Notebook 2 Classification

Umar Ali-Salaam, Carolline Osei

2022-10-23

Source: https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset

This is a dataset based off of 70,000 records of patient data (Heart Related). Columns (13): ID, Age, Height(cm), Weight(kg), Gender, Systolic Blood Pressure (AP_HIGH), Diastolic Blood Pressure (AP LOW), Cholesterol, Glucose, Smoking, Alcohol Intake, Physical Activity, Presence or Absence of cardiovascular disease.

The .csv file needed to be edited a bit in Microsoft Excel before using it in R. I just performed a split column delimiter function around semicolons, to divide the singular column that existed into 13. Each row had 13 variables in 1 column separated by semicolons, the function I ran split it up into 13 columns, making a 70,000 x 13 table.

https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings

Visit the website above to better understand Systolic and Diastolic Blood Pressure

Cleaning Data

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
library(flexclust)
```

```
## Warning: package 'flexclust' was built under R version 4.1.3

## Loading required package: grid

## Loading required package: lattice

## Loading required package: modeltools

## Loading required package: stats4
```

```
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 4.1.3

## Package 'mclust' version 5.4.10
## Type 'citation("mclust")' for citing this R package in publications.
```

```r
# Read in .csv file
heart <- read.csv("cardio_train.csv")


# Clean out any rows that have an unrealistic blood pressure (AP_HIGH & AP_LOW)
# They looked to be input errors by the person who made the data set
s <- subset(heart, AP_HIGH > 50)

s1 <- subset(s, AP_HIGH < 200)

s2 <- subset(s1, AP_LOW > 25)

s3 <- subset(s2, AP_LOW < 200)


# Removing little people(4'10") and Giants(7'3"), values are in cm
s4 <- subset(s3, HEIGHT > 147)

s5 <- subset(s4, HEIGHT < 220)


# Removing anyone below 90 lbs and above 375 lbs, the values are in kg
s6 <- subset(s5, WEIGHT > 40)

h1 <- subset(s6, WEIGHT < 180)

# AGE is in days so to get years i just divide by 365
h1$AGE <- (h1$AGE / 365)

# Removing people under 40
h <- subset(h1, AGE > 39)

# Checking for any NA values
# There is none
colSums(is.na(h))
```

```
##                ID               AGE            GENDER            HEIGHT
##                 0                 0                 0                 0
##            WEIGHT           AP_HIGH            AP_LOW       CHOLESTEROL
##                 0                 0                 0                 0
##           GLUCOSE             SMOKE           ALCOHOL PHYSICAL_ACTIVITY
##                 0                 0                 0                 0
##    CARDIO_DISEASE
##                 0
```

```r
# Everything that should be factored is factored
h$GENDER <- factor(h$GENDER)
h$CHOLESTEROL <- factor(h$CHOLESTEROL)
h$GLUCOSE <- factor(h$GLUCOSE)
h$SMOKE <- factor(h$SMOKE)
h$ALCOHOL <- factor(h$ALCOHOL)
h$PHYSICAL_ACTIVITY <- factor(h$PHYSICAL_ACTIVITY)
h$CARDIO_DISEASE <- factor(h$CARDIO_DISEASE)
```

```r
# There is now 67,685 rows

str(h)
```

```
## 'data.frame':    67685 obs. of  13 variables:
##  $ ID                : int  0 1 2 3 4 8 9 12 13 14 ...
##  $ AGE               : num  50.4 55.4 51.7 48.3 47.9 ...
##  $ GENDER            : Factor w/ 2 levels "1","2": 2 1 1 2 1 1 1 2 1 1 ...
##  $ HEIGHT            : int  168 156 165 169 156 151 157 178 158 164 ...
##  $ WEIGHT            : num  62 85 64 82 56 67 93 95 71 68 ...
##  $ AP_HIGH           : int  110 140 130 150 100 120 130 130 110 110 ...
##  $ AP_LOW            : int  80 90 70 100 60 80 80 90 70 60 ...
##  $ CHOLESTEROL       : Factor w/ 3 levels "1","2","3": 1 3 3 1 1 2 3 3 1 1 ...
##  $ GLUCOSE           : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 2 1 3 1 1 ...
##  $ SMOKE             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ALCOHOL           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PHYSICAL_ACTIVITY: Factor w/ 2 levels "0","1": 2 2 1 2 1 1 2 2 2 1 ...
##  $ CARDIO_DISEASE   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 1 1 ...
```

Train Test

