Lecture 4 and Assignment

Arielle Coq AC4140

2/11/2020

### Part 1:Implenting a Simple Prediction Pipeline

First loading in the tidyverse and viridis library into my markdown page

First will enter the data set into R and will do some basic data cleaning.

phy\_activity<-read.csv("class4\_p1.csv", header=TRUE)  
  
phy\_activity =   
 phy\_activity %>%   
 janitor::clean\_names() %>%   
 rename (PACID = "x")

Cleaning some of the data and making the features categorical instead of numerical based on the codebook provided.

phy\_activity =  
 phy\_activity %>%   
 mutate(chronic1 = recode(chronic1,"2" = "0", "1"= "1"),  
 chronic1 = factor(chronic1, levels= c("0", "1")),  
 chronic3 = recode(chronic3, "2" = "0", "1"= "1"),  
 chronic3 = factor(chronic3, levels= c("0", "1")),  
 chronic4 = recode(chronic4, "2" = "0", "1"= "1"),  
 chronic4 = factor(chronic4, levels= c("0", "1")),  
 tobacco1 = factor(tobacco1, levels= c("1", "2","3")),  
 alcohol1 = factor(alcohol1, levels= c("1", "2","3")),  
 habits5 = factor(habits5, levels= c("1", "2","3","4")),  
 habits7 = factor(habits7, levels= c("1", "2","3","4","5")),  
 agegroup = factor(agegroup, levels= c("1", "2","3","4")),  
 dem3 = recode(dem3,"2" = "0", "1"= "1"),  
 dem3 = factor(dem3, levels= c("0", "1")),  
 dem4 = recode(dem4,"2" = "0", "1"= "1"),  
 dem4 = factor(dem4, levels= c("0", "1")),  
 dem8 = recode(dem8,"2" = "0", "1"= "1"),  
 dem8 = factor(dem8, levels= c("0", "1")),  
 povertygroup = recode(povertygroup,"1" = "1", "2"= "2","3"= "3", "4"= "4","5"= "5"),  
 povertygroup = factor(povertygroup, levels= c("1", "2","3","4","5"))) %>%   
 drop\_na()

Created a training and testing set where the training data set has 70% of the data and the testing set has 30% of the data set.

set.seed(100)  
train.indices<-createDataPartition(y=phy\_activity$healthydays,p=0.7,list=FALSE)  
  
training<-phy\_activity[train.indices,]  
testing<-phy\_activity[-train.indices,]

### Question 1- Fit two prediction models using different subsets of the features in the training data. Features can overlap in the two models, but the feature sets should not be exactly the same across models.

model.1 <- lm(healthydays ~ chronic1 + chronic3 + chronic4 + bmi + agegroup, data=training)  
summary(model.1)

##   
## Call:  
## lm(formula = healthydays ~ chronic1 + chronic3 + chronic4 + bmi +   
## agegroup, data = training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -29.2538 0.5624 1.8119 3.6242 10.5620   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 31.16253 1.19865 25.998 < 2e-16 \*\*\*  
## chronic11 -1.86106 0.50301 -3.700 0.000224 \*\*\*  
## chronic31 -1.96257 0.65381 -3.002 0.002732 \*\*   
## chronic41 -4.39291 0.85535 -5.136 3.2e-07 \*\*\*  
## bmi -0.08417 0.03489 -2.413 0.015968 \*   
## agegroup2 -0.31785 0.91383 -0.348 0.728029   
## agegroup3 -1.68922 0.91602 -1.844 0.065380 .   
## agegroup4 -2.74026 0.97691 -2.805 0.005101 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.508 on 1401 degrees of freedom  
## Multiple R-squared: 0.08378, Adjusted R-squared: 0.07921   
## F-statistic: 18.3 on 7 and 1401 DF, p-value: < 2.2e-16

model.2 <- lm(healthydays ~ chronic1 + chronic3 + chronic4 + alcohol1 + dem3 + dem4 + dem8 + tobacco1 + gpaq8totmin + gpaq11days + habits7 + habits5,  
 data=training)  
summary(model.2)

