Final Project Code

Tianshu Liu, Lincole Jiang, Jiong Ma

Contents

```
library(tidyverse)
library(summarytools)
library(corrplot)
library(caret)
library(vip)
library(rpart.plot)
library(ranger)
```

1 Data Import

```
# import data
load("./recovery.RData")

set.seed(3196)
lts.dat <- dat[sample(1:10000, 2000),]
set.seed(2575)
lincole.dat <- dat[sample(1:10000, 2000),]
set.seed(5509)
amy.dat <- dat[sample(1:10000, 2000),]

dat1 <- lts.dat %>%
  merge(lincole.dat, all = TRUE) %>%
  na.omit() %>%
  select(-id) %>%
  mutate(
    gender = as.factor(gender),
    race = as.factor(race),
    smoking = as.factor(smoking),
    hypertension = as.factor(hypertension),
    diabetes = as.factor(diabetes),
    vaccine = as.factor(vaccine),
    severity = as.factor(severity),
    study = as.factor(study))

dat2 <- lts.dat %>%
  merge(amy.dat, all = TRUE) %>%
  na.omit() %>%
  select(-id) %>%
  mutate(
    gender = as.factor(gender),
    race = as.factor(race),
    smoking = as.factor(smoking),
    hypertension = as.factor(hypertension),
    diabetes = as.factor(diabetes),
    vaccine = as.factor(vaccine),
    severity = as.factor(severity),
    study = as.factor(study))

dat3 <- lincole.dat %>%
  merge(amy.dat, all = TRUE) %>%
  na.omit() %>%
  select(-id) %>%
```

```
  mutate(
    gender = as.factor(gender),
    race = as.factor(race),
    smoking = as.factor(smoking),
    hypertension = as.factor(hypertension),
    diabetes = as.factor(diabetes),
    vaccine = as.factor(vaccine),
    severity = as.factor(severity),
    study = as.factor(study))

dat <- dat1
summary(dat)
```

```
##       age        gender   race     smoking      height          weight
##  Min.   :45.00   0:1842   1:2372   0:2223   Min.   :151.2   Min.   : 56.70
##  1st Qu.:57.00   1:1781   2: 172   1:1034   1st Qu.:166.2   1st Qu.: 75.40
##  Median :60.00            3: 716   2: 366   Median :170.2   Median : 80.20
##  Mean   :60.06            4: 363            Mean   :170.2   Mean   : 80.13
##  3rd Qu.:63.00                              3rd Qu.:174.2   3rd Qu.: 84.80
##  Max.   :77.00                              Max.   :188.6   Max.   :103.40
##       bmi        hypertension diabetes      SBP             LDL        vaccine
##  Min.   :19.70   0:1891       0:3065   Min.   :102.0   Min.   : 28.0   0:1469
##  1st Qu.:25.80   1:1732       1: 558   1st Qu.:125.0   1st Qu.: 97.0   1:2154
##  Median :27.60                         Median :130.0   Median :110.0
##  Mean   :27.73                         Mean   :130.2   Mean   :110.5
##  3rd Qu.:29.40                         3rd Qu.:136.0   3rd Qu.:124.0
##  Max.   :39.80                         Max.   :158.0   Max.   :174.0
##  severity study    recovery_time
##  0:3289   A: 728   Min.   :  3.00
##  1: 334   B:2171   1st Qu.: 28.00
##           C: 724   Median : 38.00
##                    Mean   : 42.87
##                    3rd Qu.: 49.00
##                    Max.   :365.00
```

```
bin.dat1 <- dat1 %>%
  mutate(recovery_time = ifelse(recovery_time > 30, "gt30", "lt30")) %>%
  mutate(recovery_time = factor(recovery_time, levels = c("lt30", "gt30")))

bin.dat2 <- dat2 %>%
  mutate(recovery_time = ifelse(recovery_time > 30, "gt30", "lt30")) %>%
  mutate(recovery_time = factor(recovery_time, levels = c("lt30", "gt30")))

bin.dat3 <- dat3 %>%
 mutate(recovery_time = ifelse(recovery_time > 30, "gt30", "lt30")) %>%
  mutate(recovery_time = factor(recovery_time, levels = c("lt30", "gt30")))

bin.dat <- bin.dat1
summary(bin.dat)
```

```
##       age        gender   race     smoking      height          weight
##  Min.   :45.00   0:1842   1:2372   0:2223   Min.   :151.2   Min.   : 56.70
##  1st Qu.:57.00   1:1781   2: 172   1:1034   1st Qu.:166.2   1st Qu.: 75.40
##  Median :60.00            3: 716   2: 366   Median :170.2   Median : 80.20
```

```
##  Mean   :60.06            4: 363            Mean   :170.2   Mean   : 80.13
##  3rd Qu.:63.00                              3rd Qu.:174.2   3rd Qu.: 84.80
##  Max.   :77.00                              Max.   :188.6   Max.   :103.40
##       bmi        hypertension diabetes      SBP             LDL        vaccine
##  Min.   :19.70   0:1891       0:3065   Min.   :102.0   Min.   : 28.0   0:1469
##  1st Qu.:25.80   1:1732       1: 558   1st Qu.:125.0   1st Qu.: 97.0   1:2154
##  Median :27.60                         Median :130.0   Median :110.0
##  Mean   :27.73                         Mean   :130.2   Mean   :110.5
##  3rd Qu.:29.40                         3rd Qu.:136.0   3rd Qu.:124.0
##  Max.   :39.80                         Max.   :158.0   Max.   :174.0
##  severity study    recovery_time
##  0:3289   A: 728   lt30:1102
##  1: 334   B:2171   gt30:2521
##           C: 724
##
##
##
```

