Machine Learning - Assignment 5

Tom Scarberry

11/11/2020

I have created comments where code was created and utilized, but I have excluded the output from the knit file.

Load cereals data and check (code and result excluded from output)

Assign cereal names as row names (code and result excluded from output)

Normalize the data with all data in original file and confirm normalization of the data

```
library(caret)
```

```
## Loading required package: lattice
```

```
## 
## Attaching package: 'caret'
```

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

```
cereals.z.norm<-preProcess(cereals[ ,3:11 & 13:15], method = c("center","scale"))
cereals.norm<-cereals
cereals.norm<-predict(cereals.z.norm,cereals.norm[ ,3:11 & 13:15])
summary(cereals.norm)
```

```
##      mfr                type       calories          protein
##  Length:77          C:74    Min.   :-2.9195   Min.   :-1.4116
##  Class :character   H: 3    1st Qu.:-0.3533   1st Qu.:-0.4982
##  Mode  :character           Median : 0.1600   Median : 0.4152
##                             Mean   : 0.0000   Mean   : 0.0000
##                             3rd Qu.: 0.1600   3rd Qu.: 0.4152
##                             Max.   : 2.7262   Max.   : 3.1554
## 
##       fat               sodium             fiber               carbo
##  Min.   :-1.0065   Min.   :-1.9047   Min.   :-0.90290   Min.   :-2.50878
##  1st Qu.:-1.0065   1st Qu.:-0.3540   1st Qu.:-0.48333   1st Qu.:-0.71728
##  Median :-0.0129   Median : 0.2424   Median :-0.06375   Median :-0.07745
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.9807   3rd Qu.: 0.6003   3rd Qu.: 0.35582   3rd Qu.: 0.56237
##  Max.   : 3.9614   Max.   : 1.9124   Max.   : 4.97115   Max.   : 2.09795
##                                                         NA's   :1
##      sugars             potass           vitamins       shelf
##  Min.   :-1.60467   Min.   :-1.1883   Min.   :-1.2643   1:20
##  1st Qu.:-0.91953   1st Qu.:-0.7977   1st Qu.:-0.1453   2:21
##  Median :-0.00601   Median :-0.1231   Median :-0.1453   3:36
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.90751   3rd Qu.: 0.3030   3rd Qu.:-0.1453
##  Max.   : 1.82103   Max.   : 3.2855   Max.   : 3.2115
##  NA's   :1          NA's   :2
##      weight             cups            rating
##  Min.   :-3.5195   Min.   :-2.4538   Min.   :-1.7529
##  1st Qu.:-0.1968   1st Qu.:-0.6490   1st Qu.:-0.6757
##  Median :-0.1968   Median :-0.3053   Median :-0.1613
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:-0.1968   3rd Qu.: 0.7690   3rd Qu.: 0.5811
##  Max.   : 3.1260   Max.   : 2.9176   Max.   : 3.6334
##
```

Remove missing values

```r
library(tidyverse)
cereals.norm.no.na<-cereals.norm%>%drop_na()
```

Apply hierarchical clustering using euclidean distance. Ward method is the best option of agnes clustering approach as it has the highest value for the agglomerative coefficient of the different approaches evaluated (single, complete, average, or ward). Create dendrogram visual of the ward approach.

Create a new variable that identifies the assigned cluster for each cereal for comparison later in the stability analysis.

```r
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 4.0.3
```

```r
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.0.3
```

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

```
cereals.single<-agnes(cereals.norm.no.na[ ,3:11 & 13:15], metric = "euclidean", method = "single")
cereals.complete<-agnes(cereals.norm.no.na[ ,3:11 & 13:15], metric = "euclidean", method = "complete")
cereals.average<-agnes(cereals.norm.no.na[ ,3:11 & 13:15], metric = "euclidean", method = "average")
cereals.ward<-agnes(cereals.norm.no.na[ ,3:11 & 13:15], metric = "euclidean", method = "ward")

cereals.single$ac
```

```
## [1] 0.5872037
```

```
cereals.complete$ac
```

```
## [1] 0.8323943
```