##   
## Call:  
## lm(formula = healthydays ~ chronic1 + chronic3 + chronic4 + alcohol1 +   
## dem3 + dem4 + dem8 + tobacco1 + gpaq8totmin + gpaq11days +   
## habits7 + habits5, data = training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -29.590 -0.109 1.474 3.674 15.742   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 30.7567119 1.4615040 21.045 < 2e-16 \*\*\*  
## chronic11 -2.4302211 0.4517957 -5.379 8.78e-08 \*\*\*  
## chronic31 -1.3670786 0.6403974 -2.135 0.032957 \*   
## chronic41 -3.1285555 0.8482823 -3.688 0.000235 \*\*\*  
## alcohol12 -1.9002055 1.1907907 -1.596 0.110772   
## alcohol13 -2.5427159 1.1698353 -2.174 0.029906 \*   
## dem31 -0.6291045 0.4079454 -1.542 0.123270   
## dem41 -1.1439213 0.4951997 -2.310 0.021033 \*   
## dem81 0.3436284 0.4206339 0.817 0.414109   
## tobacco12 -0.6483595 1.0589172 -0.612 0.540449   
## tobacco13 0.3419828 0.6541248 0.523 0.601190   
## gpaq8totmin -0.0002094 0.0020907 -0.100 0.920249   
## gpaq11days 0.1047235 0.0746348 1.403 0.160797   
## habits72 -0.6948632 0.6516638 -1.066 0.286477   
## habits73 -0.4341581 0.6367470 -0.682 0.495455   
## habits74 -2.8622208 0.7341021 -3.899 0.000101 \*\*\*  
## habits75 -4.8229505 1.0879513 -4.433 1.00e-05 \*\*\*  
## habits52 0.5286456 0.4674866 1.131 0.258324   
## habits53 -1.9749070 0.6209524 -3.180 0.001503 \*\*   
## habits54 -4.2325524 1.0366045 -4.083 4.70e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.277 on 1389 degrees of freedom  
## Multiple R-squared: 0.1467, Adjusted R-squared: 0.135   
## F-statistic: 12.57 on 19 and 1389 DF, p-value: < 2.2e-16

### Question 2- Apply both models within the test data and determine which model is the preferred prediction model using the appropriate evaluation metric(s)

rmse(model.1, testing)

## [1] 7.246041

rmse(model.2, testing)

## [1] 7.123927

### Question 3- Describe one setting (in 1-2 sentences) where the implementation of your final model would be useful.

The implementation of my final model would be most useful in model 1 when looking at the different chronic diseases with agegroup and bmi in the model. This model could help predict chronic disease diagnosis when looking at bmi and across different age groups.

## Part 2:Conducting an Unsupervised Analysis

### Question 4

Loaded the US Arrests Data set into R.

data(USArrests)

Then I checked to see if any scaling had to be done to any of the variables.

USArrests.data.nomiss<-na.omit(USArrests)  
  
#Check means and SDs to determine if scaling is necessary  
colMeans(USArrests.data.nomiss, na.rm=TRUE)

## Murder Assault UrbanPop Rape   
## 7.788 170.760 65.540 21.232

apply(USArrests.data.nomiss, 2, sd, na.rm=TRUE)

## Murder Assault UrbanPop Rape   
## 4.355510 83.337661 14.474763 9.366385

USArrests.scale<-scale(USArrests.data.nomiss, center=TRUE, scale=TRUE)  
apply(USArrests.scale, 2, sd, na.rm=TRUE)

## Murder Assault UrbanPop Rape   
## 1 1 1 1

#Can compare sds used to scale with the sds above to ensure they are close.  
USArrests.data.nomiss$scale

## NULL

#view results of pca. Note the first three components are needed to explain at least 75% of the variance  
summary(USArrests.scale)

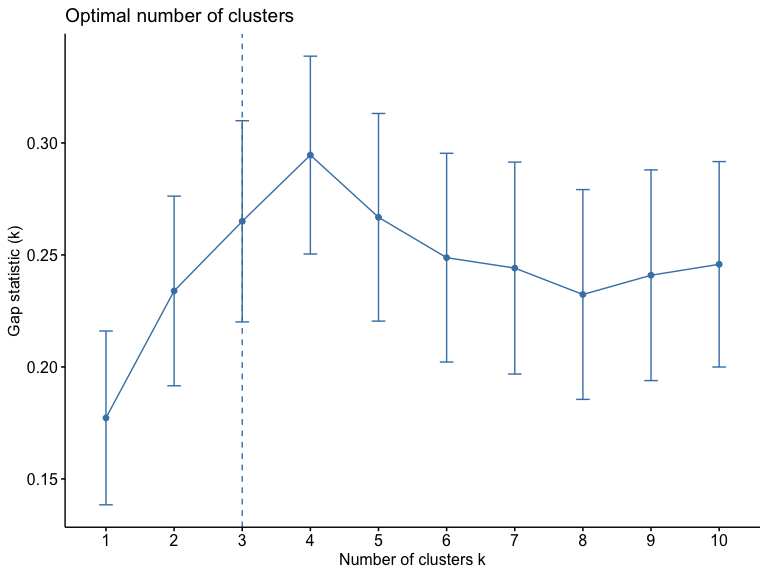
## Murder Assault UrbanPop Rape   
## Min. :-1.6044 Min. :-1.5090 Min. :-2.31714 Min. :-1.4874   
## 1st Qu.:-0.8525 1st Qu.:-0.7411 1st Qu.:-0.76271 1st Qu.:-0.6574   
## Median :-0.1235 Median :-0.1411 Median : 0.03178 Median :-0.1209   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.00000 Mean : 0.0000   
## 3rd Qu.: 0.7949 3rd Qu.: 0.9388 3rd Qu.: 0.84354 3rd Qu.: 0.5277   
## Max. : 2.2069 Max. : 1.9948 Max. : 1.75892 Max. : 2.6444

