”.*

1. **Cluster 1** should be targeted to give more offers because –

* They have a strong feeling attached to the western music as they feel motivated and energised when they listen to western music.
* They do not just take it for feeling smart or being cool. Also, they do not simply listen for only entertainment purpose. They love the music by heart.

The offers may be to give more discounts to them for specially western music.

**Prob 2 –**

**R code –**

*# # Reading the data from csv file*

*data2 <- read.csv("Diabetes.csv", header = T)*

*head(data2)*

*str(data2)*

*data2$Outcome <- factor(data2$Outcome)*

*set.seed(111)*

*# Partitioning the data into train and validation sets*

*train.index <- sample(row.names(data2), 0.6\*dim(data2)[1])*

*valid.index <- setdiff(row.names(data2), train.index)*

*train.df <- data2[train.index, ]*

*valid.df <- data2[valid.index, ]*

*# Normalizing the data*

*train.norm.df <- train.df*

*valid.norm.df <- valid.df*

*data2.norm <- data2*

*library(caret)*

*norm.values <- preProcess(train.df[, 1:8], method = c("center", "scale"))*

*train.norm.df[, 1:8] <- predict(norm.values, train.df[, 1:8])*

*valid.norm.df[, 1:8] <- predict(norm.values, valid.df[, 1:8])*

*data2.norm[, 1:8] <- predict(norm.values, data2[, 1:8])*