2 Data partition

```
# data partition
dat.matrix <- model.matrix(recovery_time ~ ., dat)[ ,-1]

set.seed(2023)
trainRows <- createDataPartition(y = dat$recovery_time, p = 0.8, list = FALSE)

train.dat <- dat[trainRows,]
train.bin.dat <- bin.dat[trainRows,]

train.dat.matrix <- model.matrix(~., train.dat)[, -1]
train.bin.dat.matrix <- model.matrix(~., train.bin.dat)[, -1]

train.x <- dat.matrix[trainRows,]
train.y <- dat$recovery_time[trainRows]
train.bin.y <- bin.dat$recovery_time[trainRows]

test.x <- dat.matrix[-trainRows,]
test.y <- dat$recovery_time[-trainRows]
test.bin.y <- bin.dat$recovery_time[-trainRows]
```

3 Primary Analysis

3.1 Exploratory analysis and data visualization

```
# data summary
st_options(plain.ascii = FALSE,
           style = "rmarkdown",
           dfSummary.silent = TRUE,
           footnote = NA,
           subtitle.emphasis = FALSE)
dfSummary(train.dat)
```

3.1.1 Data Frame Summary

train.dat
Dimensions: 2900 x 15
Duplicates: 0

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	age [numeric]	Mean (sd) : 60.1 (4.5) min < med < max: 45 < 60 < 77 IQR (CV) : 6 (0.1)	33 distinct values	: : . : : : : . . : : : : : : .	2900 (100.0%)	0 (0.0%)
2	gender [factor]	1. 0 2. 1	1468 (50.6%) 1432 (49.4%)	IIIIIIIIII IIIIIIIII	2900 (100.0%)	0 (0.0%)
3	race [factor]	1. 1 2. 2 3. 3 4. 4	1909 (65.8%) 132 (4.6%) 568 (19.6%) 291 (10.0%)	IIIIIIIIIIII III II	2900 (100.0%)	0 (0.0%)
4	smoking [factor]	1. 0 2. 1 3. 2	1763 (60.8%) 845 (29.1%) 292 (10.1%)	IIIIIIIIIII IIIII II	2900 (100.0%)	0 (0.0%)
5	height [numeric]	Mean (sd) : 170.2 (6) min < med < max: 151.2 < 170.1 < 188.6 IQR (CV) : 8 (0)	312 distinct values	: : : : . : : : : : : : .	2900 (100.0%)	0 (0.0%)
6	weight [numeric]	Mean (sd) : 80.2 (7) min < med < max: 57.1 < 80.3 < 103.4 IQR (CV) : 9.5 (0.1)	361 distinct values	. : . : : : : : : . . : : : : : : : .	2900 (100.0%)	0 (0.0%)
7	bmi [numeric]	Mean (sd) : 27.8 (2.7) min < med < max: 19.7 < 27.7 < 39.8 IQR (CV) : 3.6 (0.1)	160 distinct values	: . : : : : . : : : : : : : : .	2900 (100.0%)	0 (0.0%)
8	hypertension [factor]	1. 0 2. 1	1514 (52.2%) 1386 (47.8%)	IIIIIIIII IIIIIIII	2900 (100.0%)	0 (0.0%)
9	diabetes [factor]	1. 0 2. 1	2446 (84.3%) 454 (15.7%)	IIIIIIIIIIIIII III	2900 (100.0%)	0 (0.0%)
10	SBP [numeric]	Mean (sd) : 130.2 (8.1) min < med < max: 104 < 130 < 158 IQR (CV) : 11 (0.1)	54 distinct values	: : . : : : . . : : : : : : : :	2900 (100.0%)	0 (0.0%)
11	LDL [numeric]	Mean (sd) : 110.3 (19.9) min < med < max: 32 < 110 < 174 IQR (CV) : 27 (0.2)	116 distinct values	. : : : : : : : . . : : : : : : :	2900 (100.0%)	0 (0.0%)
12	vaccine [factor]	1. 0 2. 1	1192 (41.1%) 1708 (58.9%)	IIIIIIII IIIIIIIIII	2900 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
13	severity [factor]	1. 0 2. 1	2619 (90.3%) 281 (9.7%)	IIIIIIIIIIIIIIIII I	2900 (100.0%)	0 (0.0%)
14	study [factor]	1. A 2. B 3. C	580 (20.0%) 1750 (60.3%) 570 (19.7%)	IIII IIIIIIIIIII III	2900 (100.0%)	0 (0.0%)
15	recovery_time [numeric]	Mean (sd) : 43 (30.5) min < med < max: 3 < 38 < 365 IQR (CV) : 21 (0.7)	144 distinct values	: : : : : : : : : .	2900 (100.0%)	0 (0.0%)

```
skimr::skim_without_charts(train.dat)
```

