MachlearnAssignment5

Abhishek_Sau

26/04/2021

Hierarchical Clustering

The dataset "Cereals.csv" includes nutritional information, store display, and consumer ratings for 77 breakfast cereals. Data Preprocessing. Remove all cereals with missing values.

```
setwd("D:\\Study\\Assignments\\MachLearn\\MachLearnAssignment5_Hierarchical")
cereals_df<- read.csv("Cereals.csv")
str(cereals_df)
```

```
## 'data.frame':    77 obs. of  16 variables:
##  $ name    : chr  "100%_Bran" "100%_Natural_Bran" "All-Bran" "All-Bran_with_Extra_Fiber" ...
##  $ mfr     : chr  "N" "Q" "K" "K" ...
##  $ type    : chr  "C" "C" "C" "C" ...
##  $ calories: int  70 120 70 50 110 110 110 130 90 90 ...
##  $ protein : int  4 3 4 4 2 2 2 3 2 3 ...
##  $ fat     : int  1 5 1 0 2 2 0 2 1 0 ...
##  $ sodium  : int  130 15 260 140 200 180 125 210 200 210 ...
##  $ fiber   : num  10 2 9 14 1 1.5 1 2 4 5 ...
##  $ carbo   : num  5 8 7 8 14 10.5 11 18 15 13 ...
##  $ sugars  : int  6 8 5 0 8 10 14 8 6 5 ...
##  $ potass  : int  280 135 320 330 NA 70 30 100 125 190 ...
##  $ vitamins: int  25 0 25 25 25 25 25 25 25 25 ...
##  $ shelf   : int  3 3 3 3 3 1 2 3 1 3 ...
##  $ weight  : num  1 1 1 1 1 1 1 1.33 1 1 ...
##  $ cups    : num  0.33 1 0.33 0.5 0.75 0.75 1 0.75 0.67 0.67 ...
##  $ rating  : num  68.4 34 59.4 93.7 34.4 ...
```

```
head(cereals_df)
```

```
##                        name mfr type calories protein fat sodium fiber carbo
## 1                 100%_Bran   N    C       70       4   1    130  10.0   5.0
## 2         100%_Natural_Bran   Q    C      120       3   5     15   2.0   8.0
## 3                  All-Bran   K    C       70       4   1    260   9.0   7.0
## 4 All-Bran_with_Extra_Fiber   K    C       50       4   0    140  14.0   8.0
## 5            Almond_Delight   R    C      110       2   2    200   1.0  14.0
## 6   Apple_Cinnamon_Cheerios   G    C      110       2   2    180   1.5  10.5
##   sugars potass vitamins shelf weight cups   rating
## 1      6    280       25     3      1 0.33 68.40297
## 2      8    135        0     3      1 1.00 33.98368
## 3      5    320       25     3      1 0.33 59.42551
## 4      0    330       25     3      1 0.50 93.70491
## 5      8     NA       25     3      1 0.75 34.38484
## 6     10     70       25     1      1 0.75 29.50954
```

```r
library(DataExplorer)
```

```
## Warning: package 'DataExplorer' was built under R version 4.0.4
```

```r
introduce(cereals_df) #No. of missing values
```

```
##   rows columns discrete_columns continuous_columns all_missing_columns
## 1   77      16                3                 13                   0
##   total_missing_values complete_rows total_observations memory_usage
## 1                    4            74               1232        17296
```

```r
cereals_df1<-na.omit(cereals_df) #dataset with omitted rows with missing values
```

Apply hierarchical clustering to the data using Euclidean distance to the normalized measurements.

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.5     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.0.4

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(dendextend)
```

```
## Warning: package 'dendextend' was built under R version 4.0.4

##
## ---------------------
## Welcome to dendextend version 1.14.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## Or contact: <tal.galili@gmail.com>
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------

##
## Attaching package: 'dendextend'

## The following object is masked from 'package:stats':
##
##     cutree
```

```r
library(cluster)
library(fastDummies)
```

```r
#Identifying categorical and numeric variables


cereals_df1$name<-as.factor(cereals_df1$name)
cereals_df1$mfr<-as.factor(cereals_df1$mfr)
cereals_df1$type<-as.factor(cereals_df1$type)
cereals_df1$shelf <- as.factor(cereals_df1$shelf)

