Homework 2 Report

Classification of liver malfunction severity (Logistic Regression)

Ian Effendi

September 14, 2021

Contents

Certification

> I certify that I indeed finished reading Ch. 3 from *An Introduction to Statistical Learning*, by James Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani.

Overview

In this assignment we will:

- Extract, load, and transform (*ELT*) the `Liver.txt` dataset.
- Perform exploratory data analysis (*EDA*) on the dataset.
- Fit and analyze a logistic regression model on the dataset.
- Perform multiple cross-validation tasks at different k-fold values ($k = 3$, $k = 10$).

This report was generated as a vignette in a formal R Package structured specifically for data analysis, with the following package requirements:

```
library(reshape2)
library(dplyr)
library(forcats)
library(ggplot2)
library(corrplot)
```

ELT

The `Liver.txt` file contains a header-less set of comma-separated fields, with one record per newline.

Extraction

We begin by getting the filepath for `Liver.txt`, which is stored in the package's `inst/extdata` folder:

```
# Retrieve paths to data files.
liver.filepath <- system.file(
  "extdata",
  "Liver.txt",
  package = "RIT.STAT745.HW2",
  mustWork = TRUE
)
```

Next, we prepare the column names:

```
# Prepare the variable names.
liver.variables <- c(
  # X1 through X5 are quantitative blood test results. (Unknown tests).
  "blood.1",
  "blood.2",
  "blood.3",
  "blood.4",
  "blood.5",
  # X6: No. of alcoholic beverages consumed.
  "drinks",
  # X7: Liver condition severity.
  "severity"
)
```

Then, we prepare the column types:

```
# Prepare the variable types.
liver.types <- readr::cols(
  # X1 through X5 - Quantitative blood test results.
  readr::col_integer(),
  readr::col_integer(),
  readr::col_integer(),
  readr::col_integer(),
  readr::col_integer(),
  # X6 - No. of alcoholic beverages.
  readr::col_double(),
  # X7 - Severity group.
  readr::col_factor()
)
```

Now we extract and save the `Liver.txt` file to place in our `data/` folder as a `liver_data.rda` file. This also allows us to refer to the data with `liver_data`.

```
# Read the dataset into memory.
liver.txt <- readr::read_csv(
  # File is located in inst/extdata
  file = liver.filepath,
  # First row is NOT a header row.
  col_names = liver.variables,
  # Column types known in advance.
  col_types = liver.types
)
```

```
# Store as tibble.
liver_data <- dplyr::as_tibble(liver.txt)

# Write *.csv file in data-raw/
readr::write_csv(liver_data, "data-raw/liver_data.csv")

# Save the imported Liver.txt tibble.
usethis::use_data(liver_data, overwrite = TRUE)
# v Saving 'liver_data' to 'data/liver_data.rda'
# * Document your data (see 'https://r-pkgs.org/data.html')
```

Loading

The `*.rda` data files are saved within the `data/` folder. When the package is installed (eg., `devtools::load_all("RIT.STAT745.HW2")` from the project directory), this attaches our `liver_data` tibble to the working environment:

```
library(RIT.STAT745.HW2)

str(liver_data)
# tibble [345 x 7] (S3: tbl_df/tbl/data.frame)
#  $ blood.1 : int [1:345] 85 85 86 91 87 98 88 88 92 90 ...
#  $ blood.2 : int [1:345] 92 64 54 78 70 55 62 67 54 60 ...
#  $ blood.3 : int [1:345] 45 59 33 34 12 13 20 21 22 25 ...
#  $ blood.4 : int [1:345] 27 32 16 24 28 17 17 11 20 19 ...
#  $ blood.5 : int [1:345] 31 23 54 36 10 17 9 11 7 5 ...
#  $ drinks  : num [1:345] 0 0 0 0 0 0 0.5 0.5 0.5 0.5 ...
#  $ severity: Factor w/ 2 levels "1","2": 1 2 2 2 2 2 1 1 1 1 ...
```

Transformation

Transformation of the dataset is minimal. In our case we want to encode our response variable `severity` to properly mention our *positive* and *negative* class labels. We can trivally declare a lookup table that provides us with an adequate data dictionary:

```
severity_map
```

status	value	code
severe	1	1
not severe	2	0

```
liver <- liver_data %>%
  mutate(severity = fct_recode(severity,
    "0" = "2",
    "1" = "1"
  )) %>%
  mutate(severity = fct_rev(severity))

print(paste(c("Levels in `liver$severity`: ", levels(liver$severity)), collapse = " "))
# [1] "Levels in `liver$severity`:  0 1"
```