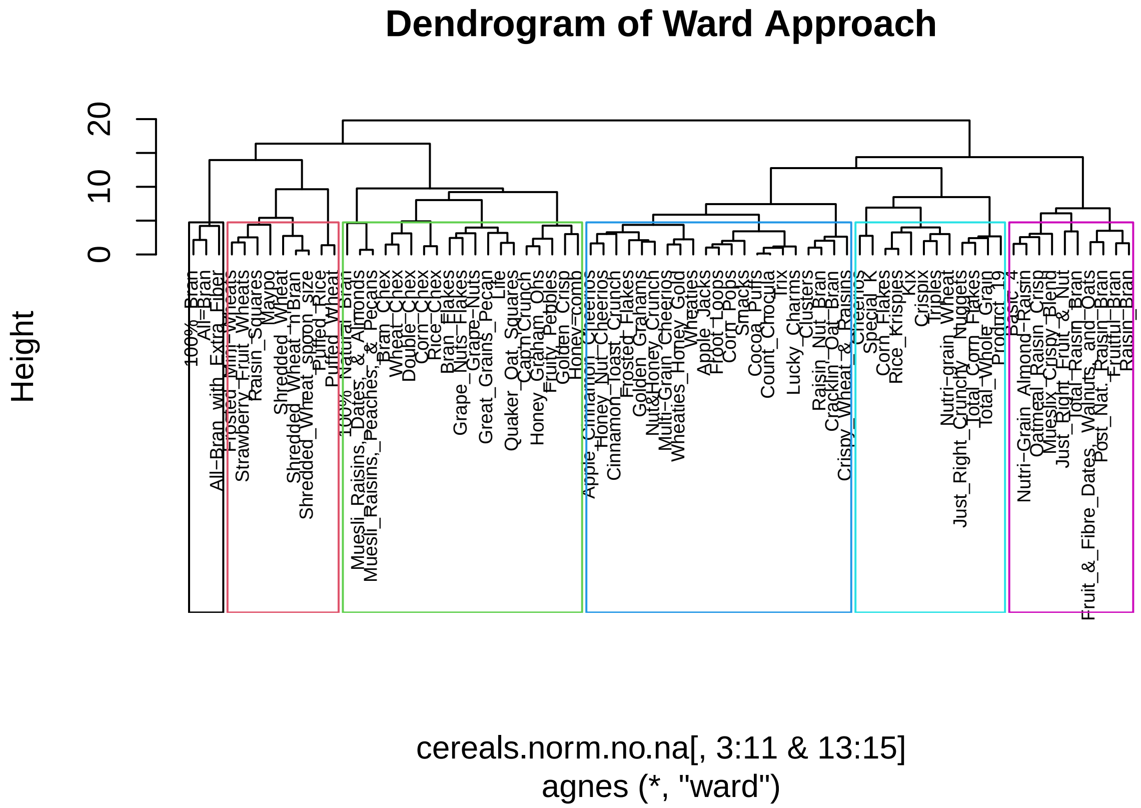
```
cereals.average$ac
```

```
## [1] 0.7581175
```

```
cereals.ward$ac
```

```
## [1] 0.9002742
```

```
pltree(cereals.ward, cex =0.6, hang =-1, main = "Dendrogram of Ward Approach")
rect.hclust(cereals.ward,k=6, border = 1:6)
```



```
cereal.cluster<-cutree(cereals.ward, k=6)
cereals.ward.cluster<-cbind(cereals.norm.no.na,cereal.cluster)
cereals.ward.cluster$cereal.cluster<-as.factor(cereals.ward.cluster$cereal.cluster)
```

Hierarchical Clustering starts grouping the two closest points together by distance into clusters until all points eventually make one cluster. It uses the data itself to create the clusters and as more clusters are made, the model creates a hierarchical visual that helps the modeler chose the number of clusters based on distance of the clusters (and thus corresponding data points) from one another. The model will always achieve the same result as long as the same method of modeling is selected, unlike K-means.