#creating dummy variables
vaar <- colnames(cereals_df1)
num_var <- c("calories", "protein", "fat", "sodium","fiber","carbo","sugars","potass","vitamins","weigh

cat_var<-cereals_df1[which(colnames(cereals_df1) %in% c('name','mfr','type','shelf'))]
cat_var<-data.frame(apply((cereals_df1[which(colnames(cereals_df1) %in% c('name','mfr','type','shelf'))


dummy_vars <- fastDummies::dummy_columns(cat_var %>% select(-name))
num_vars <- cereals_df1[,c(4:12, 14:16)]
cereals_df2 <-cbind(cereals_df1$name,dummy_vars,num_vars) %>% select(-c(mfr, type, shelf))
```

Normalizing the data set

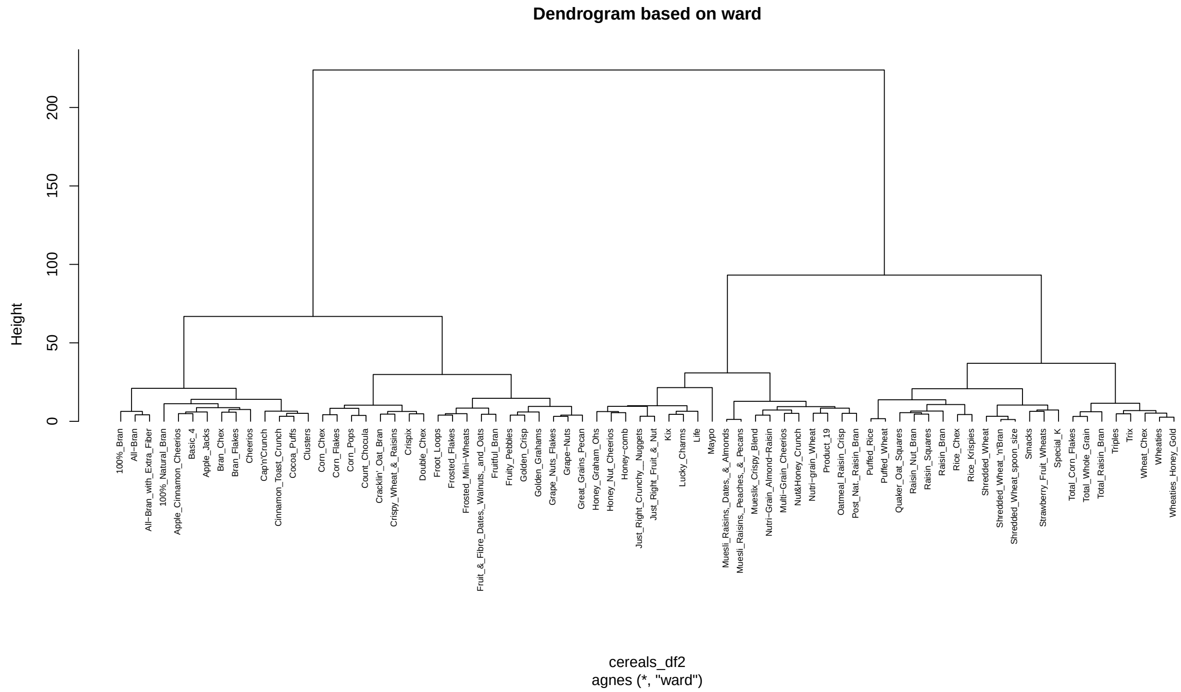
```r
cereals_df2[,c(2:25)]<- scale(cereals_df2[,c(2:25)], scale = TRUE, center = TRUE)
```

Q1. Use Agnes to compare the clustering from single linkage, complete linkage, average linkage, and Ward. Choose the best method.

```r
hc1 <- agnes(cereals_df2, method = "complete")
hc2 <- agnes(cereals_df2, method = "average")
hc3 <- agnes(cereals_df2, method = "single")
hc4 <- agnes(cereals_df2, method = "ward")

ac <- c(hc1$ac,hc2$ac,hc3$ac, hc4$ac)
ac_method <- c(hc1$method,hc2$method,hc3$method, hc4$method)
ac_df <- data.frame(ac_method, ac)
ac_df
```

```
##   ac_method        ac
## 1  complete 0.9357221
## 2   average 0.8777588
## 3    single 0.7192344
## 4      ward 0.9787001
```