*# Using KNN*

*library(FNN)*

*accuracy <- data.frame(k = seq(1,20,1), Accuracy = rep(0, 20))*

*for (i in 1:20){*

*nn <- knn(train = train.norm.df[, 1:8], test = valid.norm.df[, 1:8],*

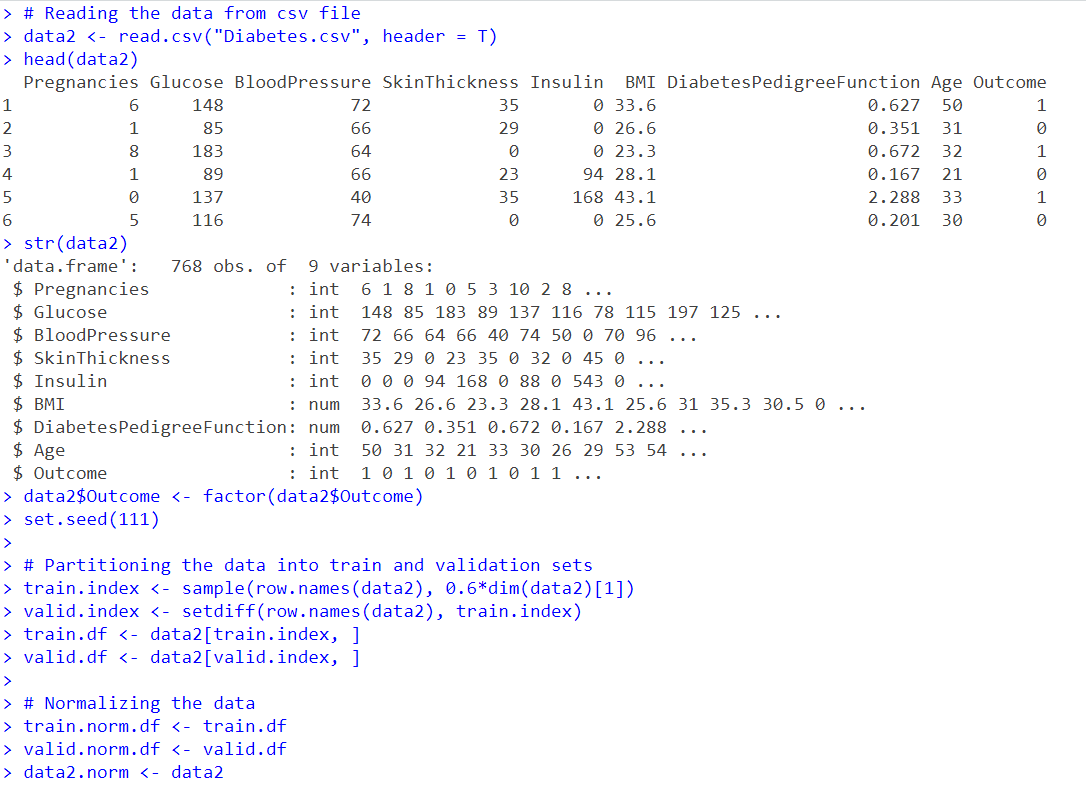
*cl = train.norm.df[, 9], k = i)*

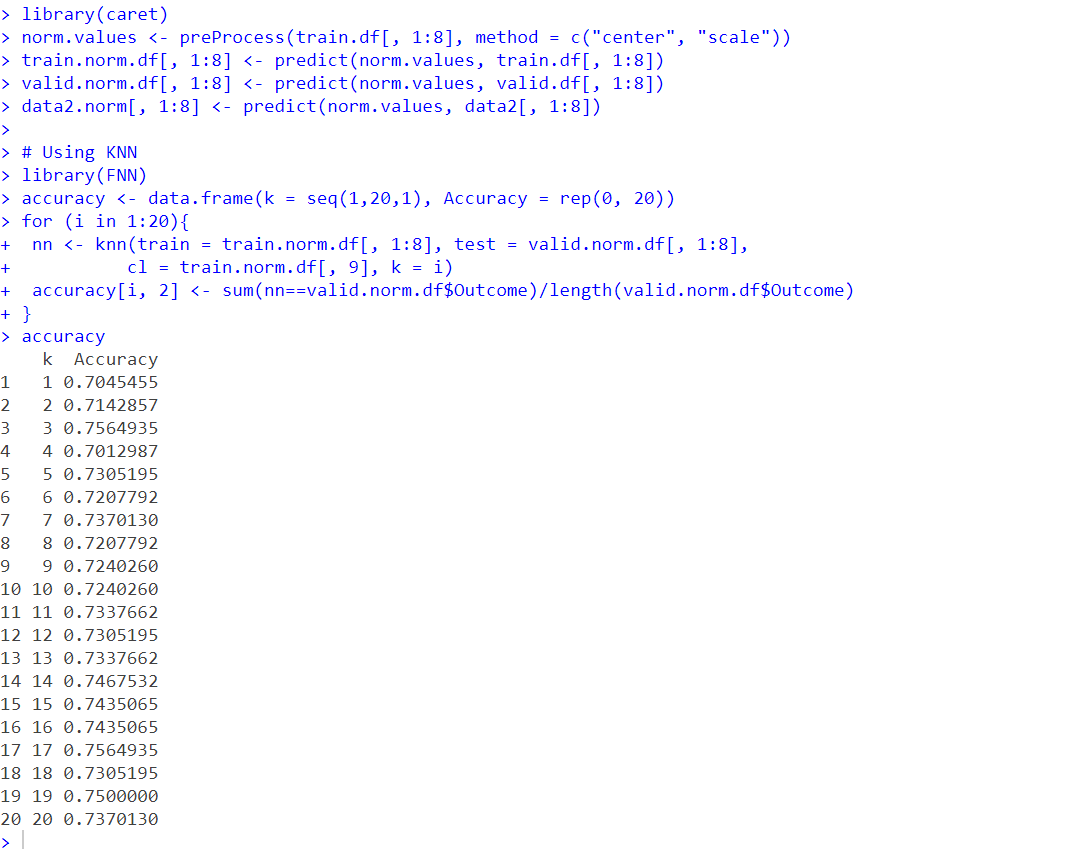
*accuracy[i, 2] <- sum(nn==valid.norm.df$Outcome)/length(valid.norm.df$Outcome)*

*}*

*accuracy*

**Output –**

****

****

1. From the above code and output, we can see how we have applied kNN algorithm for the 8 predictor variables. We have taken multiple values of k ranging from 1 to 20 and for all of them, we have estimated the accuracies.
2. From the output, we can see the best k value is 3 and 17 as they are giving same accuracy which are higher than all other k values. However, comparing 17 and 3, if we take k=3, it means noise would have higher influence on outcome. As we have significantly good number of samples, 17 will produce good results. So, best value of k is 17. This means, the model would measure distance from 17 nearby points to classify a new input. Here, k=17 would give good results as it is going to measure 17 nearby distances.
3. As calculated in the table, for k=17, the accuracy for the same would be 0.7564935 or simply 75.64935%