#bc.pca$rotation

Scaling is necessary for this data set because when looking at the means and sd, they are very different, therefore I scaled the data to make them all on the same scale.

Set the distance using the Euclidian method, since our features are continuous and numercial. Also sets a dismmilarty matrix to get the distance and looked at the different hierarchical clustering methods: Single, Complete and Avearge. I also performed a gap statistic to tell me the optimal number of clusstr to should have.

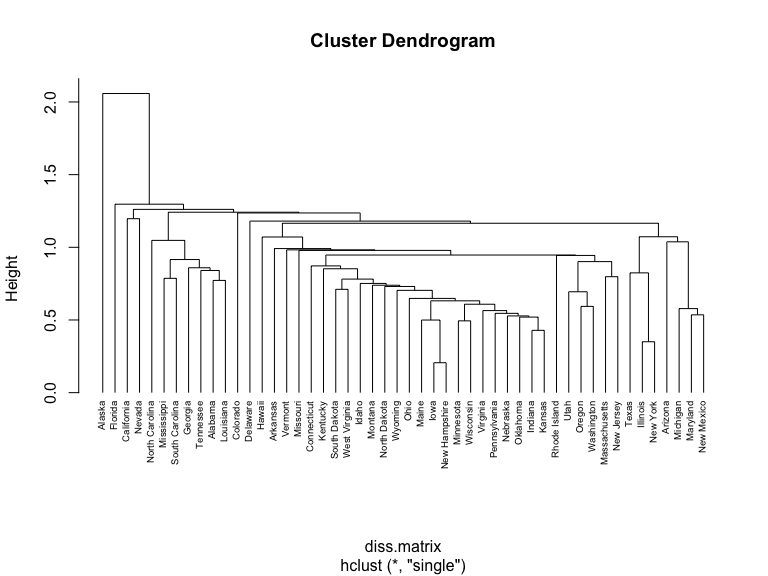
set.seed(100)  
  
gap\_stat <- clusGap(USArrests.scale, FUN = hcut, nstart = 25, K.max = 10, B = 50, mc\_metric = "euclidean")  
fviz\_gap\_stat(gap\_stat)



# Create Dissimilarity matrix  
diss.matrix <- dist(USArrests.scale, method = "euclidean")  
  
# Hierarchical clustering using Single Linkage  
hc1 <- hclust(diss.matrix, method = "single" )  
  
groups.1<-cutree(hc1,4)  
aggregate(USArrests.scale,list(groups.1),mean)

## Group.1 Murder Assault UrbanPop Rape  
## 1 1 -0.07710374 -0.1155488 -0.05683055 -0.181203  
## 2 2 0.50786248 1.1068225 -1.21176419 2.484203  
## 3 3 0.64561903 1.1188219 1.41349461 2.356085  
## 4 4 1.74767144 1.9707777 0.99898006 1.138967

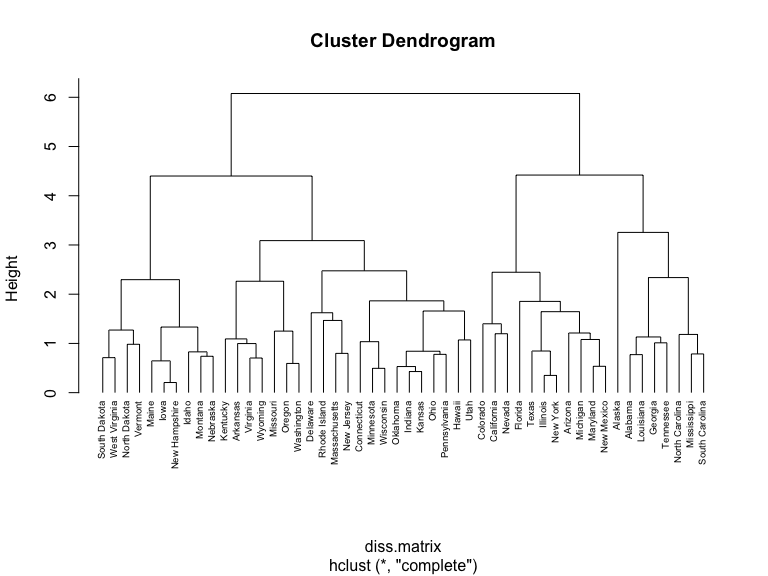
# Plot the obtained dendrogram using Single  
plot(hc1, cex = 0.6, hang = -1)



