M6 : Machine Learning Miniproject

Léo Besançon & Larissa Geiser

2024-05-16

1. Data loading and description

```
data <- read.csv('ObesityDataSet_raw_and_data_sinthetic.csv')
dim(data)
```

```
## [1] 2111   17
```

```
sum(is.na(data)) #no missing value
```

```
## [1] 0
```

```
head(data)
```

```
##   Gender Age Height Weight family_history_with_overweight FAVC FCVC NCP
## 1 Female  21   1.62   64.0                            yes   no    2   3
## 2 Female  21   1.52   56.0                            yes   no    3   3
## 3   Male  23   1.80   77.0                            yes   no    2   3
## 4   Male  27   1.80   87.0                             no   no    3   3
## 5   Male  22   1.78   89.8                             no   no    2   1
## 6   Male  29   1.62   53.0                             no  yes    2   3
##        CAEC SMOKE CH2O SCC FAF TUE       CALC                MTRANS
## 1 Sometimes    no    2  no   0   1         no Public_Transportation
## 2 Sometimes   yes    3 yes   3   0  Sometimes Public_Transportation
## 3 Sometimes    no    2  no   2   1 Frequently Public_Transportation
## 4 Sometimes    no    2  no   2   0 Frequently               Walking
## 5 Sometimes    no    2  no   0   0  Sometimes Public_Transportation
## 6 Sometimes    no    2  no   0   0  Sometimes            Automobile
##            NObeyesdad
## 1       Normal_Weight
## 2       Normal_Weight
## 3       Normal_Weight
## 4  Overweight_Level_I
## 5 Overweight_Level_II
## 6       Normal_Weight
```

The data comes from more than 2’000 individuals. It consists of an evaluation of 16 physical and behavioral attributes. The last column indicates the weight class of the individual.

1.1. Data harmonization

To be able to use this dataset with the different algorithms, we need to modify the variables encoded with characters into numbers.

```r
#################### DATA HARMONIZATION

# gender : 0 = female, 1 = male
data$Gender[data$Gender == 'Female'] <- 0
data$Gender[data$Gender == 'Male'] <- 1
data$Gender <- as.numeric(data$Gender)

# family history with overweight
data$family_history_with_overweight[data$family_history_with_overweight ==
'no'] <- 0
data$family_history_with_overweight[data$family_history_with_overweight ==
'yes'] <- 1
data$family_history_with_overweight <-
as.numeric(data$family_history_with_overweight)

# FAVC (frequency of highly caloric food)
data$FAVC[data$FAVC == 'no'] <- 0
data$FAVC[data$FAVC == 'yes'] <- 1
data$FAVC <- as.numeric(data$FAVC)

# CAEC (snacking between meals)
data$CAEC[data$CAEC == 'no'] <- 0
data$CAEC[data$CAEC == 'Sometimes'] <- 1
data$CAEC[data$CAEC == 'Frequently'] <- 2
data$CAEC[data$CAEC == 'Always'] <- 3
data$CAEC <- as.numeric(data$CAEC)

# smoke
data$SMOKE[data$SMOKE == 'no'] <- 0
data$SMOKE[data$SMOKE == 'yes'] <- 1
data$SMOKE <- as.numeric(data$SMOKE)

# SCC (calories monitoring)
data$SCC[data$SCC == 'no'] <- 0
data$SCC[data$SCC == 'yes'] <- 1
data$SCC <- as.numeric(data$SMOKE)

# CALC (alcohol consumption)
data$CALC[data$CALC == 'no'] <- 0
data$CALC[data$CALC == 'Sometimes'] <- 1
data$CALC[data$CALC == 'Frequently'] <- 2
data$CALC[data$CALC == 'Always'] <- 3
data$CALC <- as.numeric(data$CALC)
```

```
# MTRANS (mode of transportation)
data$MTRANS <- as.factor(data$MTRANS) # better to encode with numbers ? i
don't think so

# NObeyesdad (obesity)
data$NObeyesdad <- as.factor(data$NObeyesdad)
colnames(data)[colnames(data) == 'NObeyesdad'] <- 'Obesity'

# results
head(data)
```

```
##   Gender Age Height Weight family_history_with_overweight FAVC FCVC NCP
CAEC
## 1      0  21   1.62   64.0                              1    0    2   3
1
## 2      0  21   1.52   56.0                              1    0    3   3
1
## 3      1  23   1.80   77.0                              1    0    2   3
1
## 4      1  27   1.80   87.0                              0    0    3   3
1
## 5      1  22   1.78   89.8                              0    0    2   1
1
## 6      1  29   1.62   53.0                              0    1    2   3
1
##   SMOKE CH2O SCC FAF TUE CALC                MTRANS             Obesity
## 1     0    2   0   0   1    0 Public_Transportation       Normal_Weight
## 2     1    3   1   3   0    1 Public_Transportation       Normal_Weight
## 3     0    2   0   2   1    2 Public_Transportation       Normal_Weight
## 4     0    2   0   2   0    2               Walking  Overweight_Level_I
## 5     0    2   0   0   0    1 Public_Transportation Overweight_Level_II
## 6     0    2   0   0   0    1            Automobile       Normal_Weight
```

