Classification_EDA_SVM

Yijin Wang

2023-05-06

```
library(janitor)
library(tidyverse)
library(AppliedPredictiveModeling)
library(lattice)
library(caret)
library(corrplot)
library(GGally)
library(miscset)
library(ggpubr)
library(knitr)
library(rpart)
library(rpart.plot)
library(ranger)
library(kernlab)
```

1. Load Data

```
load("final_data.RData")
```

2. Train/test split

```
set.seed(2)
final_data <- final_data %>%
  mutate(binary_recovery_time = factor(binary_recovery_time))
levels(final_data$binary_recovery_time) = c("low", "high")
training_rows <- createDataPartition(final_data$binary_recovery_time,
                                     p = 0.8,
                                     list = F)
```

```
train_x <- model.matrix(binary_recovery_time~., final_data %>% select(-id, -recovery_time))[training_ro
train_y <- final_data$binary_recovery_time[training_rows]
test_x <- model.matrix(binary_recovery_time~., final_data %>% select(-id, -recovery_time))[-training_ro
test_y <- final_data$binary_recovery_time[-training_rows]
```

```
training_set <- final_data[training_rows,]
```

3.EDA

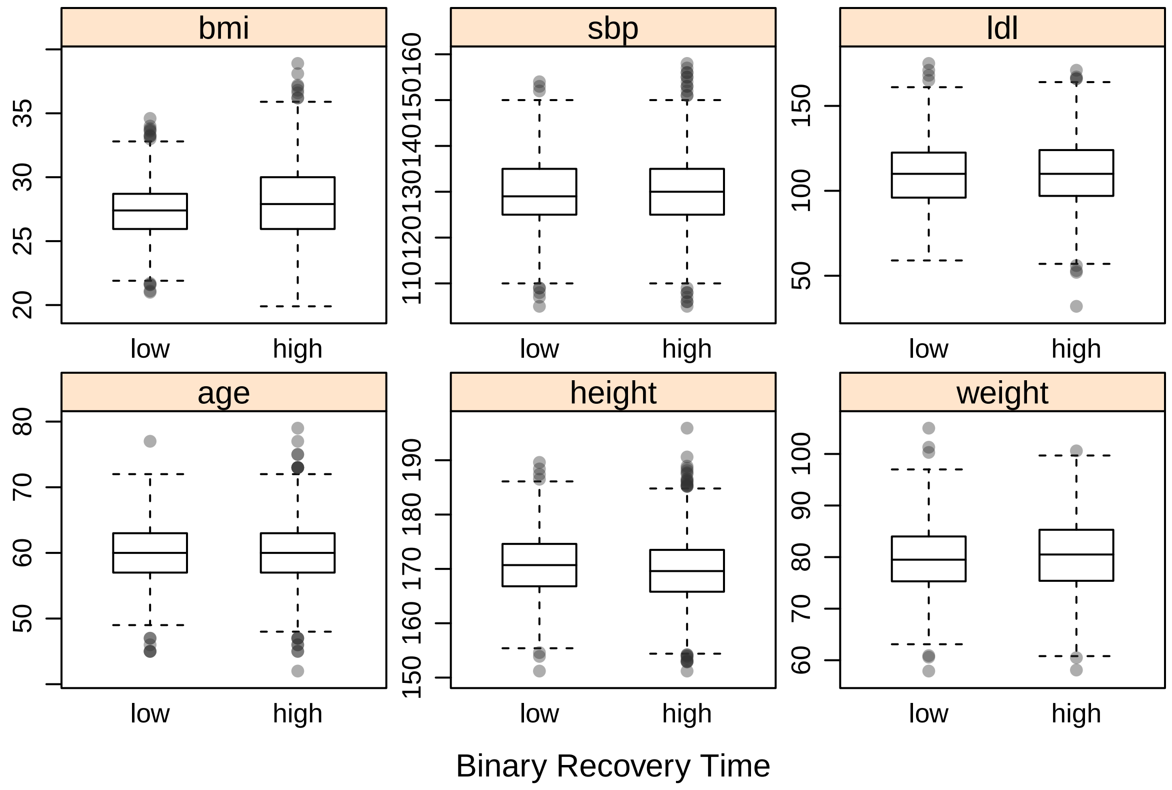
basic summary statistics

```
summary(training_set)
```

```
##        id             age         gender   race     smoking      height     
##  Min.   :    5   Min.   :42.00   0:1475   1:1840   0:1784   Min.   :151.2  
##  1st Qu.: 2510   1st Qu.:57.00   1:1413   2: 151   1: 832   1st Qu.:166.1  
##  Median : 4909   Median :60.00            3: 599   2: 272   Median :170.1  
##  Mean   : 4962   Mean   :60.07            4: 298            Mean   :169.9  
##  3rd Qu.: 7469   3rd Qu.:63.00                              3rd Qu.:173.9  
##  Max.   :10000   Max.   :79.00                              Max.   :195.9  
##      weight             bmi        hypertension diabetes      sbp     
##  Min.   : 57.90   Min.   :19.90   0:1525       0:2445   Min.   :105  
##  1st Qu.: 75.40   1st Qu.:25.98   1:1363       1: 443   1st Qu.:125  
##  Median : 80.10   Median :27.70                         Median :130  
##  Mean   : 80.08   Mean   :27.80                         Mean   :130  
##  3rd Qu.: 84.90   3rd Qu.:29.50                         3rd Qu.:135  
##  Max.   :105.00   Max.   :38.90                         Max.   :158  
##       ldl        vaccine  severity study    recovery_time   
##  Min.   : 32.0   0:1177   0:2617   A: 582   Min.   :  2.00  
##  1st Qu.: 97.0   1:1711   1: 271   B:1734   1st Qu.: 28.00  
##  Median :110.0                     C: 572   Median : 39.00  
##  Mean   :110.2                              Mean   : 42.96  
##  3rd Qu.:123.0                              3rd Qu.: 50.00  
##  Max.   :175.0                              Max.   :365.00  
##  binary_recovery_time
##  low : 896           
##  high:1992           
##                      
##                      
##                      
##
```