```r
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.1.3
```

```r
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.1.3
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:flexclust':
##
##     bclust
```

```r
# Splittling data into train and test
# Reducing data size to 10,000 for the sake of the speed of the SVM function

h <- h[1:10000,]

split = sample.split(h, SplitRatio = 0.8)

hTrain = subset(h, split == TRUE)

hTest = subset(h, split == FALSE)
```

```
# Showing length of each dataset

cat("Train data has", nrow(hTrain), "rows.")
```

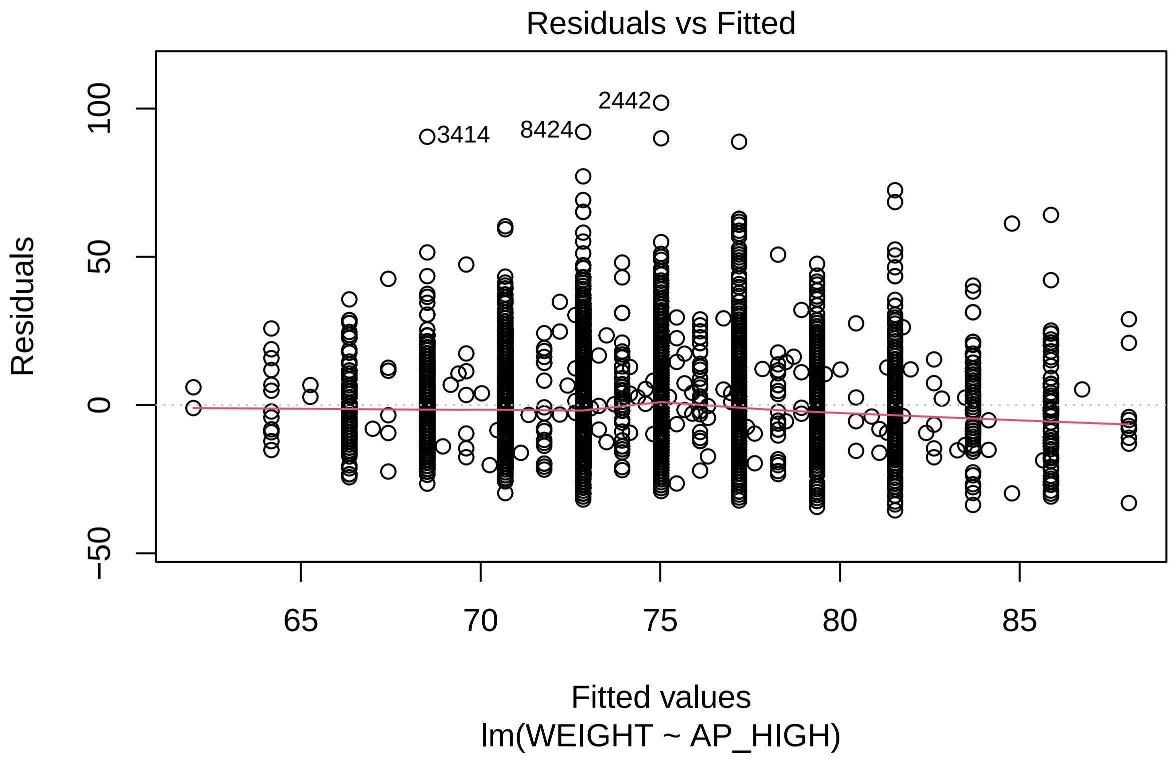
```
## Train data has 7693 rows.
```

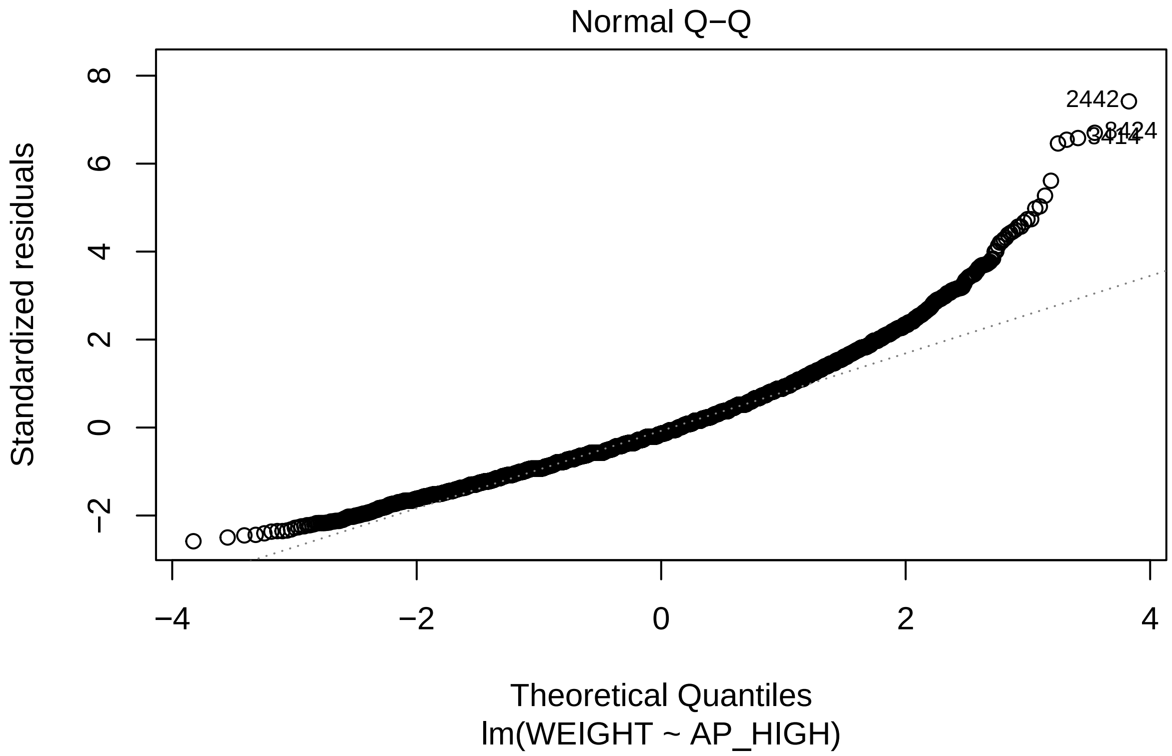
```
cat("\nTest data has", nrow(hTest), "rows.")
```

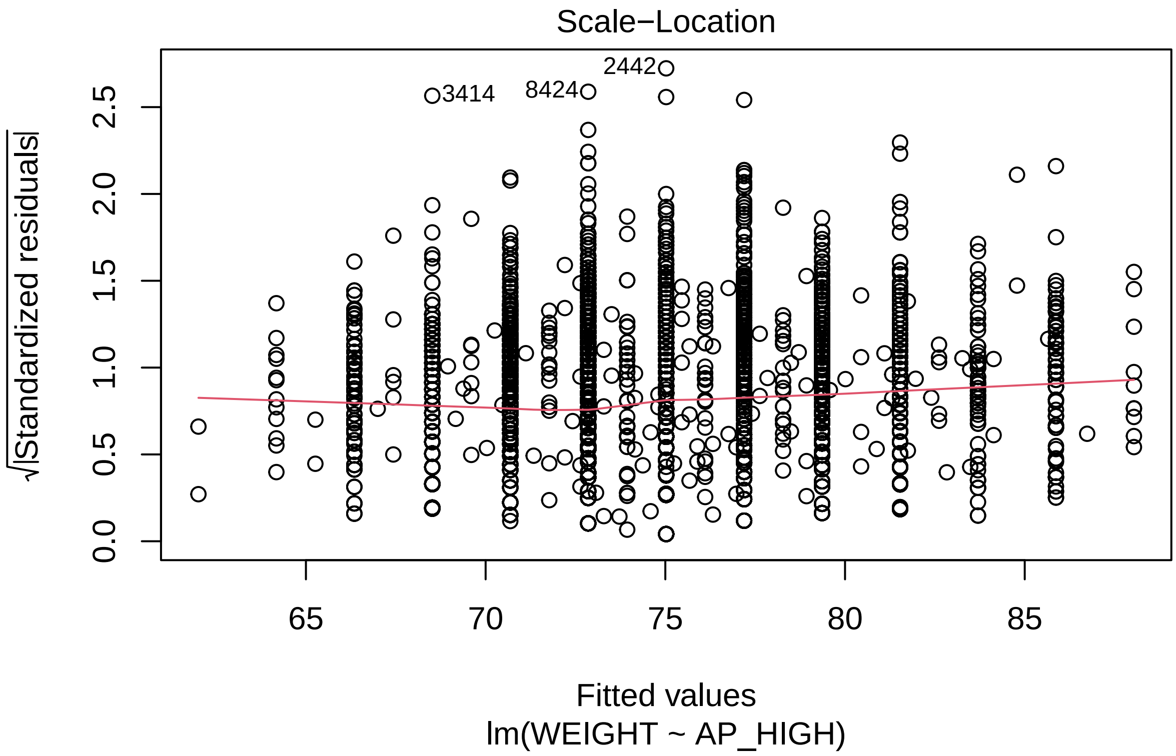
```
##
## Test data has 2307 rows.
```

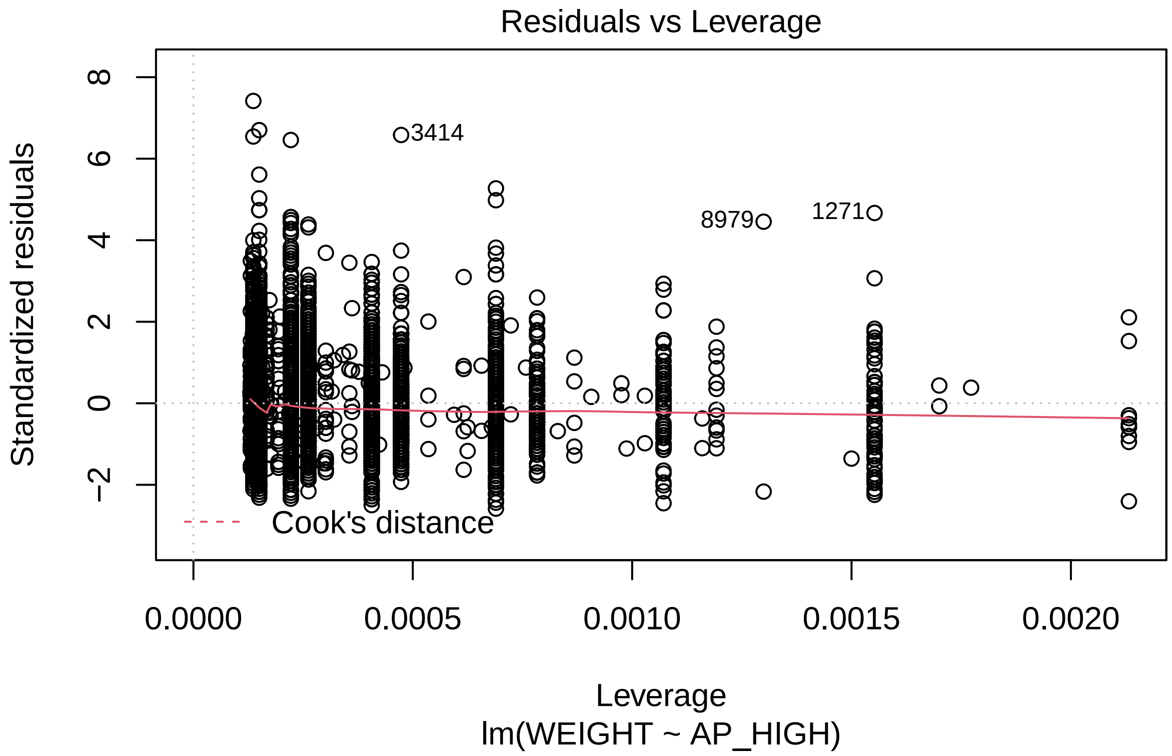
Exploring data

