Assignment 5

Yash Bhanushali

04/16/2022

## Problem Definition

Elementary school cafeterias would like to select a set of cereals to include in their daily menus. A new cereal is served each day, although all cereals should contribute to a balanced diet. The purpose of this task is to locate the “healthy cereals” cluster.

## Importing R libraries

library(readr)  
library(tidyverse)  
library(cluster)  
library(caret)  
library(dendextend)  
library(factoextra)  
library(RColorBrewer)

## Importing Dataset

77 morning cereals are included in the ‘Cereals.csv’ dataset, which includes nutritional information, store display, and consumer ratings.

Cereals <- read\_csv("D:/MSBA/Fundamentals of Machine Learning/ML assignment 5/Cereals.csv")  
# Examining the dataset  
View(Cereals)

1 - Applied hierarchical clustering to the data using Euclidean distance to the normalized measurements, and using Agnes to compare the clustering from single, complete, average, and Ward linkage methods and choosing the best method.

## Data Preparation

## Data cleaning and Scaling

#Checking NULL values in the dataset at column level.  
colSums(is.na(Cereals))

## name mfr type calories protein fat sodium fiber   
## 0 0 0 0 0 0 0 0   
## carbo sugars potass vitamins shelf weight cups rating   
## 1 1 2 0 0 0 0 0

#Removing missing values which are present in the Cereals dataset  
Cereals <- na.omit(Cereals)  
#Using only the numerical variables for clustering  
Cereals\_numeric <- Cereals %>% select\_if(is.numeric)  
head(Cereals\_numeric)

## # A tibble: 6 x 13  
## calories protein fat sodium fiber carbo sugars potass vitamins shelf weight  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 70 4 1 130 10 5 6 280 25 3 1  
## 2 120 3 5 15 2 8 8 135 0 3 1  
## 3 70 4 1 260 9 7 5 320 25 3 1  
## 4 50 4 0 140 14 8 0 330 25 3 1  
## 5 110 2 2 180 1.5 10.5 10 70 25 1 1  
## 6 110 2 0 125 1 11 14 30 25 2 1  
## # ... with 2 more variables: cups <dbl>, rating <dbl>

#Scaling the dataset using (Z-Score) standardization   
scaled\_cereals <- as.data.frame(scale(Cereals\_numeric))

## Model Construction

We can conclude from the problem definition that this challenge falls under the category of “Unsupervised Learning”. As a result, I attempted to uncover patterns and classify comparable objects into clusters using the “Hierarchical Clustering” technique.

##Analyzing clustering from “Single”, “Complete”, “Average”, and “Ward” linkage approaches using Agnes method..

# methods to assess  
m <- c( "average", "single", "complete", "ward")  
names(m) <- c( "average", "single", "complete", "ward")

##Using a function to calculate the linkage methods’ coefficients. The function argument accepts a “character vector(x)” as an input, which corresponds to the “agnes” function’s argument “method.”

ac <- function(x) {  
 agnes(scaled\_cereals, metric = "euclidean", method = x)$ac  
}

##Mapping character vector and ac function using map function which return the vector of linkage coefficients.

map\_dbl(m, ac)

## average single complete ward   
## 0.7766075 0.6067859 0.8353712 0.9046042

##From above Agnes function we can see that “r names(which.max(map\_dbl(m, ac)))” linkage has strong clustering structure, with agglomerative coefficient of “r round(max(map\_dbl(m, ac)), 2)”, So choosing “r names(which.max(map\_dbl(m, ac)))” linkage method\*\* for further cluster analysis.

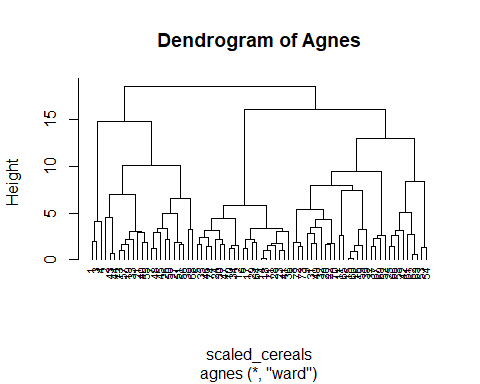
Question 2 - Estimating the optimal number of clusters.

