R project

**Dataset**:

File name: heart rate.xlsx. Part of our project will do the monitoring of heart rate, then let the Robert do necessary action when the wear has emergent situation.

**k-Means**

> heart = read.csv("Desktop/Heart rate.csv")

> heart.feature = heart

> results <- kmeans(heart.feature, 3)

> results

K-means clustering with 3 clusters of sizes 21, 42, 37

Cluster means:

X Heat.Rate

1 20.71429 111.57143

2 79.40476 84.54762

3 34.59459 72.89189

Clustering vector:

[1] 1 1 1 1 1 3 1 1 1 1 3 1 3 3 1 3 3 3 3 3 1 3 3 3 1 3 1 1 1 3 3 3 1 3 3 3 3 3 1 3 3

[42] 3 3 3 3 3 1 3 3 3 3 1 3 3 3 3 2 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

[83] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:

[1] 15707.43 19512.52 11782.49

(between\_SS / total\_SS = 63.9 %)

Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss"

[6] "betweenss" "size" "iter" "ifault"

> heart = read.csv("Desktop/Heart rate.csv")

> heart.feature = heart

> results <- kmeans(heart.feature, 3)

> results

K-means clustering with 3 clusters of sizes 14, 48, 38

Cluster means:

Heart.Rate

1 127.57143

2 69.56250

3 91.21053

Clustering vector:

[1] 1 3 3 3 3 2 1 1 3 3 2 3 2 2 3 2 2 3 2 2 3 2 2 2 3 2 1 1 3 2 3 2 1 2 2 3 2 2 1 2 2

[42] 2 2 3 3 2 1 2 3 2 2 1 2 2 2 3 3 1 1 2 2 3 2 3 3 3 3 2 3 2 3 2 2 3 3 2 2 2 3 3 1 3

[83] 2 2 2 2 3 2 3 1 1 3 2 3 3 2 2 3 2 3

Within cluster sum of squares by cluster:

[1] 3563.429 2969.812 2208.316

(between\_SS / total\_SS = 81.4 %)

Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss"

[6] "betweenss" "size" "iter" "ifault"

**Decision tree**

install.packages("rpart")

install.packages("caret")

install.packages("rpart.plot")

library(rpart.plot)

library(rpart)

library(caret)

> model <- rpart(Status~HeartRate, heart)

> pred <- predict(model, heart, type="class")

> c <- confusionMatrix(pred, heart$Status)

> print(c)

> prp(model)

Result:

Confusion Matrix and Statistics

Reference

Prediction Higher Lower Normal

Higher 9 0 0

Lower 0 10 0

Normal 0 0 81

Overall Statistics

Accuracy : 1

95% CI : (0.9638, 1)

No Information Rate : 0.81

P-Value [Acc > NIR] : 7.055e-10

Kappa : 1

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: Higher Class: Lower Class: Normal

Sensitivity 1.00 1.0 1.00

Specificity 1.00 1.0 1.00

Pos Pred Value 1.00 1.0 1.00

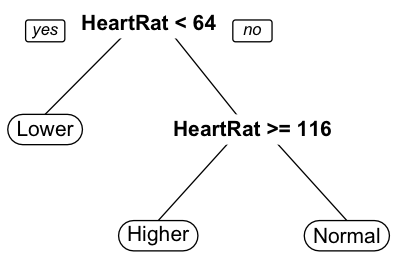
Neg Pred Value 1.00 1.0 1.00

Prevalence 0.09 0.1 0.81

Detection Rate 0.09 0.1 0.81

Detection Prevalence 0.09 0.1 0.81

Balanced Accuracy 1.00 1.0 1.00



**k-Medians**

> install.packages("flexclust")

> library(flexclust)

> x<-rbind(heart$HeartRate)

> x<- t(x)

> median=kcca(x,3,family=kccaFamily("kmedians"))

> print(median)

Results:

kcca object of family ‘kmedians’

call:

kcca(x = x, k = 3, family = kccaFamily("kmedians"))

cluster sizes:

1 2 3

10 50 40

**Expectation Maximisation (EM)**

> install.packages("EMCluster")

> library(EMCluster, quietly = TRUE)

> ret.em <- init.EM(x, nclass = 3, method = "em.EM")

> emobj <- simple.init(x, nclass = 3)

> ret.init <- emcluster(x, emobj, assign.class = TRUE)

> ret.init

Results:

Method:

n = 100, p = 1, nclass = 3, flag = , logL = -438.7302.

nc:

[1] 9 46 45

pi:

[1] 0.07364 0.39077 0.53558

>

> ret <- emcluster(x, emobj, assign.class = TRUE)

> summary(ret)

Method:

n = 100, p = 1, nclass = 3, flag = , total parameters = 8,

logL = -438.7302, AIC = 893.4605, BIC = 914.3018.

nc:

[1] 9 46 45

pi:

[1] 0.07364 0.39077 0.53558

**Hierarchical Clustering**

> d<-dist(as.matrix(x))

> hc<-hclust(d)

> plot(hc)