Table 2: Data summary

Name	train.dat
Number of rows	2900
Number of columns	15
Column type frequency:	
factor	8
numeric	7
Group variables	None

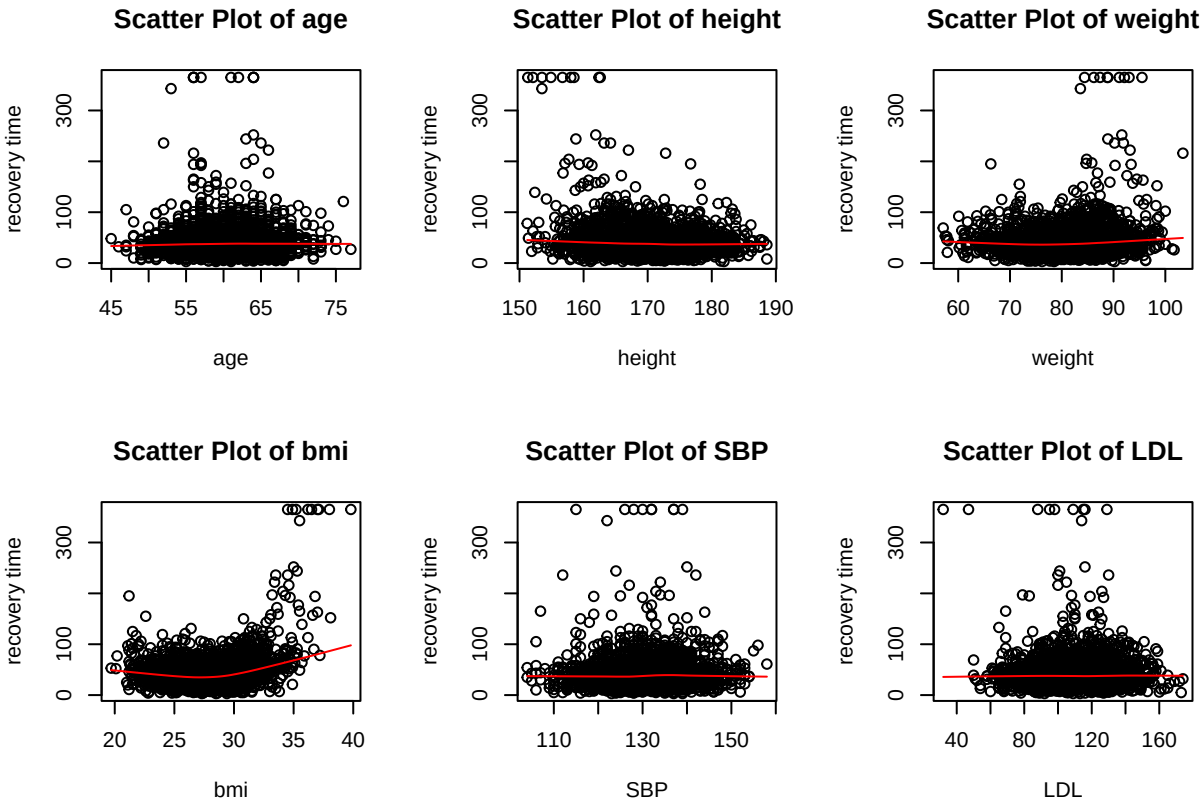
Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gender	0	1	FALSE	2	0: 1468, 1: 1432
race	0	1	FALSE	4	1: 1909, 3: 568, 4: 291, 2: 132
smoking	0	1	FALSE	3	0: 1763, 1: 845, 2: 292
hypertension	0	1	FALSE	2	0: 1514, 1: 1386
diabetes	0	1	FALSE	2	0: 2446, 1: 454
vaccine	0	1	FALSE	2	1: 1708, 0: 1192
severity	0	1	FALSE	2	0: 2619, 1: 281
study	0	1	FALSE	3	B: 1750, A: 580, C: 570