K-means identifies cluster centroid points and then begins to identify points close to the centroids to create clusters. For K-means the user must identify to the number of clusters and the model seeks to minimize the sum of distances between the data and chosen centroid. The model works using a stochastic process, so varying cluster centroid locations can create variance in the model when run at different times.

Using the Dendrogram, I would select six clusters using a distance of approximately 12 from the scale on the ward dendrogram graphic.

##————————————————————————————

Partition the data and check stability by adding the new cluster identification data to the partitioned data set and comparing the clusters to identify whether clusters stay the same or change with the smaller data sample. Create new dendrogram of ward approach of the data subset.

```
library(groupdata2)
```

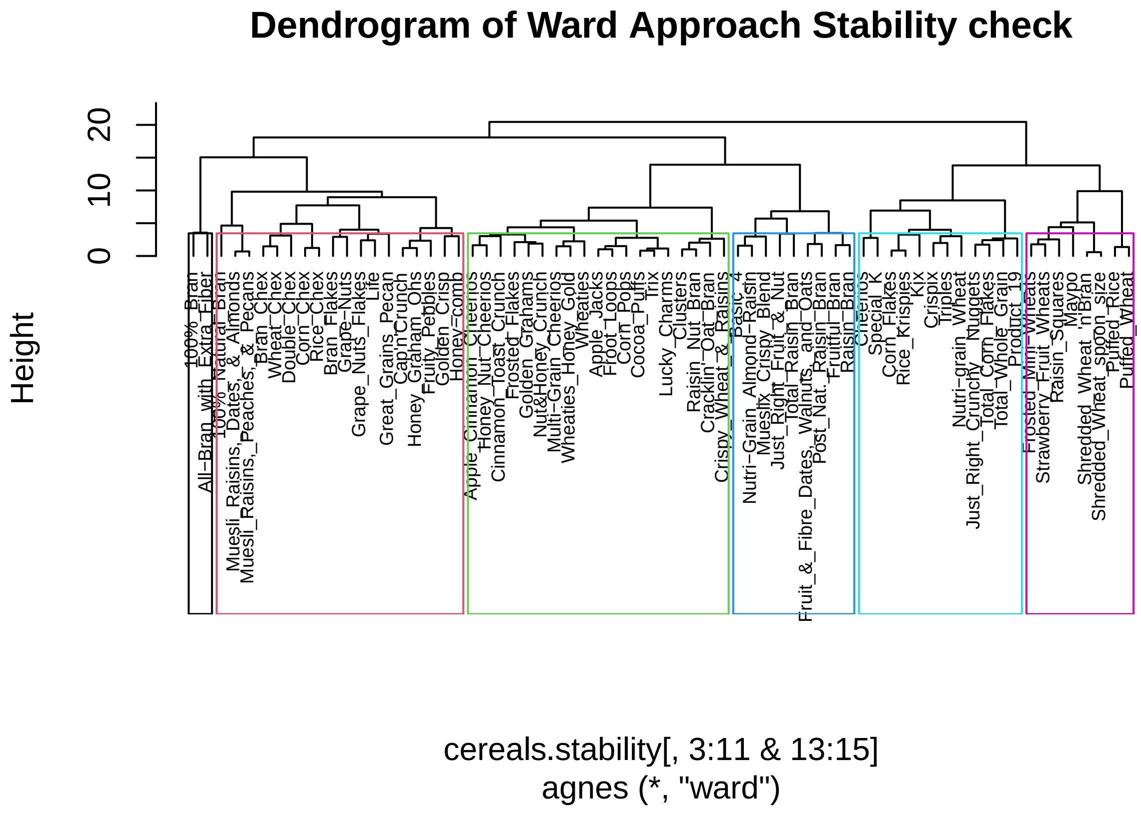
```
## Warning: package 'groupdata2' was built under R version 4.0.3
```

```
set.seed(123)
Train.index=createDataPartition(cereals.ward.cluster$shelf, p=0.9, list=FALSE)
cereals.stability=cereals.ward.cluster[Train.index,]


cereals.ward.stability<-agnes(cereals.stability[ ,3:11 & 13:15], metric = "euclidean", method = "ward")
cereals.ward.stability$ac
```

```
## [1] 0.901469
```

```
pltree(cereals.ward.stability, cex =0.6, hang =-1, main = "Dendrogram of Ward Approach Stability check"
rect.hclust(cereals.ward.stability,k=6, border = 1:6)
```



```
cereal.cluster.s<-cutree(cereals.ward.stability, k=6)
cereals.ward.stability<-cbind(cereals.stability,cereal.cluster.s)
cereals.ward.stability$cereal.cluster.s<-(cereals.ward.stability$cereal.cluster.s)

cereals.ward.stability%>%group_by(cereal.cluster)%>%
  summarise(Cluster.match=n_distinct(cereal.cluster.s),
            Cluster.mean=mean(cereal.cluster.s),
            cluster.sd=sd(cereal.cluster.s))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 6 x 4
##   cereal.cluster Cluster.match Cluster.mean cluster.sd
##   <fct>                  <int>        <dbl>      <dbl>
## 1 1                          1            1          0
## 2 2                          1            2          0
## 3 3                          1            3          0
## 4 4                          1            4          0
## 5 5                          1            5          0
## 6 6                          1            6          0
```

```
cereals.ward.stability$cereal.cluster<-as.integer(cereals.ward.stability$cereal.cluster)

check<-ifelse(cereals.ward.stability$cereal.cluster==cereals.ward.stability$cereal.cluster.s, "True", "
check.df<-as.data.frame(check)
check.df$check<-as.factor(check.df$check)

summary(check.df)
```

```
##   check
##  True:68
```