EDA

Explore

```
# Structure of the dataframe.
dim(liver) # 345 observations and 3 variables.
# [1] 345   7
```

```
# X1 through X5 are blood test results.
# X6 is no. of alcoholic beverages drunk.
# X7 is severity response, encoded as "Group 1" = 1 and "Group 2" = 0
str(liver)
# tibble [345 x 7] (S3: tbl_df/tbl/data.frame)
#  $ blood.1 : int [1:345] 85 85 86 91 87 98 88 88 92 90 ...
#  $ blood.2 : int [1:345] 92 64 54 78 70 55 62 67 54 60 ...
#  $ blood.3 : int [1:345] 45 59 33 34 12 13 20 21 22 25 ...
#  $ blood.4 : int [1:345] 27 32 16 24 28 17 17 11 20 19 ...
#  $ blood.5 : int [1:345] 31 23 54 36 10 17 9 11 7 5 ...
#  $ drinks  : num [1:345] 0 0 0 0 0 0 0.5 0.5 0.5 0.5 ...
#  $ severity: Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 2 2 2 ...
```

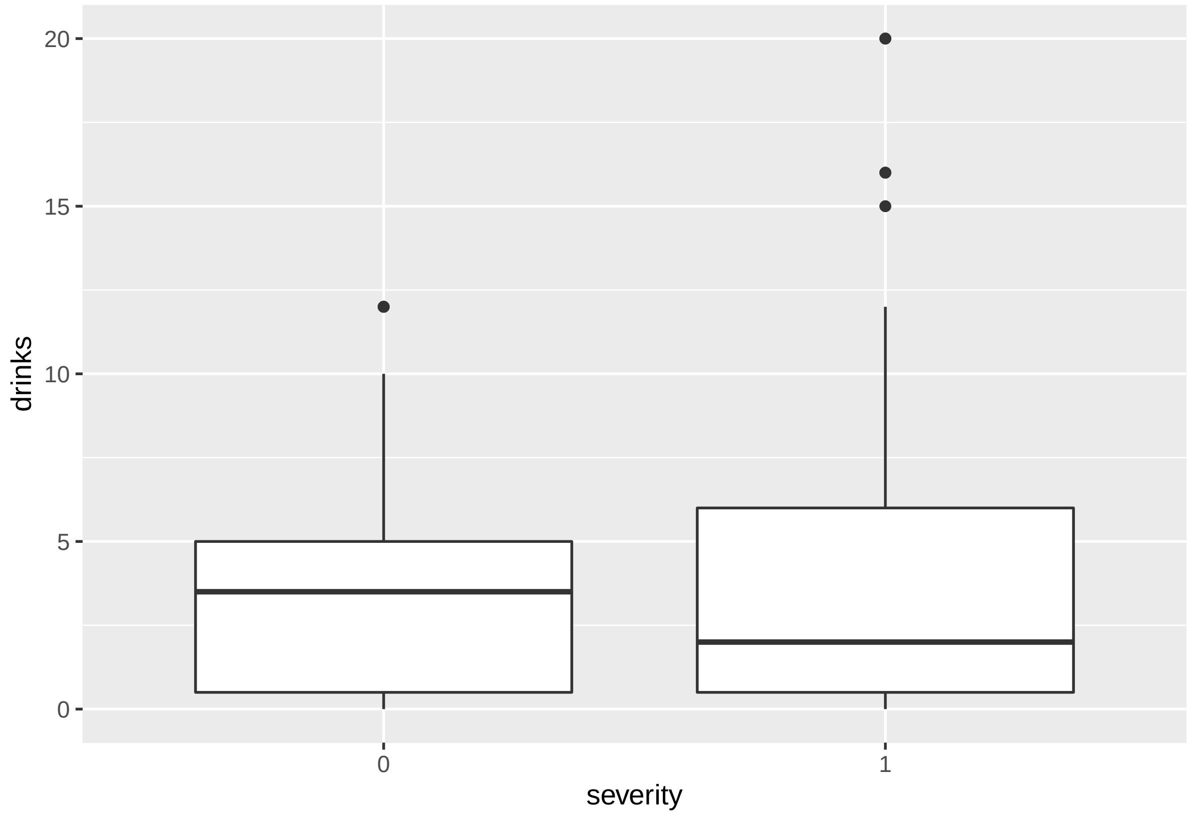
Analyze

```
# Summarize the blood test predictors.
summary(liver[,1:5]) # No units were provided with dataset.
#     blood.1          blood.2          blood.3
#  Min.   : 65.00   Min.   : 23.00   Min.   :  4.00
#  1st Qu.: 87.00   1st Qu.: 57.00   1st Qu.: 19.00
#  Median : 90.00   Median : 67.00   Median : 26.00
#  Mean   : 90.16   Mean   : 69.87   Mean   : 30.41
#  3rd Qu.: 93.00   3rd Qu.: 80.00   3rd Qu.: 34.00
#  Max.   :103.00   Max.   :138.00   Max.   :155.00
#     blood.4         blood.5
#  Min.   : 5.00   Min.   :  5.00
#  1st Qu.:19.00   1st Qu.: 15.00
#  Median :23.00   Median : 25.00
#  Mean   :24.64   Mean   : 38.28
#  3rd Qu.:27.00   3rd Qu.: 46.00
#  Max.   :82.00   Max.   :297.00
```