```r
pltree(hc4, cex = 0.6, hang = -1, main = "Dendrogram based on ward", labels = cereals_df2$`cereals_df1$
```

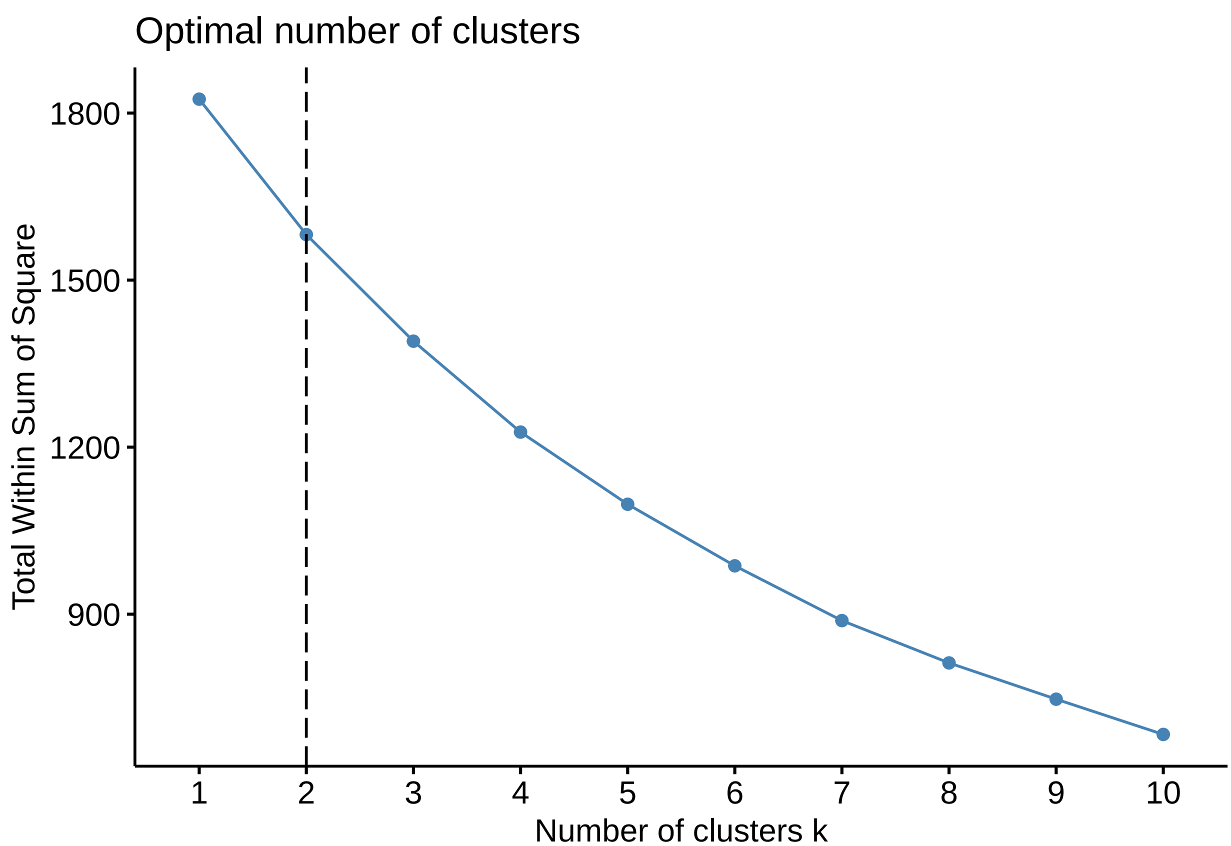


As per the above table, ward method has the highest agglomerative coefficient, i.e closest to one. Hence, it gives the best clusters.

Q2. How many clusters would you choose?

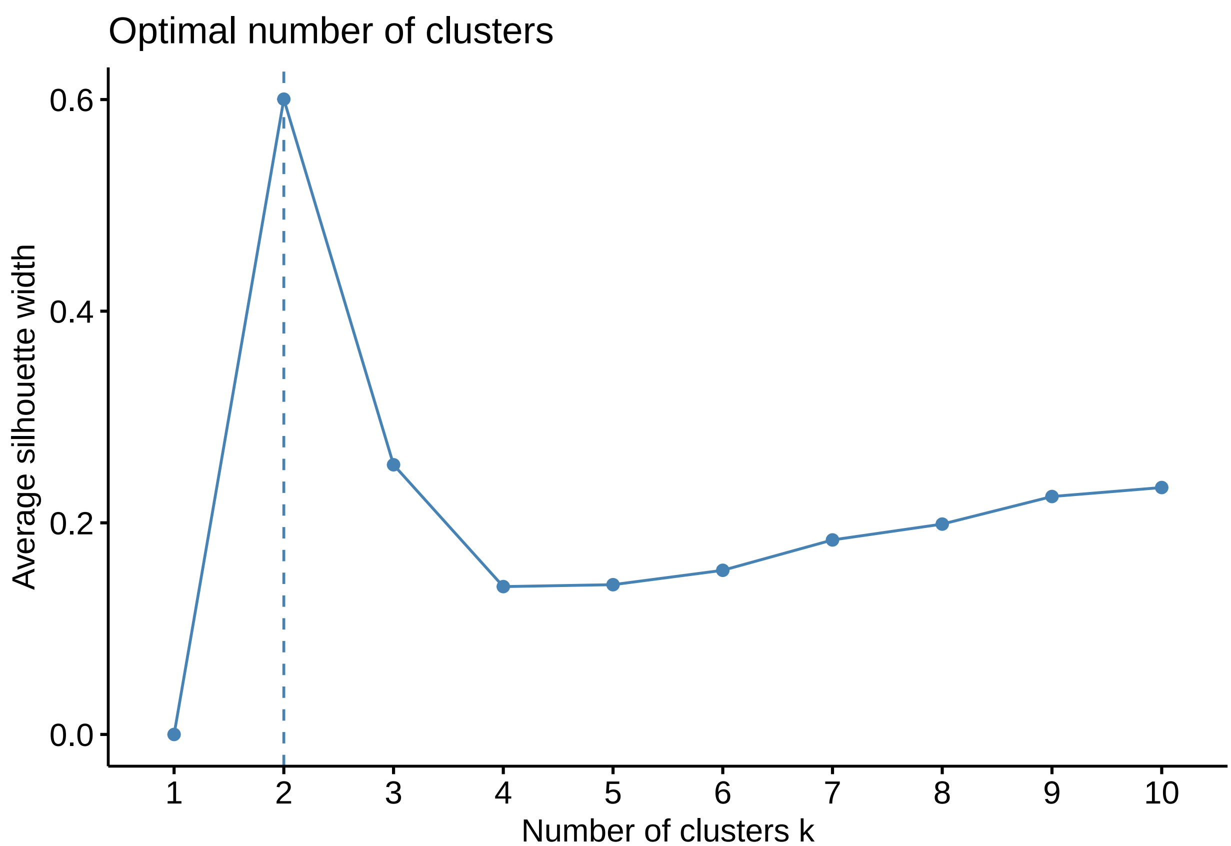
```
fviz_nbclust(cereals_df2, hcut, method = "wss")+
 geom_vline(xintercept = 2, linetype = 5)
```