```
# Exploring data through linear regression, correlation test, and histogram
# distribution of AGE

LM <- lm(WEIGHT ~ AP_HIGH, data = hTrain)

plot(LM)
```




Normal Q−Q
Standardized residuals
8
6
4
2
0
−2
2442
3424
3414
−4
−2
0
2
4
Theoretical Quantiles
lm(WEIGHT ~ AP_HIGH)

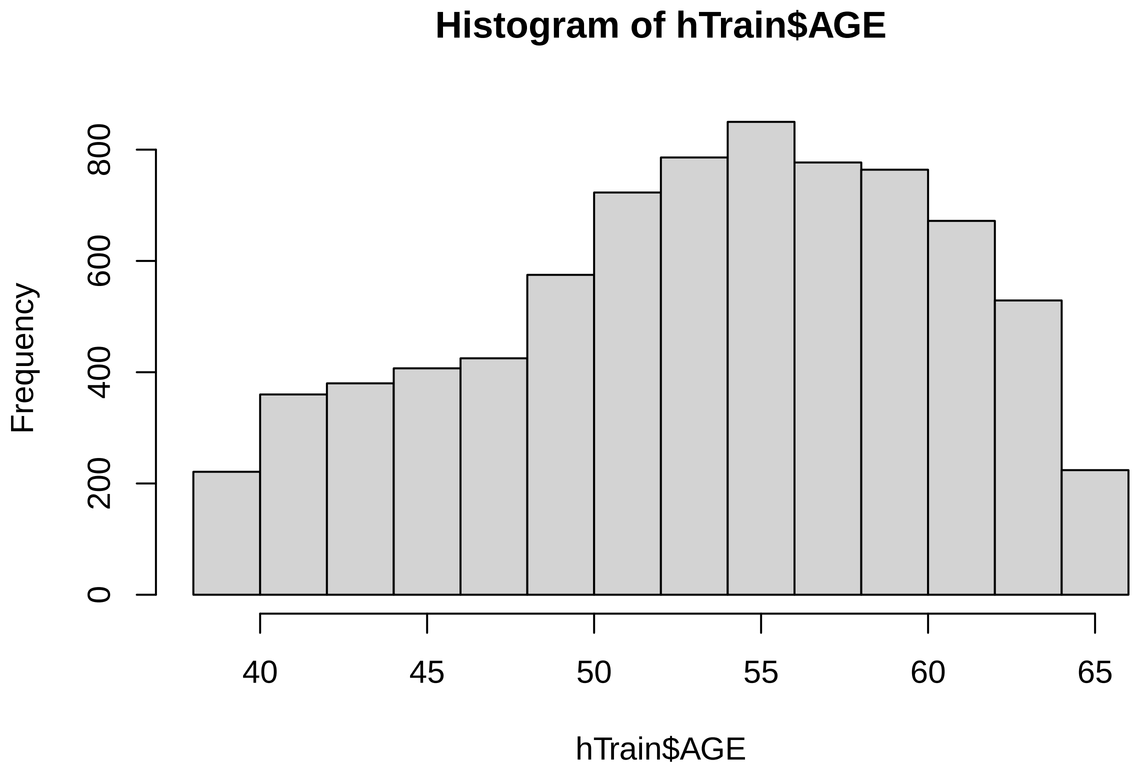

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
65
70
75
80
85
Fitted values
lm(WEIGHT ~ AP_HIGH)
3414
8424
2442



```
cor.test(hTrain$WEIGHT, hTrain$AP_HIGH)
```

```
##
##  Pearson's product-moment correlation
##
## data:  hTrain$WEIGHT and hTrain$AP_HIGH
## t = 22.435, df = 7691, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2267527 0.2687020
## sample estimates:
##       cor
## 0.2478435
```