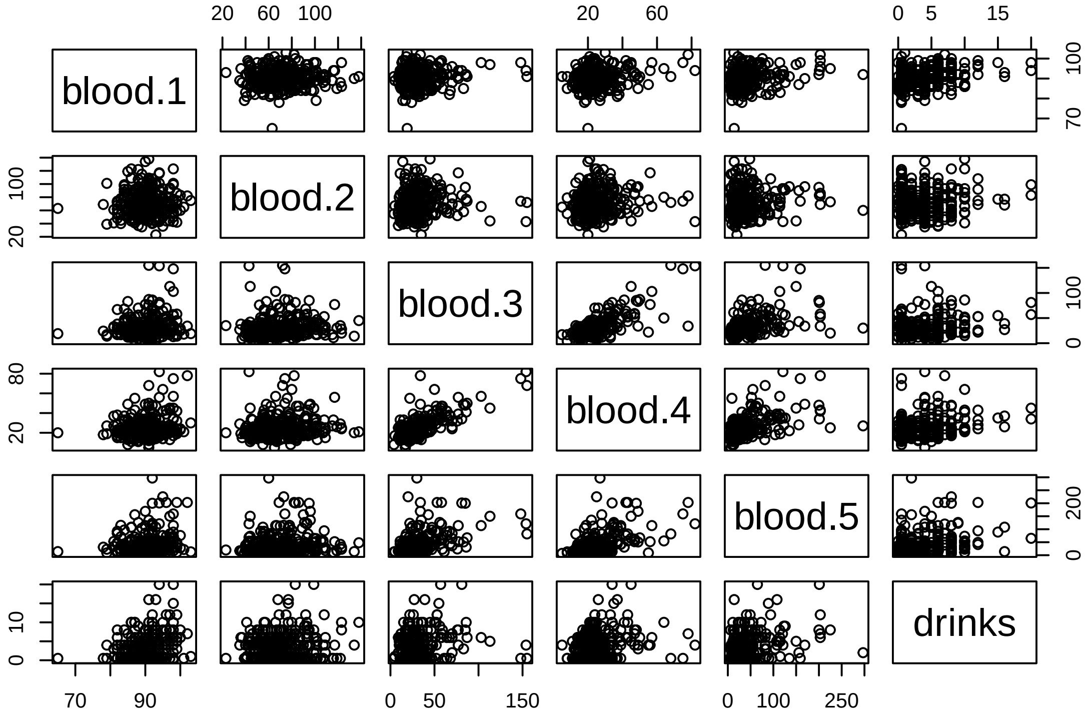
```
# Summarize the alcoholic beverage predictor.
summary(liver[,6])
#      drinks
#  Min.   : 0.000
#  1st Qu.: 0.500
#  Median : 3.000
#  Mean   : 3.455
#  3rd Qu.: 6.000
#  Max.   :20.000
# Median: 3 drinks, Mean: 3.5 drinks, skewed by Max: 20 drinks.
```

```
# Summarize the severity.
summary(liver[,7])
#  severity
#  0:200
#  1:145
# num(Group 2) > num(Group 1)
```

```
# Boxplot shows smaller median, wider spread in severe cases.
# Drinks may not be a strong predictor of liver malfunction.
ggplot(data = liver, mapping = aes(x = severity, y = drinks)) +
  geom_boxplot()
```



```
# blood.3 and blood.4 demonstrate multicollinearity.
pairs(liver %>% select(-severity))
```


blood.1
blood.2
blood.3
blood.4
blood.5
drinks
20
60
100
20
60
0
5
15
100
70
100
20
100
0
80
20
200
0
10
0
70
90
0
50
150
0
100
250

Classification

For this classification problem, we want to predict whether or not a patient's liver malfunction will be severe, given the predictors available in the dataset.