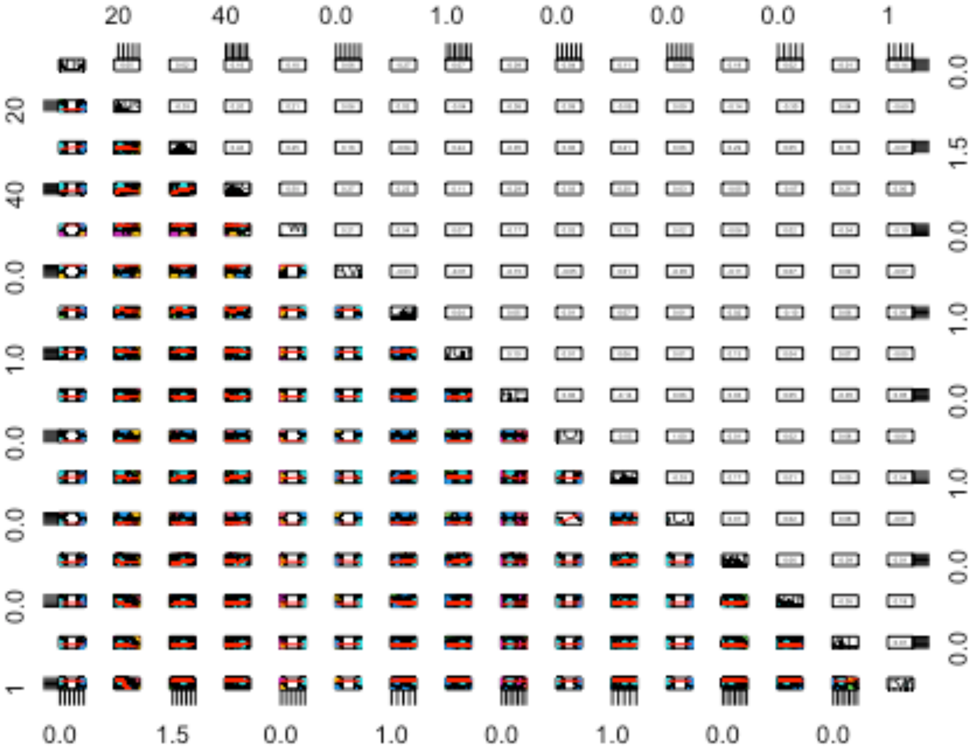
2. 1st technique : k-means clustering

We think that K-means clustering is a good technique for this dataset. We already know that we have 7 categories for obesity, so it is interesting to see if the technique is able to separate the data into the right clusters.