quantative variables

```
theme <- transparentTheme(trans = 0.4)
theme$plot.symbol$col = rgb(.2, .2, .2, .4)
theme$plot.symbol$pch = 16
theme$plot.line$col = rgb(1, 0, 0, .7)
theme$plot.line$lwd <- 2
trellis.par.set(theme)
featurePlot(x = training_set %>% dplyr::select(-recovery_time, -binary_recovery_time, -id, -gender, -ra
            y = training_set$binary_recovery_time,
            plot = "box",
            pch = "|",
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free")),
            auto.key = list(columns = 2),
            labels = c("Binary Recovery Time", ""))
```



## SVM

```
ctrl <- trainControl(method = "cv", number = 10)
# tunes over both cost and sigma
svmr_grid <- expand.grid(C = exp(seq(-2,2,len=40)),
                         sigma = exp(seq(-1,2,len=30)))
set.seed(2)

library(doParallel)
```

```
## Loading required package: foreach
```

```
##
## Attaching package: 'foreach'
```

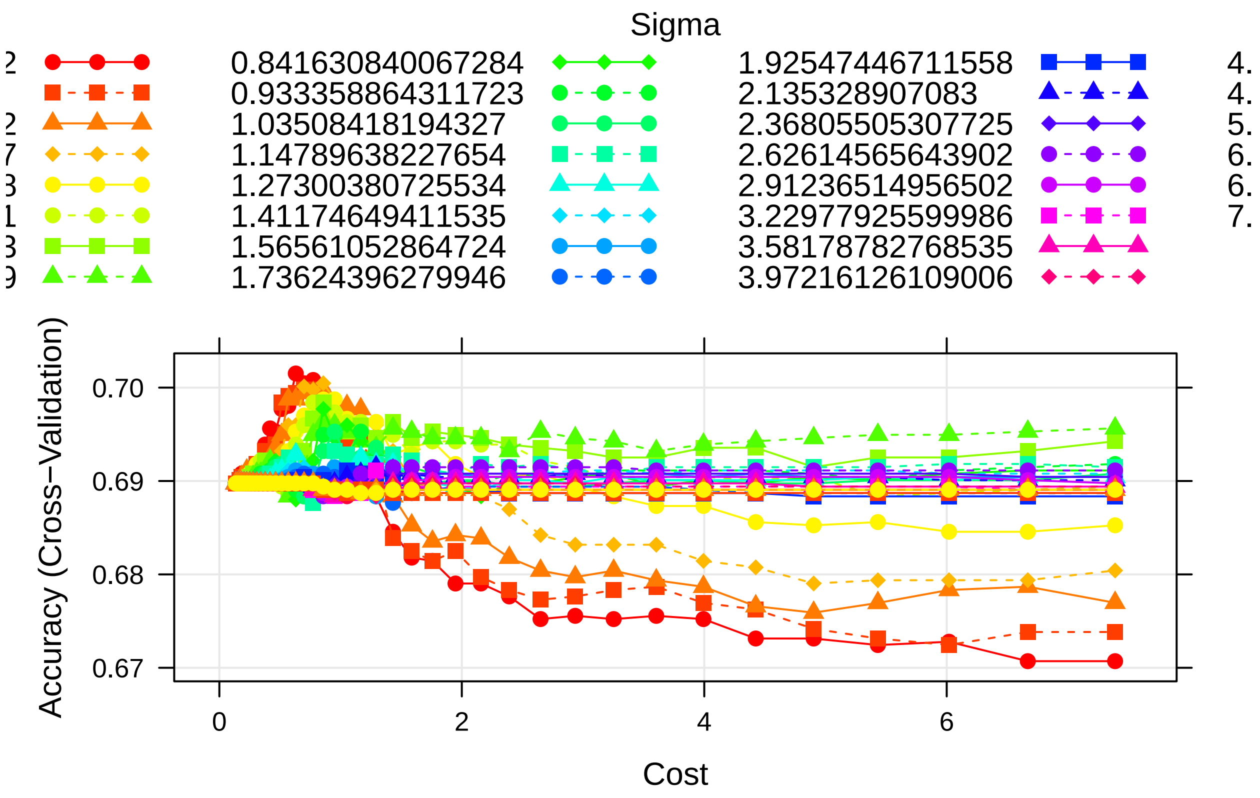
```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
Mycluster = makeCluster(detectCores()-2)
registerDoParallel(Mycluster)
svmr_fit <- train(binary_recovery_time ~.,
                  data = training_set %>% select(-id, -recovery_time),
                  method = "svmRadialSigma",
                  tuneGrid = svmr_grid,
                  trControl = ctrl)
stopCluster(Mycluster)
registerDoSEQ()

myCol<- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
              superpose.line = list(col = myCol))
plot(svmr_fit, highlight = TRUE, par.settings = myPar)
```



```
svmr_fit$bestTune
```

```
##         sigma         C
## 451 0.3678794 0.6303132
```

```
svm_train_prediction <- predict(svmr_fit$finalModel, newdata = train_x)
mean(svm_train_prediction!=train_y)
```

```
## [1] 0.2171053
```

```r
svm_test_prediction <- predict(svmr_fit$finalModel, newdata = test_x)
```

```r
mean(svm_test_prediction!=test_y)
```

```
## [1] 0.2884882
```