Theory

The probability of a test observation having the positive class label outcome ("Group 1") is described by p as such:

$$p = Pr(Y = 1)$$

Likewise, the probability of a test observation having the negative class label outcome ("Group 2") is described by q as such:

$$q = 1 - p = Pr(Y = 0)$$

$severity$ can only take one of two values: 0 or 1. We denote $p = Pr(severity = 1)$ and we will fit a logistic regression model:

$$ln[\frac{p}{(1-p)}] = \beta_0 + \beta_1(blood.1) + \cdots + \beta_5(blood.5) + \beta_6(drinks)$$

We will predict $severity = 1$ if $p >= \pi_0$ and as 0 otherwise. π_0 represents some arbitrary threshold probability we can select, but we'll begin with $\pi_0 = 0.5$.

Model Fitting

We can fit a logistic regression model with Group 1 as the positive class as follows:

```
liver.fit <- glm(
  severity ~ .,
  # Equivalent to:  severity ~ blood.1 + blood.2 + blood.3 + blood.4 + blood.5 + drinks,
  data = liver,
  family = "binomial"
)
```

```
summary(liver.fit)
#
# Call:
# glm(formula = severity ~ ., family = "binomial", data = liver)
#
# Deviance Residuals:
#     Min       1Q   Median       3Q      Max
# -1.9635  -0.9652  -0.5912   1.0482   2.4190
#
# Coefficients:
#              Estimate Std. Error z value Pr(>|z|)
# (Intercept) -5.990258   2.685458  -2.231 0.025706 *
# blood.1      0.063984   0.029645   2.158 0.030901 *
# blood.2      0.019525   0.006759   2.889 0.003870 **
# blood.3      0.064106   0.012300   5.212 1.87e-07 ***
# blood.4     -0.123198   0.024273  -5.076 3.86e-07 ***
# blood.5     -0.018947   0.005602  -3.382 0.000719 ***
# drinks       0.068080   0.040380   1.686 0.091795 .
# ---
# Signif. codes:
# 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# (Dispersion parameter for binomial family taken to be 1)
#
#     Null deviance: 469.47  on 344  degrees of freedom
# Residual deviance: 411.01  on 338  degrees of freedom
# AIC: 425.01
#
# Number of Fisher Scoring iterations: 5
```

The fitted model summary suggests `drinks` is not a significant predictor of severity outcomes, but `blood.1~blood.5` all appear to play an important effect on the response.