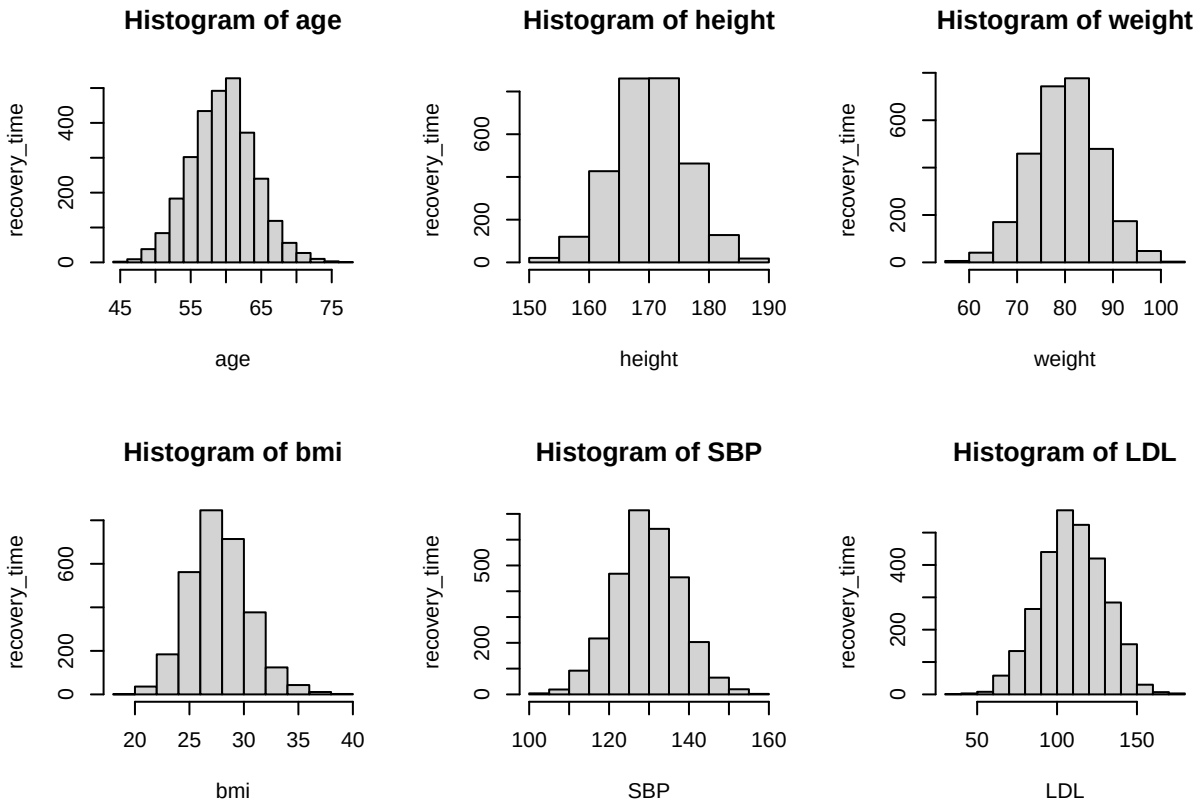
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
age	0	1	60.07	4.51	45.0	57.0	60.00	63.0	77.0
height	0	1	170.17	6.04	151.2	166.1	170.15	174.1	188.6
weight	0	1	80.20	7.00	57.1	75.4	80.30	84.9	103.4
bmi	0	1	27.76	2.73	19.7	25.9	27.70	29.5	39.8
SBP	0	1	130.19	8.08	104.0	125.0	130.00	136.0	158.0
LDL	0	1	110.27	19.87	32.0	97.0	110.00	124.0	174.0
recovery_time	0	1	43.02	30.51	3.0	28.0	38.00	49.0	365.0

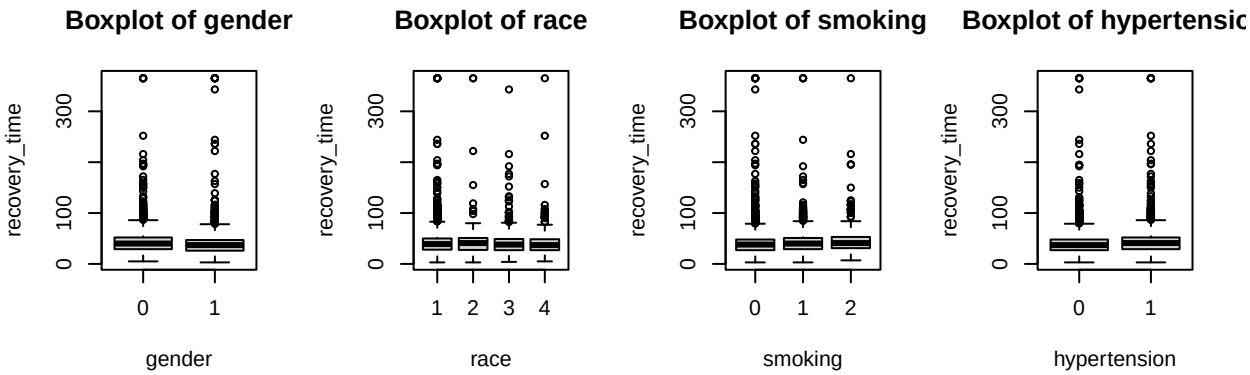
```
########################################################################
## Remember to edit the next chunk if you do any modification here:)
########################################################################

# EDA
cts_var = c("age", "height", "weight", "bmi", "SBP", "LDL")
fct_var = c("gender", "race", "smoking", "hypertension", "diabetes", "vaccine", "severity", "study")

# scatter plot of continuous predictors
par(mfrow=c(2, 3))
for (i in 1:length(cts_var)){
  var = cts_var[i]
  plot(recovery_time~train.dat[,var],
       data = train.dat,
       ylab = "recovery time",
       xlab = var,
       main = str_c("Scatter Plot of ", var))
  lines(stats::lowess(train.dat[,var], train.dat$recovery_time), col = "red", type = "l")
}
```