2.1. Data visualisation

We use the ´psych´ package to visualize the data

```
#################### DATA VISUALISATION

pairs.panels(data[1:16],
            ellipses = F,
            pch= 21,
            bg = data$Obesity)
```

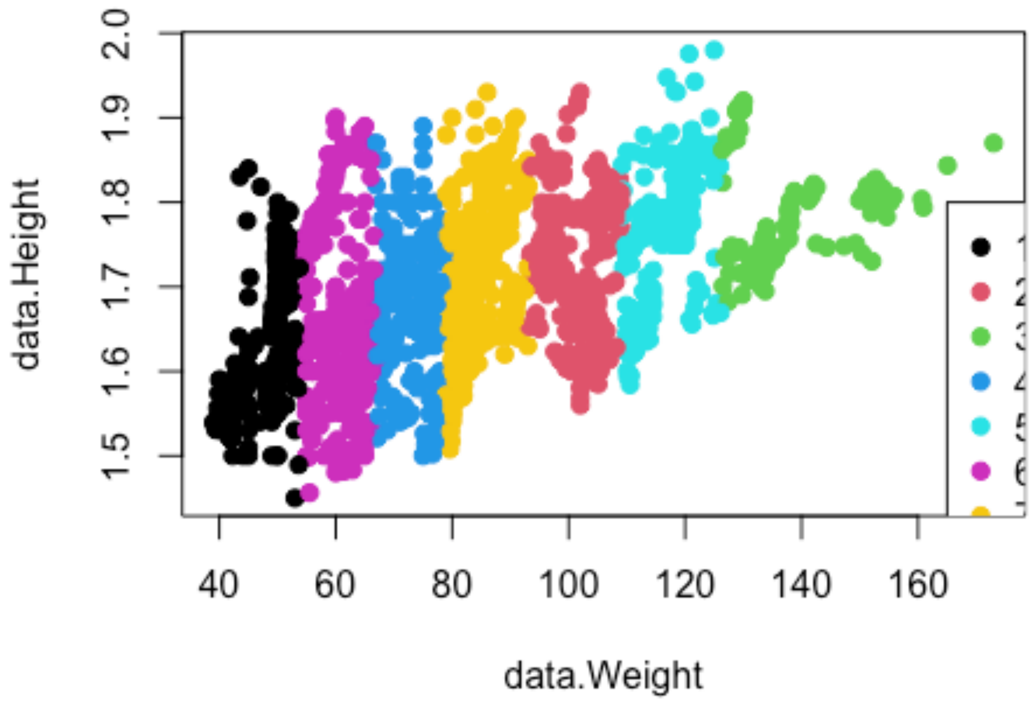


The best segregating variables seem to be ´weight´ and ´height´

2.2. K-means algorithm

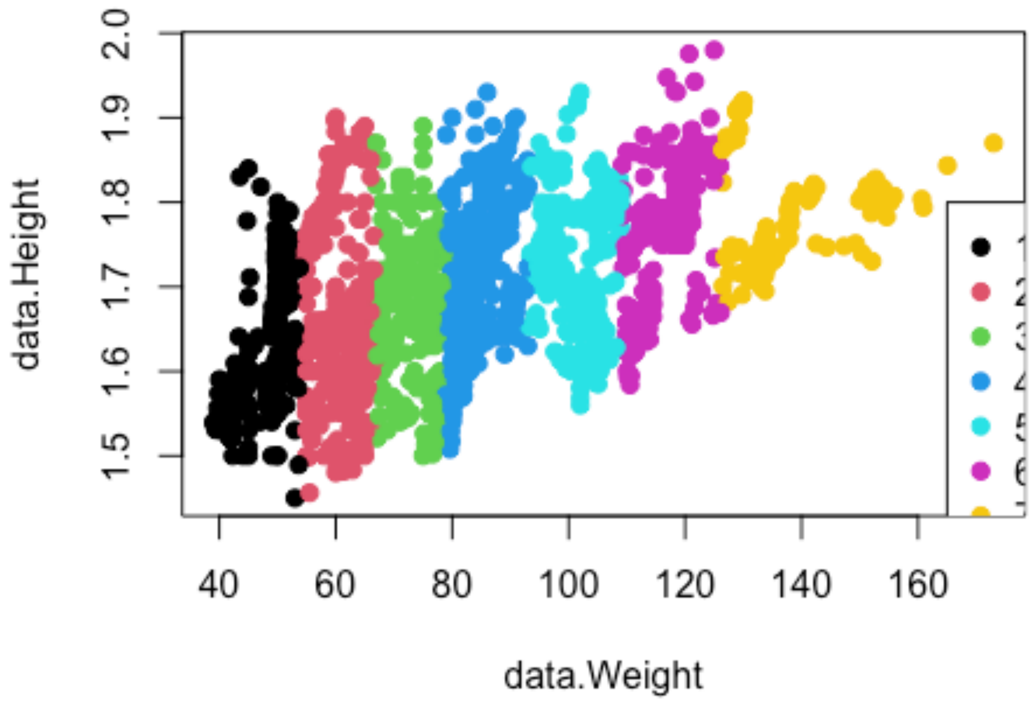
We use the k-means algorithm from the ´stats´ package.

```
#################### K-MEANS ALGORITHM

datakmeans <- data.frame(data$Weight, data$Height, data$Obesity)

# call kmeans again but this time passing the centers calculated in the
previous step
km <- kmeans(datakmeans[,1:2], 7)

# plot of the results
plot(datakmeans[1:2],
     col=km$cluster,
     pch=19)
legend(165, 1.8,
       c("1","2", "3", "4", "5", "6", "7"),
       pch=19,
       col=c(1:7))
```



The clusters aren’t in the right order. We modify the cluster “names” to then measure accuracy :

```
#################### K-MEANS CLUSTER ORDERING