The model's structure is stable as all cereals stay in the initial clusters after pulling a sample of 90% of the original cereals and re-doing the clusters with the ward approach.

##————————————————————————————

Create a summary table of the key information for identification of the healthy cereal choices for the elementary school using the six clusters identified in the ward hierarchical model.

```
Healthly.cereals.a<-cereals.ward.cluster%>%group_by(cereal.cluster)%>%
  summarise(Fat=mean(fat),
            Sodium=mean(sodium),
            Sugar=mean(sugars),
            Fiber=mean(fiber),
            Potassium=mean(potass),
            Vitamins=mean(vitamins))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
Healthly.cereals.a
```

```
## # A tibble: 6 x 7
##   cereal.cluster    Fat  Sodium   Sugar   Fiber Potassium Vitamins
##   <fct>           <dbl>   <dbl>   <dbl>   <dbl>     <dbl>    <dbl>
## 1 1              -0.344  0.203  -0.767   3.71       3.00    -0.145
## 2 2               0.301  0.0290  0.0661 -0.130     -0.131   -0.204
## 3 3               0.176  0.129   0.690  -0.413     -0.441   -0.145
## 4 4               0.385  0.386   0.885   0.629      1.05     0.526
## 5 5              -0.427  0.968  -0.939  -0.448     -0.573    0.974
## 6 6              -0.896 -1.88   -1.07   -0.0171    -0.115   -0.767
```

First select the criteria desired to determine the healthiest cereal cluster. For this evaluation I will evaluate cereals where lower: fat, sodium, and sugar are healthier options and where higher: fiber, potassium, and vitamins are healthier options. The result of this evaluation leads to cluster 1 being the healthiest choices. While 1 is a bit high in sodium and below average for vitamins, it offers better than average values for the remainder of the selected criteria with especially high values for fiber and potassium. This will allow kids to start their school day off well.

The data should be normalized for the cluster analysis, otherwise it would be subject to the scale for each variable and a variable with much higher values can skew the model results.

In order to compare the two cluster model results, a modeler could run a Hierarchical clustering model and select the number of clusters and then use the number of clusters (starts for the k-means algorithm) as in input into k-means clusters to evaluate the similarities/differences of the resulting clusters.

Hierarchical clustering is more a consistent modeling approach and allows a visual (dendrogram) that easily identifies various clustering options based on distance within or between clusters as you evaluate higher and higher level clusters within the hierarchy. K-means models are variable between executions with the initial selection of the centroid locations.