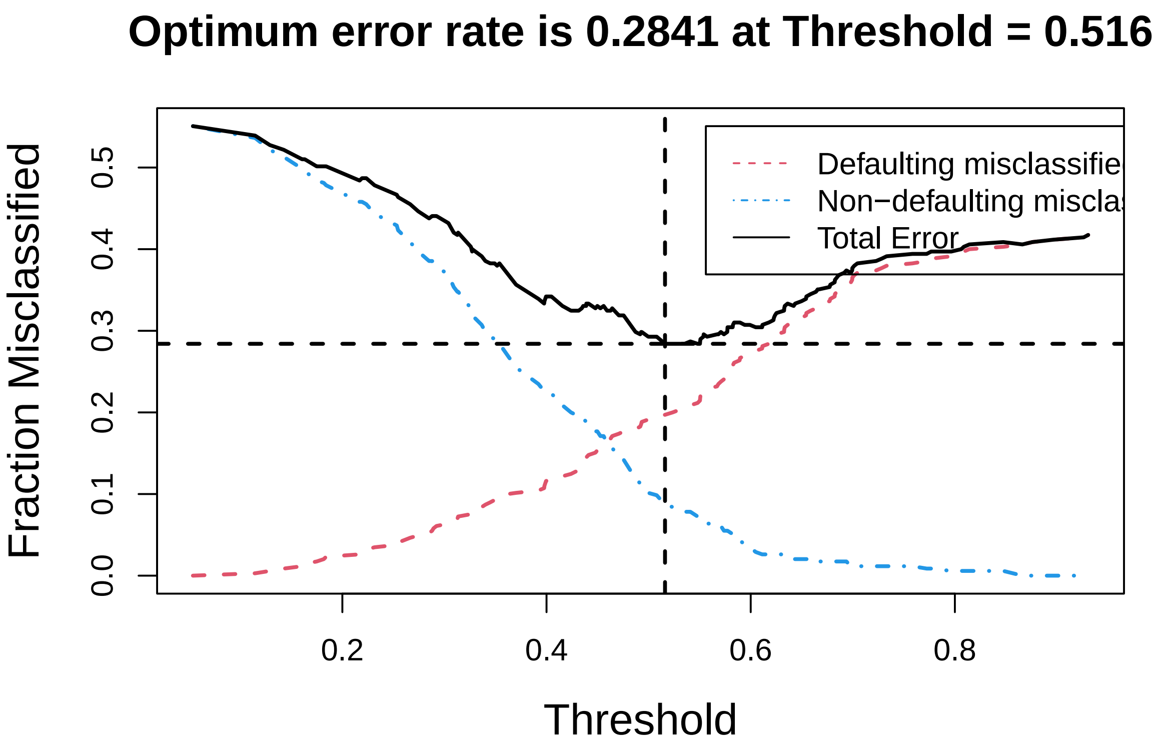
Model Performance

```
# See "R/utils-confmat.R" for helper function definitions.
# Find predictions and make misclassification table.
p.threshold = 0.5
liver.prediction <- (liver.fit$fitted >= p.threshold)
liver.truth <- map.where(liver$severity == "1")
Classif.table <- table(liver.prediction, liver.truth)
kable(Classif.table)
```

	FALSE	TRUE
FALSE	165	67
TRUE	35	78

```
ERR <- round(Error.rate.f(Classif.table), digits = 4)
# [1] "Total error rate (p >= 0.5): 0.2957"
# [2] "False negative error rate (p >= 0.5): 0.4203"
```

```
liver.ERR.rates <- All.error.rates.f(
  Truth = (liver$severity == "1"),
  Pred = liver.fit$fitted
)
Plot.Error.f(lista = liver.ERR.rates, cex=1.4)
```



As shown by the plot, the total error rate is optimized when the threshold is ≈ 0.516. The misclassification for the optimal case:

```
p.threshold = 0.516
liver.prediction <- (liver.fit$fitted >= p.threshold)
liver.truth <- map.where(liver$severity == "1")
Classif.table <- table(liver.prediction, liver.truth)
kable(Classif.table)
```