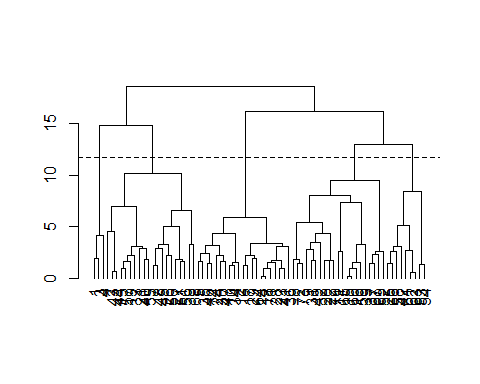
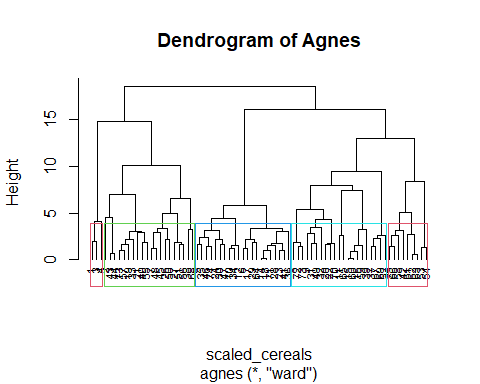
# Hierarchical clustering using Ward Linkage  
hc\_cereals <- agnes(scaled\_cereals, method = "ward")

## Visualizing the Dendogram

##Passing model object “hc\_cereals” to “pltree” to produce “dendogram”.

pltree(hc\_cereals, cex = 0.7, hang = -1, main = "Dendrogram of Agnes")



##From below dendrogram, we observe that cut associated with largest gaps generates “2” clusters.  
 

##Hierarchical clustering is used to determine the optimal number of clusters. This optimal number of clusters can be determined by looking at the largest difference of heights. So from above analysis choosing optimal number of clusters “k = 5”

Question 3 - Checking Cluster stability

# Cutting the tree  
cluster\_assignment <- cutree(hc\_cereals, k=5)  
cereals\_data\_clustered <- mutate(scaled\_cereals, cluster = cluster\_assignment)  
# partitioning the cluster  
set.seed(150)  
index <- createDataPartition(cereals\_data\_clustered$cluster, p = 0.7, list = FALSE)  
part\_A <- cereals\_data\_clustered[index,]  
part\_B <- cereals\_data\_clustered[-index,]  
# Finding cluster centroid for partition A  
part\_A\_centroids <- part\_A %>% gather("features","values",-cluster) %>%   
 group\_by(cluster,features) %>% summarise(mean\_values = mean(values)) %>%   
 spread(features,mean\_values)  
cluster\_prediction\_B <- data.frame(data=seq(1,nrow(part\_B),1),  
 Partition\_B\_cluster=rep(0,nrow(part\_B)))  
# Here row binding each test data datapoint to partition a centroids,   
# and finding the minmum distance from each cluster centroid.  
for (x in 1:nrow(part\_B)) {  
 cluster\_prediction\_B$Partition\_B\_cluster[x] <-  
   
 which.min(as.matrix(get\_dist(as.data.frame(  
 rbind(part\_A\_centroids[-1], part\_B[x, -length(part\_B)])  
 )))[6, -6])  
}  
# Comparing Partition B data labels with the original data labels.  
cluster\_prediction\_B <- cluster\_prediction\_B %>% mutate(original\_clusters = part\_B$cluster)  
mean(cluster\_prediction\_B$Partition\_B\_cluster == cluster\_prediction\_B$original\_clusters)

## [1] 1

##According to the results of the preceding analysis, the original and anticipated clusters are identical. As a result, conculding clusters are quite stable.