# Hierarchical clustering using Complete Linkage  
hc2 <- hclust(diss.matrix, method = "complete" )  
  
groups.2 <-cutree(hc2,4)  
aggregate(USArrests.scale,list(groups.2),mean)

## Group.1 Murder Assault UrbanPop Rape  
## 1 1 1.4463290 0.9838289 -0.8317925 0.3529110  
## 2 2 0.7499801 1.1199128 0.9361748 1.2156432  
## 3 3 -0.4400338 -0.4353831 0.3607592 -0.2830385  
## 4 4 -1.0579703 -1.1046626 -1.1219527 -1.0251554

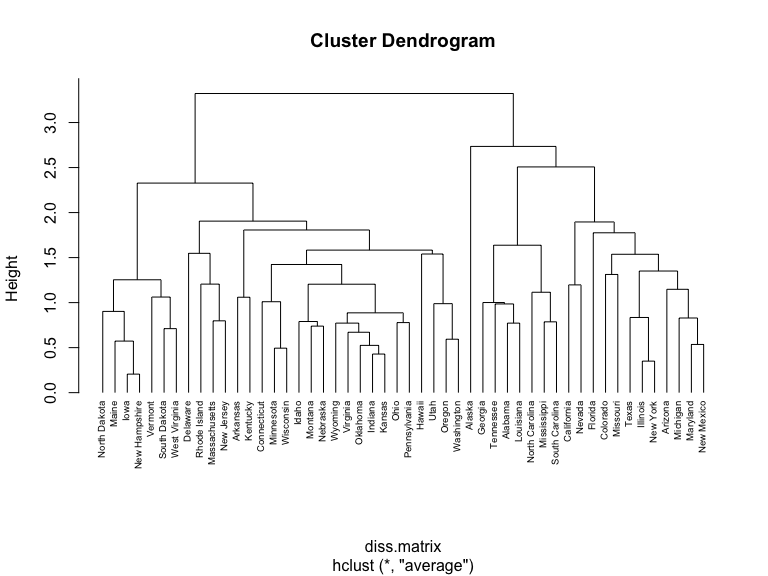
# Plot the obtained dendrogram using Complete  
plot(hc2, cex = 0.6, hang = -1)



# Hierarchical clustering using Average Linkage  
hc3 <- hclust(diss.matrix, method = "average" )  
  
groups.3<-cutree(hc3,4)  
aggregate(USArrests.scale,list(groups.3),mean)

## Group.1 Murder Assault UrbanPop Rape  
## 1 1 1.5803956 0.9662584 -0.7775109 0.04844071  
## 2 2 0.5078625 1.1068225 -1.2117642 2.48420294  
## 3 3 0.7106707 1.0338263 0.8838371 1.17633437  
## 4 4 -0.6699560 -0.6758849 -0.1317235 -0.56464334

# Plot the obtained dendrogram using Average  
plot(hc3, cex = 0.6, hang = -1)



The optimal number of clusters to use using a clear, data driven strategy is three cluster for each of the different types of compositions.

The composition for using the single linkage method shows that the third group has the highest mean for Murder, UrbanPop, group 1 has the lowest for these categories 7.648 and 66.0425 respectively. The highest mean for Assault is in group 4 and the highest mean for Rape is in group 2.

The composition for using the complete and avearage linkage methods show the same values as one other. The dendrograms are quite similar however the average dendrogram has more clusters than the complete dendrogram. They both show the highest mean for Assualt and Rape in group 1, highest mean for murder is in group 4 and the highest mean for UrbanPop is in group 2.

The dendrograms are different from each other when using the different methods, however as mentioned earlier both the complete and average methods produced the same results when looking for clusters across the 4 features. The single method produce different results from the previos two models. The complete and average models appear to be clearer and more distinct compared to the the single method model.

### Question 5

One question that can be addressed in 2020, would be to see if the change in UrbanPop can predict if there is more crime: Murder, Assualt, and Rape across the different states.