```
hist(hTrain$AGE)
```



SVM Linear Kernel

```
# Perfoming SVM "Linear" on the presence of heart disease

svmLM10 <- svm(CARDIO_DISEASE ~ ., data = hTrain, kernel = "linear",
             cost = 10, gamma = 0.5, scale = TRUE)

svmLM1 <- svm(CARDIO_DISEASE ~ ., data = hTrain, kernel = "linear",
             cost = 1, gamma = 0.1, scale = TRUE)

summary(svmLM10)
```

```
##
## Call:
## svm(formula = CARDIO_DISEASE ~ ., data = hTrain, kernel = "linear",
##     cost = 10, gamma = 0.5, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  10
##
## Number of Support Vectors:  5044
```

```
##
##  ( 2522 2522 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
summary(svmLM1)
```

```
##
## Call:
## svm(formula = CARDIO_DISEASE ~ ., data = hTrain, kernel = "linear",
##     cost = 1, gamma = 0.1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  5046
##
##  ( 2524 2522 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

SVM Polynomial

```
# Perfoming SVM "Polynomial" on the presence of heart disease

svmP10 <- svm(CARDIO_DISEASE ~ ., data = hTrain, kernel = "polynomial",
             cost = 10, gamma = 0.5, scale = TRUE)

svmP1 <- svm(CARDIO_DISEASE ~ ., data = hTrain, kernel = "polynomial",
             cost = 1, gamma = 0.1, scale = TRUE)

summary(svmP10)
```

```
##
## Call:
## svm(formula = CARDIO_DISEASE ~ ., data = hTrain, kernel = "polynomial",
##     cost = 10, gamma = 0.5, scale = TRUE)
##
##
## Parameters:
```

```
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  10
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  4541
##
##  ( 2244 2297 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
summary(svmP1)
```

```
##
## Call:
## svm(formula = CARDIO_DISEASE ~ ., data = hTrain, kernel = "polynomial",
##     cost = 1, gamma = 0.1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  5155
##
##  ( 2572 2583 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

SVM Radial

```
# Perfoming SVM "Radial" on the presence of heart disease

svmR10 <- svm(CARDIO_DISEASE ~ ., data = hTrain, kernel = "radial",
             cost = 10, gamma = 0.5, scale = TRUE)

summary(svmR10)
```

```
##
```

```
## Call:
## svm(formula = CARDIO_DISEASE ~ ., data = hTrain, kernel = "radial",
##     cost = 10, gamma = 0.5, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  10
##
## Number of Support Vectors:  5403
##
##  ( 2625 2778 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```r
svmR1 <- svm(CARDIO_DISEASE ~ ., data = hTrain, kernel = "radial",
             cost = 1, gamma = 0.1, scale = TRUE)

summary(svmR1)
```

```
##
## Call:
## svm(formula = CARDIO_DISEASE ~ ., data = hTrain, kernel = "radial",
##     cost = 1, gamma = 0.1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  4816
##
##  ( 2388 2428 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