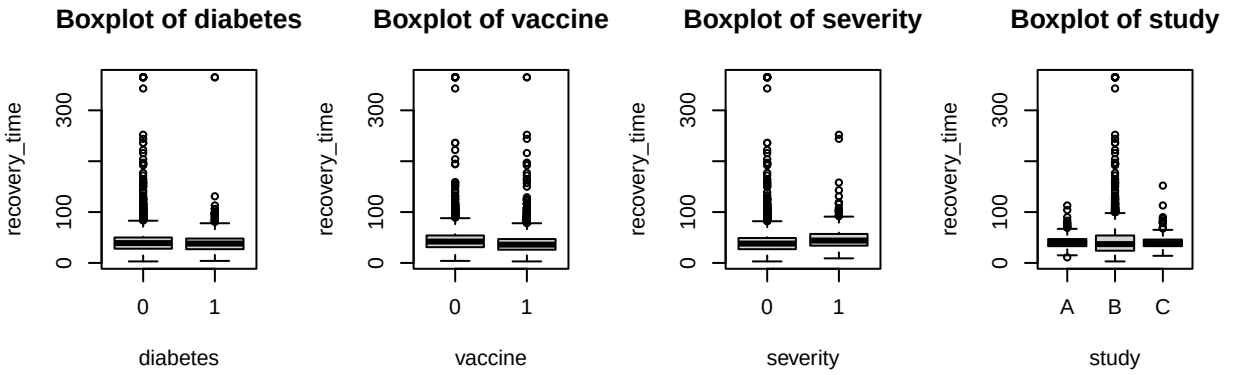



```
for (i in 1:length(cts_var)){
  var = cts_var[i]
  hist(train.dat[,var],
       ylab = "recovery_time",
       xlab = var,
       main = str_c("Histogram of ", var))
}
```

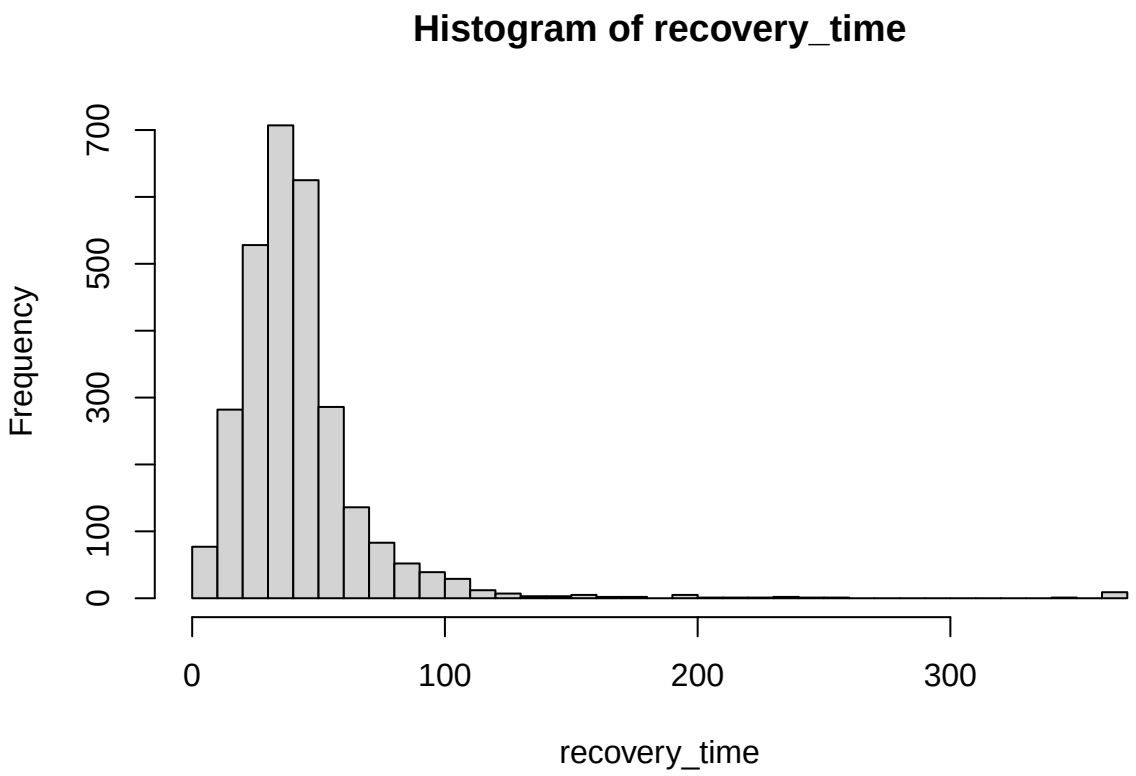



```
# boxplot of categorical predictors
par(mfrow=c(2, 4))
for (i in 1:length(fct_var)){
  var = fct_var[i]
  plot(recovery_time~train.dat[,var],
       data = train.dat,
       ylab = "recovery_time",
       xlab = var,
       main = str_c("Boxplot of ", var))
}
```



