<https://topepo.github.io/caret/measuring-performance.html>

## Part I: Implementing a Simple Prediction Pipeline

#Question 1

```{r}

#Q1: 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. Clearly state which features were used in the two models.

#NTS: Now I'm tuning my hyperparameter using cross validation (through the train control function.) and doing a 3-fold cross-validation.

control.settings <- trainControl(method = "cv", number = 3)

#Now I'm running my train function

set.seed(1000)

model1\_again <- train(healthydays ~ gpaq8totmin + gpaq11days + bmi, data = class4, method = "glm", family = "gaussian", trControl = control.settings)

model1\_again

model2\_again <- train(healthydays ~ gpaq8totmin + gpaq11days + bmi + povertygroup + agegroup, data = class4, method = "glm", family = "gaussian", trControl = control.settings)

model2\_again

```

My first model includes 3 variables: BMI, gpaq8totmin(minutes of total physical activity on home chores on an average day) and gpaq11days (during the last 7 days, on how many days did you walk to get to and from places?). The root mean square error was 7.654.

My second model includes these three variables plus poverty group (is your household’s annual income from all source: 1 = <100%, 2 = 100-199%, 3 = 200-399%, 4 = 400-599%, 5 = 600%, 6 = Don’t know) and agegroup (group 1 = 18-24, group 2 = 25-44, group 3 = 45-64, and group 4 = 65+). The root mean square error was 7.444.

#Check means and SDs to determine if scaling is necessary

colMeans(USArrests, na.rm = TRUE)

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

#Question 4

```{r}

#Q4 Task: Conduct a hierarchical clustering analysis. Be sure to specify the linkage method used. Within your analysis, make sure you do both of the following:

# 1. Determine the optimal number of clusters using a clear, data-driven strategy.

# 2. Describe the composition of each cluster in terms of the original input features

data(USArrests)

#Check means and SDs to determine if scaling is necessary

colMeans(USArrests, na.rm = TRUE)

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

# NTS: First I need to create Dissimilarity matrix

diss\_matrix <- dist(USArrests, method = "euclidean")

#NTS: Then hierarchical clustering using Complete Linkage

clusters\_h <- hclust(diss\_matrix, method = "complete" )

# Plot the obtained dendrogram

plot(clusters\_h, cex = 0.4, hang = -1)

#NTS: Now I need to choose where to cut across my dendrogram to choose my number of clusters. In order to do that I have to creating a function to use within clusGap. I am now using an average linkage. After running the function, I will plot the gapstat to visualize it

hclusCut <- function(x, k) list(cluster = cutree(hclust(dist(x, method = "euclidian"), method ="average"), k = k))

gap\_stat <- clusGap(USArrests, FUN = hclusCut, K.max = 10, B = 50)

fviz\_gap\_stat(gap\_stat)

#NTS: Now, use the number of clusters from gap statistic to obtain cluster assignment for each observation

clusters\_h3 <- cutree(clusters\_h, k = 3)

table(clusters\_h3)

Q4

# FIRST WAY I DID IT BEFORE THE ALTERNATIVE WAY

#Q4 Task: Conduct a hierarchical clustering analysis. Be sure to specify the linkage method used. Within your analysis, make sure you do both of the following:

# 1. Determine the optimal number of clusters using a clear, data-driven strategy.

# 2. Describe the composition of each cluster in terms of the original input features

data(USArrests)

# NTS: First I need to create Dissimilarity matrix

diss\_matrix <- dist(USArrests, method = "euclidean")

#NTS: Then hierarchical clustering using Complete Linkage

clusters\_h <- hclust(diss\_matrix, method = "complete" )

# Plot the obtained dendrogram

plot(clusters\_h, cex = 0.4, hang = -1)

#NTS: Now I need to choose where to cut across my dendrogram to choose my number of clusters. In order to do that I have to creating a function to use within clusGap. I am now using an average linkage. After running the function, I will plot the gapstat to visualize it

hclusCut <- function(x, k) list(cluster = cutree(hclust(dist(x, method = "euclidian"), method = "complete"), k = k))

gap\_stat <- clusGap(USArrests, FUN = hclusCut, K.max = 10, B = 50)

fviz\_gap\_stat(gap\_stat)

#NTS: Now, use the number of clusters from gap statistic to obtain cluster assignment for each observation

clusters\_h3 = cutree(clusters\_h, k = 3)

table(clusters\_h3)

#Now I want to take those values and put it back onto my original dataset