```
## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion
```




```
fviz_nbclust(cereals_df2, hcut, method = "silhouette")
```

```
## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x): NAs introduced by coercion

## Warning in stats::dist(x): NAs introduced by coercion
```




```
cereals_df2<- cereals_df2 %>% mutate(cluster= cutree(hc4, k=2))
```

As per the dendogram, I will choose 2 clusters.

Q3. Comment on the structure of the clusters and on their stability.

```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
set.seed(12)

split_index <- createDataPartition(cereals_df2$rating, p=0.6, times = 1, list = FALSE)

cereal_part1 <- cereals_df2[split_index, ]
cereal_part2 <- cereals_df2[-split_index, ]

centroid1<- cereal_part1 %>% select_if(is.numeric) %>% filter(cluster==1) %>% colMeans()
centroid2 <- cereal_part1 %>% select_if(is.numeric) %>% filter(cluster==2) %>% colMeans()

centroid <- rbind(centroid1, centroid2)
```

```
cluster_B <- data.frame(data=seq(1,nrow(cereal_part2),1),clusterB = rep(0,nrow(cereal_part2)))

for(x in 1:nrow(cereal_part2))
{ cluster_B$clusterB <- which.min(as.matrix(get_dist(as.data.frame(rbind(centroid[,-25],cereal_part2[x,

cluster_B <- cluster_B %>% mutate(orig_clusters = cereal_part2$cluster)

mean(cluster_B$clusterB==cluster_B$orig_clusters)
```

```
## [1] 0.3928571
```

Answer: The clusters are not stable as per the comparision.

Q4. The elementary public schools would like to choose a set of cereals to include in their daily cafeterias. Every day a different cereal is offered, but all cereals should support a healthy diet. For this goal, you are requested to find a cluster of "healthy cereals." Should the data be normalized? If not, how should they be used in the cluster analysis?

```
healthy_cereals <- data.frame(cereals_df2 %>% filter(cluster==2) %>% select_if(is.numeric) %>% colMeans
```

Cluster 2 has cereals those are rich in protein, Vitamin, Carbs as well as low in sugar and sodium. Hence, Cereals in cluster 2 can be included to support a healthy diet