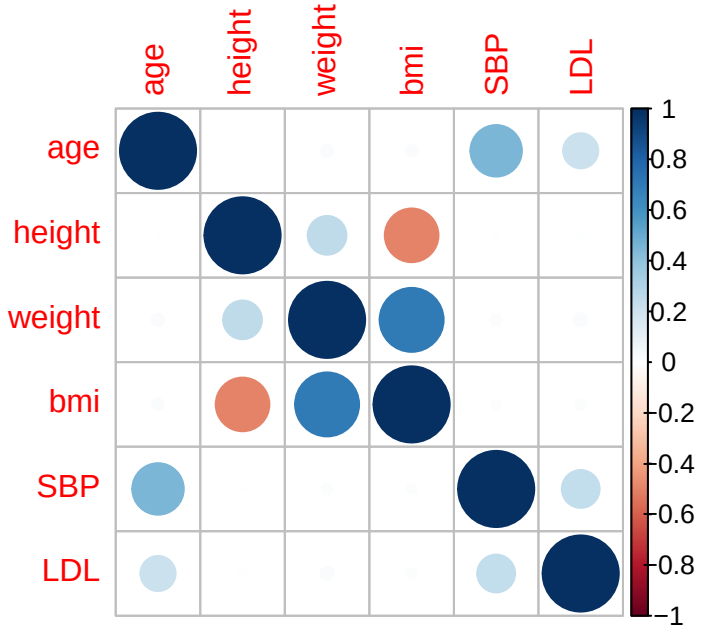



```
# histogram of response
par(mfrow=c(1, 1))
hist(train.dat$recovery_time,
     breaks = 50,
     main = "Histogram of recovery_time",
     xlab = "recovery_time")
```




```
# correlation
par(mfrow=c(1, 1))
corrplot(cor(train.dat[,cts_var]), method = "circle", type = "full",
         title = "Correlation plot of continuous variables",
         mar = c(2, 2, 4, 2))
```

Correlation plot of continuous variables



3.2 Model Training

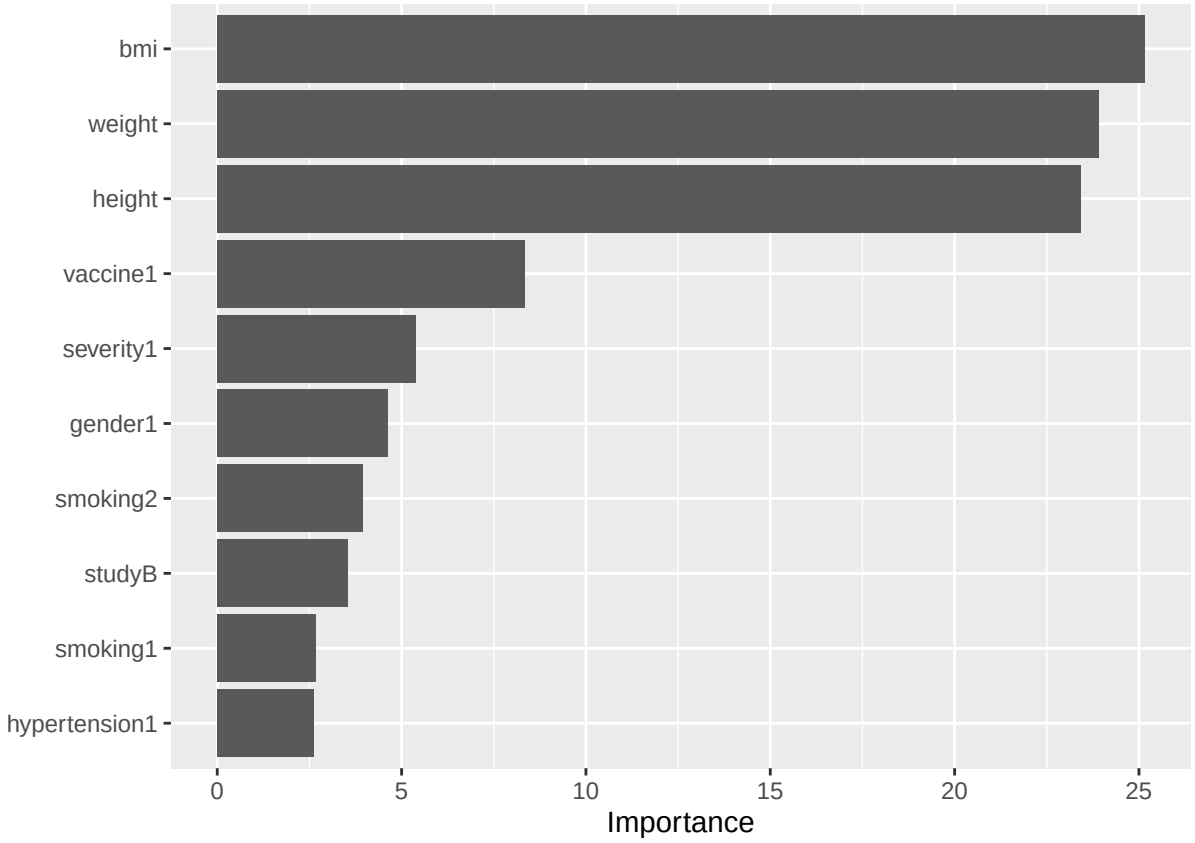
```
ctrl1 <- trainControl(method = "cv", number = 5)
```