# re-order kmeans clusters according to mean weight
ordered_center <- order(km$centers[,1])
ordered_labels <- match(km$cluster,ordered_center)
# plot of the results
plot(datakmeans[1:2],
     col=ordered_labels,
     pch=19)
legend(165, 1.8,
       c("1","2", "3", "4", "5", "6", "7"),
       pch=19,
       col=c(1:7))
```



The clusers are now in the right order. We measure the accuracy of the technique with a confusion matrix :

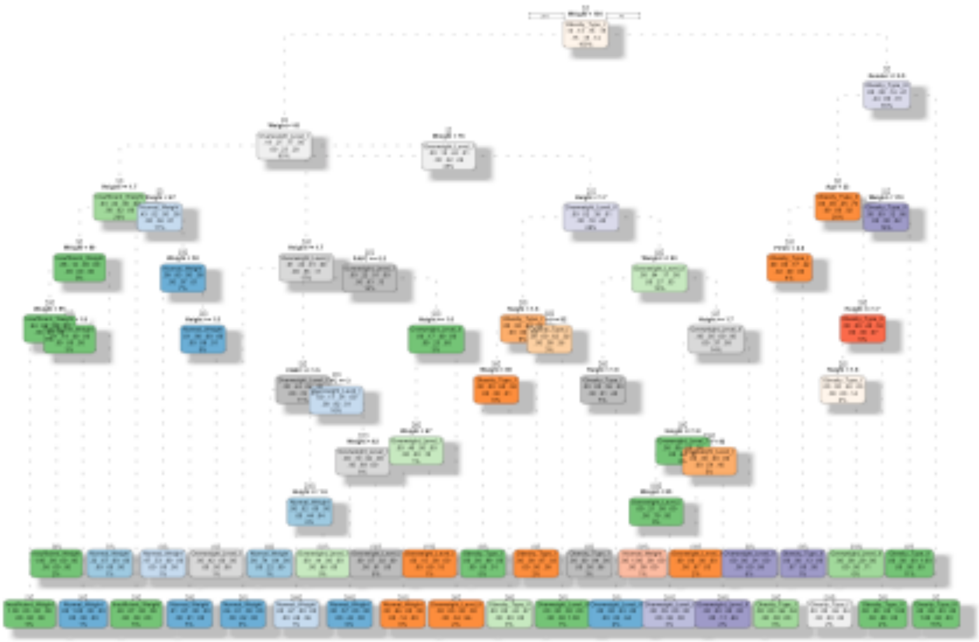
```
#################### K-MEANS CONFUSION MATRIX
cm <- table(label=data$Obesity, cluster=ordered_labels)
cm ; cat( sum(diag(cm)) / sum(cm) )

##                        cluster
## label                     1   2   3   4   5   6   7
##   Insufficient_Weight 208  64   0   0   0   0   0
##   Normal_Weight        57 141  73  16   0   0   0
##   Obesity_Type_I        0   0  28 156 146  21   0
##   Obesity_Type_II       0   0   0   1  61 215  20
##   Obesity_Type_III      0   0   0   0  81 110 133
##   Overweight_Level_I    2  53 144  91   0   0   0
##   Overweight_Level_II   0  22  59 183  26   0   0

## 0.2174325
```