*Finding a cluster of “healthy cereals.”*

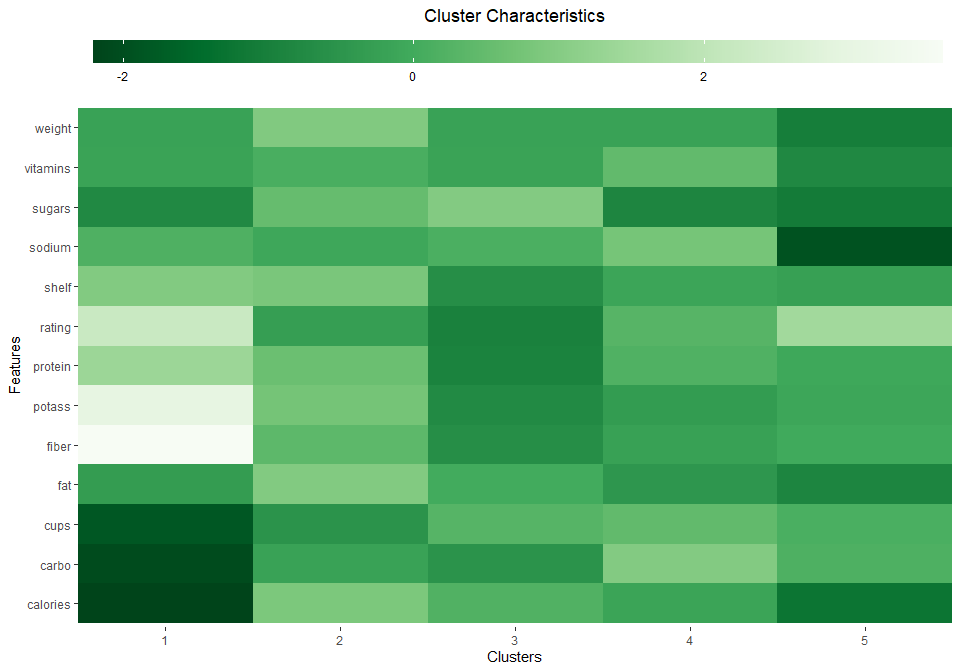
##Finding centroids of each cluster to determined the cluster characteristics.

split\_data <- split(cereals\_data\_clustered, cereals\_data\_clustered$cluster)  
split\_means <- lapply(split\_data, colMeans)  
(centroids <- do.call(rbind, split\_means))

## calories protein fat sodium fiber carbo  
## 1 -2.2018711 1.38174776 -0.3310734 0.17279012 3.64131237 -2.0718749  
## 2 0.8553248 0.59163927 0.9435592 -0.08898011 0.38141771 -0.2003584  
## 3 0.1978117 -0.91996886 0.0000000 0.12101140 -0.66198437 -0.5423583  
## 4 -0.1621407 0.18662567 -0.4729620 0.77112209 -0.21003997 0.9626860  
## 5 -1.2499969 -0.06420242 -0.8828625 -1.94150793 -0.02664224 0.1551013  
## sugars potass vitamins shelf weight cups rating  
## 1 -0.7894824 2.9837813 -0.18184220 0.9419715 -0.2008324 -1.8452553 2.2426479  
## 2 0.5143002 0.7475659 0.09849786 0.8217889 0.9235649 -0.5477863 -0.2928786  
## 3 0.9583619 -0.7415648 -0.18184220 -0.6604628 -0.2008324 0.2779676 -0.9636465  
## 4 -0.8659505 -0.3485391 0.45893508 -0.1453946 -0.2008324 0.4577648 0.2916795  
## 5 -1.0953551 -0.1122758 -0.80482011 -0.2598542 -1.0482044 0.1156788 1.4712151  
## cluster  
## 1 1  
## 2 2  
## 3 3  
## 4 4  
## 5 5

##Visualizing the clusters

hm.palette <-  
 colorRampPalette(rev(brewer.pal(9, 'Greens')), space = 'Lab')  
data.frame(centroids) %>% gather("features", "values",-cluster) %>%  
 ggplot(aes(  
 x = factor(cluster),  
 y = features,  
 fill = values  
 )) +   
 geom\_tile() + theme\_classic() +  
 theme(  
 axis.line = element\_blank(),  
 legend.position = "top",  
 legend.justification = "left",  
 plot.title = element\_text(hjust = 0.5),  
 legend.title = element\_blank(),  
 legend.key.width = unit(4.5, "cm")  
 ) +  
 scale\_x\_discrete(expand = c(0, 0)) +  
 scale\_fill\_gradientn(colours = hm.palette(100)) +  
 labs(title = "Cluster Characteristics",  
 x = "Clusters",  
 y = "Features",  
 fill = "Centroids")



##Based on the preceding cluster analysis and data, we can deduce that cluster1 is beneficial to children. As a result, this can be recommended for use in elementary public schools’ daily lunches.

##We also need to standardize the data such that each variable has the same scale. If the variables’ scales aren’t the same, the model may be skewed toward the variables with larger magnitudes.