3.2.1 Linear Model

```
set.seed(2023)

lm.fit <- train(train.x, train.y,
                method = "lm",
                trControl = ctrl1)

coef(lm.fit$finalModel)
```

```
##   (Intercept)           age       gender1         race2         race3
## -3.190120e+03  1.163953e-01 -4.443893e+00  2.189010e+00 -6.599719e-01
##         race4      smoking1      smoking2        height        weight
## -1.156806e+00  2.905693e+00  6.427376e+00  1.866280e+01 -2.014323e+01
##           bmi hypertension1     diabetes1           SBP           LDL
##  6.056969e+01  4.165589e+00 -1.152370e+00 -7.863399e-02 -4.215262e-02
##      vaccine1     severity1        studyB        studyC
## -8.133542e+00  8.747096e+00  4.368587e+00 -6.869681e-01
```

```
vip(lm.fit$finalModel)
```



3.2.2 LASSO

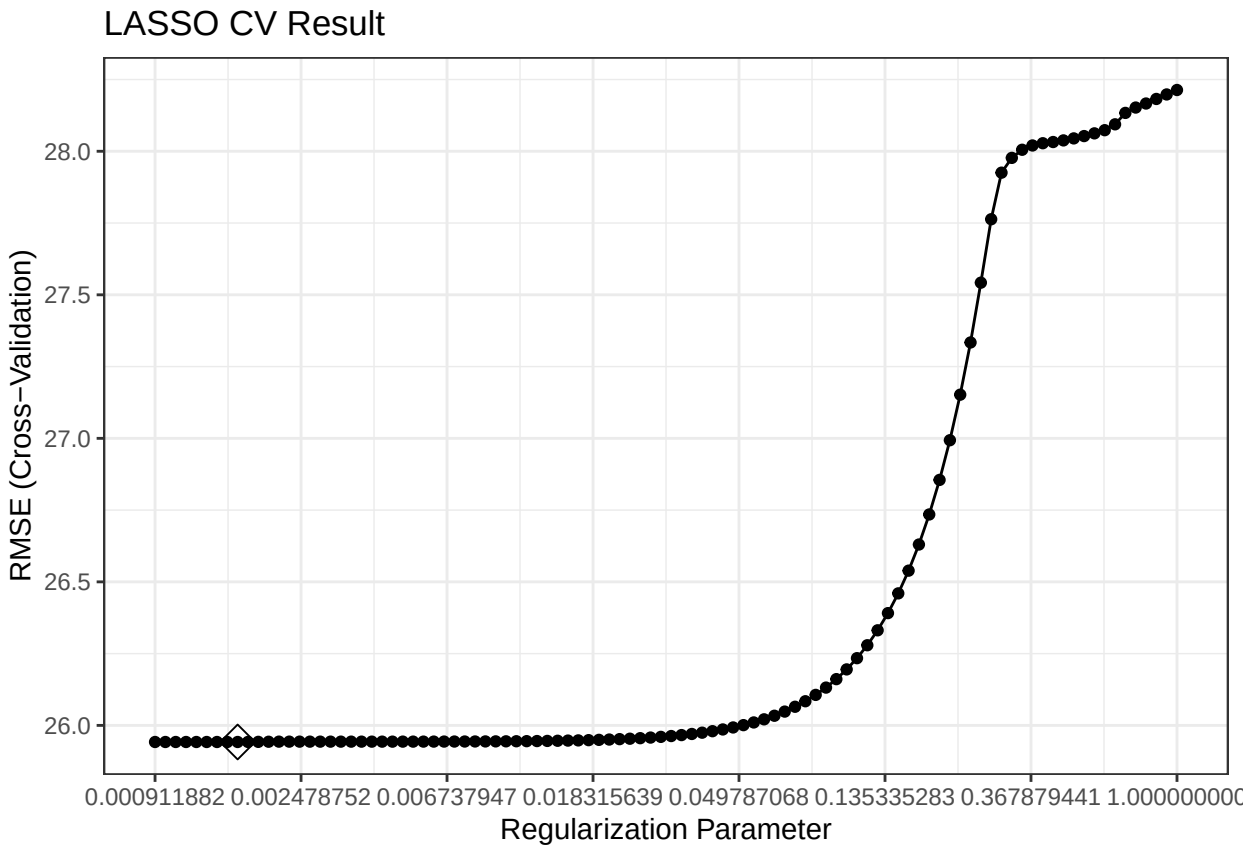
```
set.seed(2023)
lasso.fit <- train(train.x, train.y,
                   method = "glmnet",
                   tuneGrid = expand.grid(
                     alpha = 1,
                     lambda = exp(seq(0, -7, length=100))),
                   trControl = ctrl1)

lasso.fit$bestTune
```

```
##   alpha      lambda
## 9     1 0.001605462
```