The technique is not very accurate, only 21% of the clustering is right.

3. 2nd technique : Decision tree

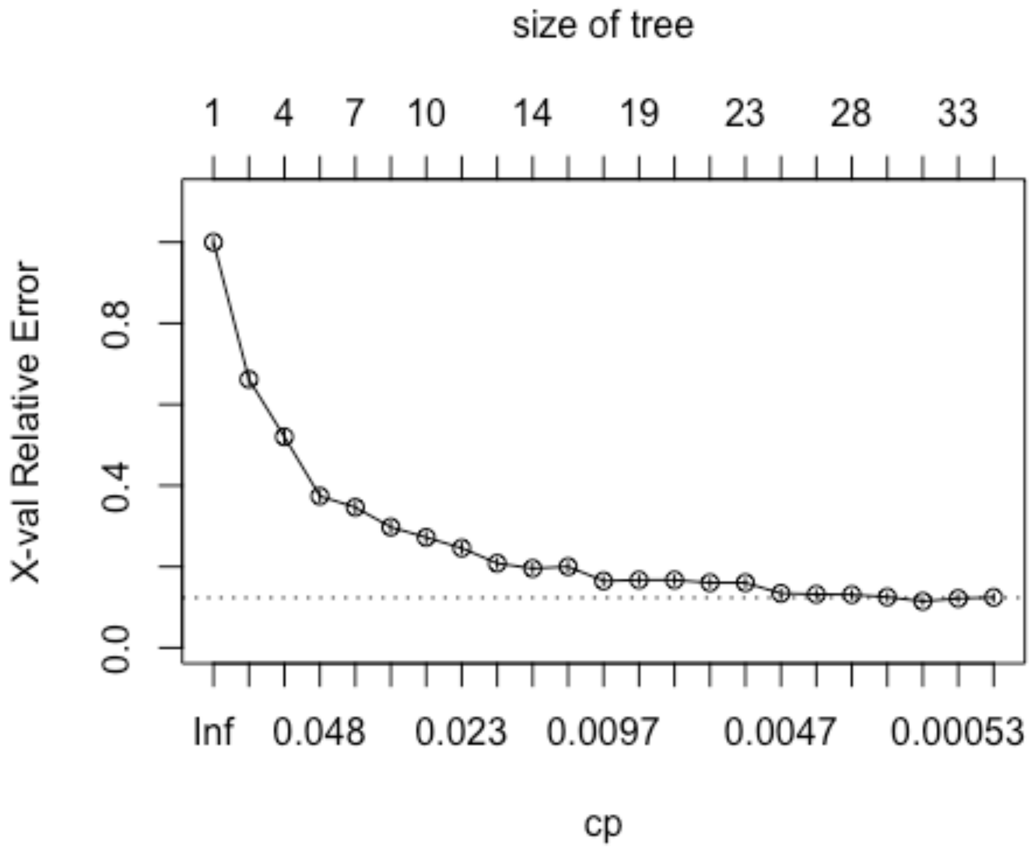
3.1. Separate train and test sets

```
##################### TRAIN AND TEST SETS

n <- nrow(data)
sel <- sort(sample.int(n, n/4))
data.train <- data[-sel,]
data.test <- data[sel,]

##################### DECISION TREE ALGORITHM

h <- rpart(Obesity ~ . , data = data.train, method ='class', cp=0, xval=500)

fancyRpartPlot(h, caption = NULL, type = 1)

## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```

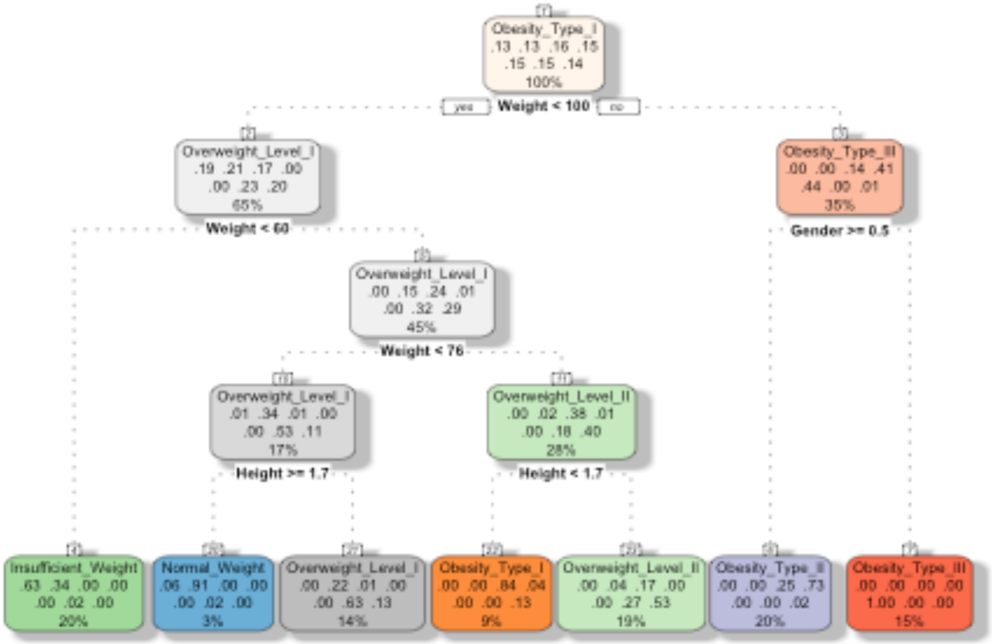


The tree is too big, we need to truncate it. We adjusted the cp value until we had 7 categories.

```
#################### DECISION TREE PRUNING

plotcp(h)
```



```
h_pruned <- prune(h, cp=0.03)

fancyRpartPlot(h_pruned, caption=NULL, type=2)
```



The second tree looks better, and we have indeed our 7 categories. We can now measure its accuracy :

```
##################### DECISION TREE EVALUATION

obesity_pred <- predict(h_pruned, data.test, type= 'class')

conf_table <- table(true=data.test$Obesity, predicted = obesity_pred)

n <- sum(conf_table)
error = (n - sum(diag(conf_table)))/n

cat(sprintf("The relative prediction error is %4.1f%%",error*100))

## The relative prediction error is 33.0%
```

For such a dataset and mini-project, an error of ´{r} error*100´% is not that bad.