Evaluation

```r
# Showing classification accuracy of SVM of each kernel
# through the presence of heart disease
```

```r
predLM10 <- predict(svmLM10, newdata = hTest)

cat("\n\nLinear, cost = 10, gamma = 0.5\n")
```

```
##
##
## Linear, cost = 10, gamma = 0.5
```

```r
table(predLM10, hTest$CARDIO_DISEASE)
```

```
##
## predLM10   0   1
##        0 987 404
##        1 200 716
```

```r
cat("\nAccuracy:", mean(predLM10 == hTest$CARDIO_DISEASE), "\n\n")
```

```
##
## Accuracy: 0.7381881
```

```r
predLM1 <- predict(svmLM1, newdata = hTest)

cat("\n\nLinear, cost = 1, gamma = 0.1\n")
```

```
##
##
## Linear, cost = 1, gamma = 0.1
```

```r
table(predLM1, hTest$CARDIO_DISEASE)
```

```
##
## predLM1   0   1
##       0 987 404
##       1 200 716
```

```r
cat("\nAccuracy:", mean(predLM1 == hTest$CARDIO_DISEASE), "\n\n")
```

```
##
## Accuracy: 0.7381881
```

```r
predP10 <- predict(svmP10, newdata = hTest)

cat("\n\nPolynomial, cost = 10, gamma = 0.5\n")
```

```
##
##
## Polynomial, cost = 10, gamma = 0.5
```

```
table(predP10, hTest$CARDIO_DISEASE)
```

```
##
## predP10   0   1
##       0 949 411
##       1 238 709
```

```
cat("\nAccuracy:", mean(predP10 == hTest$CARDIO_DISEASE), "\n\n")
```

```
##
## Accuracy: 0.7186823
```

```
predP1 <- predict(svmP1, newdata = hTest)

cat("\n\nPolynomial, cost = 1, gamma = 0.1\n")
```

```
##
##
## Polynomial, cost = 1, gamma = 0.1
```

```
table(predP1, hTest$CARDIO_DISEASE)
```

```
##
## predP1   0   1
##      0 964 385
##      1 223 735
```

```
cat("\nAccuracy:", mean(predP1 == hTest$CARDIO_DISEASE), "\n\n")
```

```
##
## Accuracy: 0.7364543
```

```
predR10 <- predict(svmR10, newdata = hTest)

cat("\n\nRadial, cost = 10, gamma = 0.5\n")
```

```
##
##
## Radial, cost = 10, gamma = 0.5
```

```
table(predR10, hTest$CARDIO_DISEASE)
```

```
##
## predR10   0   1
##       0 826 396
##       1 361 724
```

```
cat("\nAccuracy:", mean(predR10 == hTest$CARDIO_DISEASE), "\n\n")
```

```
##
## Accuracy: 0.6718682
```

```
predR1 <- predict(svmR1, newdata = hTest)

cat("\n\nRadial, cost = 1, gamma = 0.1\n")
```

```
##
##
## Radial, cost = 1, gamma = 0.1
```

```
table(predR1, hTest$CARDIO_DISEASE)
```

```
##
## predR1   0   1
##      0 955 365
##      1 232 755
```

```
cat("\nAccuracy:", mean(predR1 == hTest$CARDIO_DISEASE), "\n\n")
```

```
##
## Accuracy: 0.7412224
```

Analysis

```
# Explaining results

cat("  All of the accuarcies were fairly close to each other, at about 0.72. In terms\n
of changing the gamma and cost parameters, the biggest change seen was in the radial\n
kernel, a change of 0.05. This is due to the fact that the other two kernels don't\n
need to use gamma, so changing gamma for them is pointless. The gamma allows you\n
to change the curve fidelity, which in turn allows you to fit more data points on\n
one side compared to the other. So the higher the gamma, the more points you're\n
allowed to fit and vice versa.\n\n
  In terms of cost, it's similar to gamma in the fact that you can determine how\n
many points are included on either side of the line. Lowering it allows more points\n
to be included on one side, and raising it does the opposite. Unsurprisingly lowering\n
the cost increased the accuracy of each kernel, except for linear. It seemed to have\n
no affect on the linear kernel.")
```

```
##   All of the accuarcies were fairly close to each other, at about 0.72. In terms
##
## of changing the gamma and cost parameters, the biggest change seen was in the radial
##
## kernel, a change of 0.05. This is due to the fact that the other two kernels don’t
##
```

```
## need to use gamma, so changing gamma for them is pointless. The gamma allows you
##
## to change the curve fidelity, which in turn allows you to fit more data points on
##
## one side compared to the other. So the higher the gamma, the more points you're
##
## allowed to fit and vice versa.
##
##
##   In terms of cost, it's similar to gamma in the fact that you can determine how
##
## many points are included on either side of the line. Lowering it allows more points
##
## to be included on one side, and raising it does the opposite. Unsurprisingly lowering
##
## the cost increased the accuracy of each kernel, except for linear. It seemed to have
##
## no affect on the linear kernel.
```