	FALSE	TRUE
FALSE	170	68
TRUE	30	77

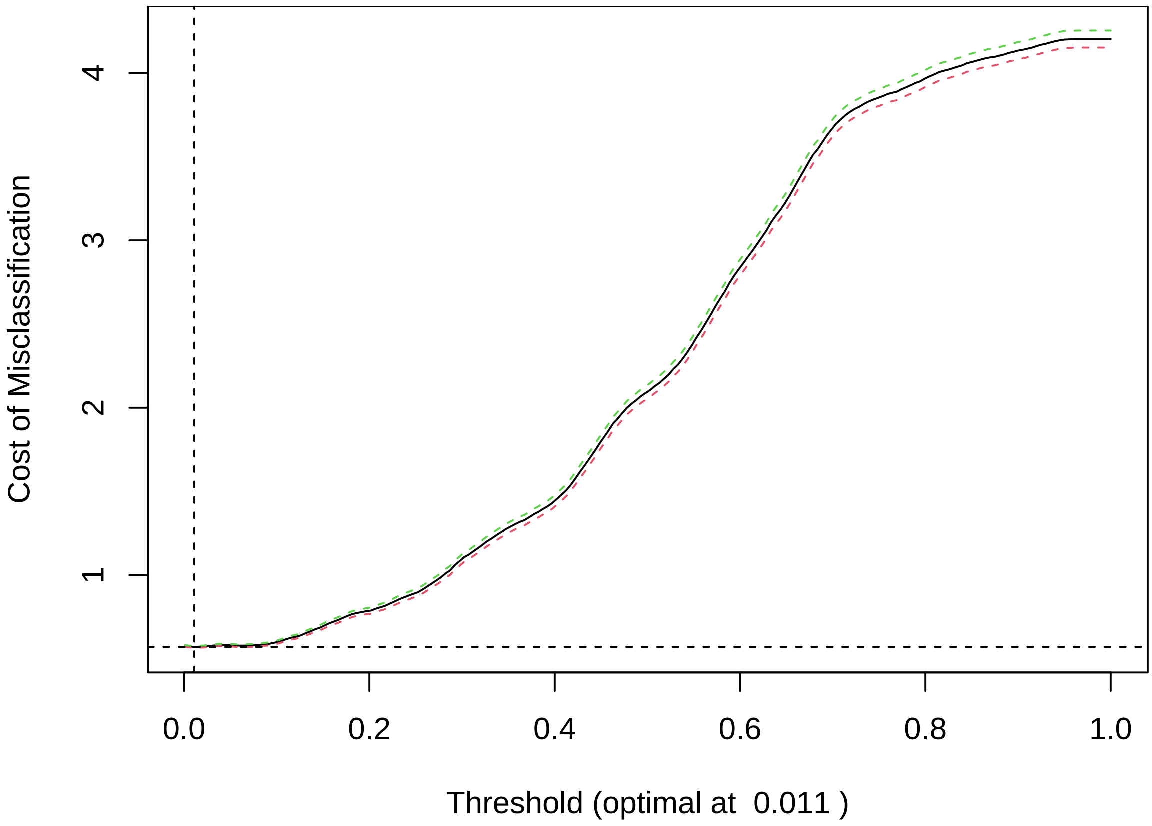
Cross-Valdiation

We use the following settings for our CV runs:

```
k.folds = c(10,3)
t.limit = c(0.001, 1.0)
n.thresholds = 200
n.rounds = 100
```

$k = 10$ -fold CV

Below is a 10-fold cross-validation, using 100 rounds:

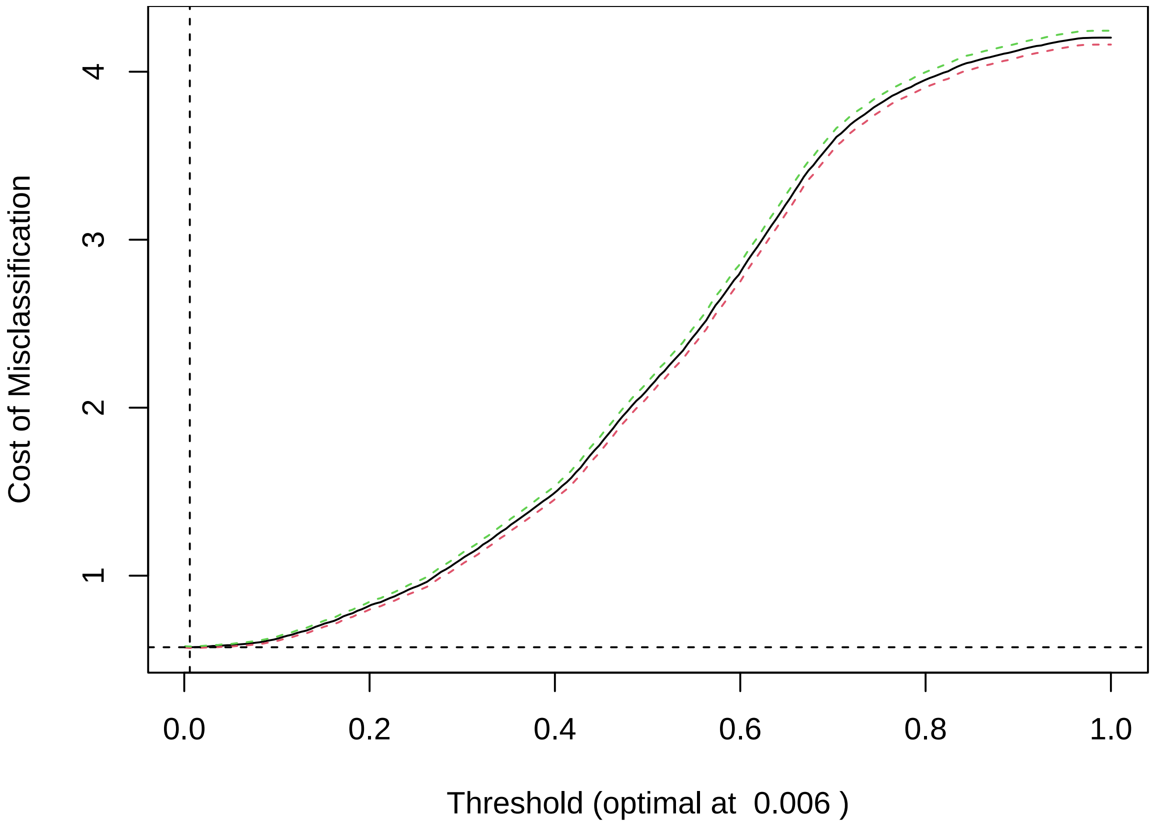


Minimum Cost	Minimum Threshold
0.5709235	0.0110402

```
# $Mean
#         [,1]   [,2]
# [1,] 142.08 138.76
# [2,] 157.87 144.97
#
# $StErr
#           [,1]       [,2]
# [1,] 1.3401854 1.37239156
# [2,] 0.8632397 0.01714466
```

$k = 3$ -fold CV

Below is a 3-fold cross-validation, using 100 rounds:



Minimum Cost	Minimum Threshold
0.574029	0.0060201

```
# $Mean
#         [,1]   [,2]
# [1,]  77.25  74.77
# [2,] 177.99 144.94
#
# $StErr
#           [,1]       [,2]
# [1,] 3.620324 3.66076260
# [2,] 1.247583 0.02386833
```

Session Information

This document was generated from an R Markdown Notebook (See the `vignettes/HW2_report.Rmd` in the package's sub-directory). The setup chunk for this document sets the root directory to the project root directory using the `rprojroot` package; all file paths are relative to the project root.

```
# R version 4.1.1 (2021-08-10)
# Platform: x86_64-w64-mingw32/x64 (64-bit)
# Running under: Windows 10 x64 (build 19042)
#
# Matrix products: default
#
# locale:
# [1] LC_COLLATE=English_United States.1252
# [2] LC_CTYPE=English_United States.1252
# [3] LC_MONETARY=English_United States.1252
# [4] LC_NUMERIC=C
# [5] LC_TIME=English_United States.1252
#
# attached base packages:
# [1] stats     graphics  grDevices datasets  utils
# [6] methods   base
#
# other attached packages:
# [1] RIT.STAT745.HW2_0.1.1 corrplot_0.90
# [3] ggplot2_3.3.5         forcats_0.5.1
# [5] dplyr_1.0.7           reshape2_1.4.4
# [7] rprojroot_2.0.2       knitr_1.36
#
# loaded via a namespace (and not attached):
#  [1] tinytex_0.34     bslib_0.3.0      tidyselect_1.1.1
#  [4] xfun_0.26        purrr_0.3.4      colorspace_2.0-2
#  [7] vctrs_0.3.8      generics_0.1.0   htmltools_0.5.2
# [10] usethis_2.0.1    yaml_2.2.1       utf8_1.2.2
# [13] rlang_0.4.11     jquerylib_0.1.4  pillar_1.6.3
# [16] glue_1.4.2       withr_2.4.2      DBI_1.1.1
# [19] bit64_4.0.5      lifecycle_1.0.1  plyr_1.8.6
# [22] stringr_1.4.0    munsell_0.5.0    gtable_0.3.0
# [25] codetools_0.2-18 evaluate_0.14    labeling_0.4.2
# [28] fastmap_1.1.0    tzdb_0.1.2       parallel_4.1.1
# [31] fansi_0.5.0      highr_0.9        Rcpp_1.0.7
# [34] readr_2.0.2      renv_0.14.0      scales_1.1.1
# [37] desc_1.4.0       jsonlite_1.7.2   vroom_1.5.5
# [40] farver_2.1.0     fs_1.5.0         bit_4.0.4
# [43] hms_1.1.1        digest_0.6.28    stringi_1.7.4
# [46] grid_4.1.1       cli_3.0.1        tools_4.1.1
# [49] sass_0.4.0       magrittr_2.0.1   tibble_3.1.3
# [52] crayon_1.4.1     pkgconfig_2.0.3  ellipsis_0.3.2
# [55] assertthat_0.2.1 rmarkdown_2.11   rstudioapi_0.13
# [58] R6_2.5.1         compiler_4.1.1
```