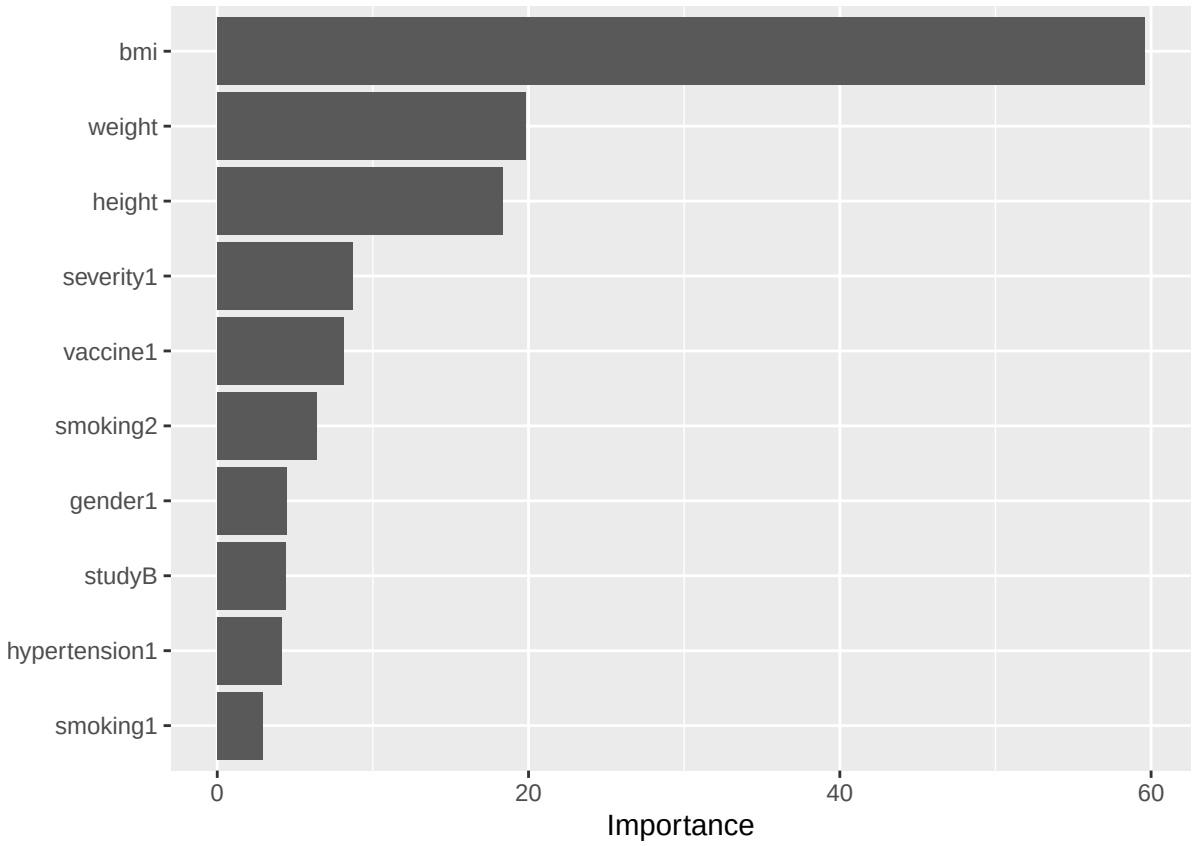
```
coef(lasso.fit$finalModel, s = lasso.fit$bestTune$lambda)
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)   -3.134172e+03
## age            1.153955e-01
## gender1       -4.441866e+00
## race2          2.191861e+00
## race3         -6.681255e-01
## race4         -1.149670e+00
## smoking1       2.901232e+00
## smoking2       6.400802e+00
## height         1.833161e+01
## weight        -1.979266e+01
## bmi            5.956877e+01
## hypertension1  4.150461e+00
## diabetes1     -1.160249e+00
## SBP           -7.746419e-02
## LDL           -4.212203e-02
## vaccine1      -8.147730e+00
## severity1      8.730928e+00
## studyB         4.369356e+00
## studyC        -6.781352e-01
```

```
ggplot(lasso.fit, highlight = TRUE) +
  labs(title="LASSO CV Result") +
  scale_x_continuous(trans='log',n.breaks = 10) +
  theme_bw()
```




```
ggsave("./figure/lasso_cv.jpeg", dpi = 500)

vip(lasso.fit$finalModel)
```



3.2.3 Ridge

```
set.seed(2023)
ridge.fit <- train(train.x, train.y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 0,
                                          lambda = exp(seq(1, -5, length=100))),
                   trControl = ctrl1)

ridge.fit$bestTune
```

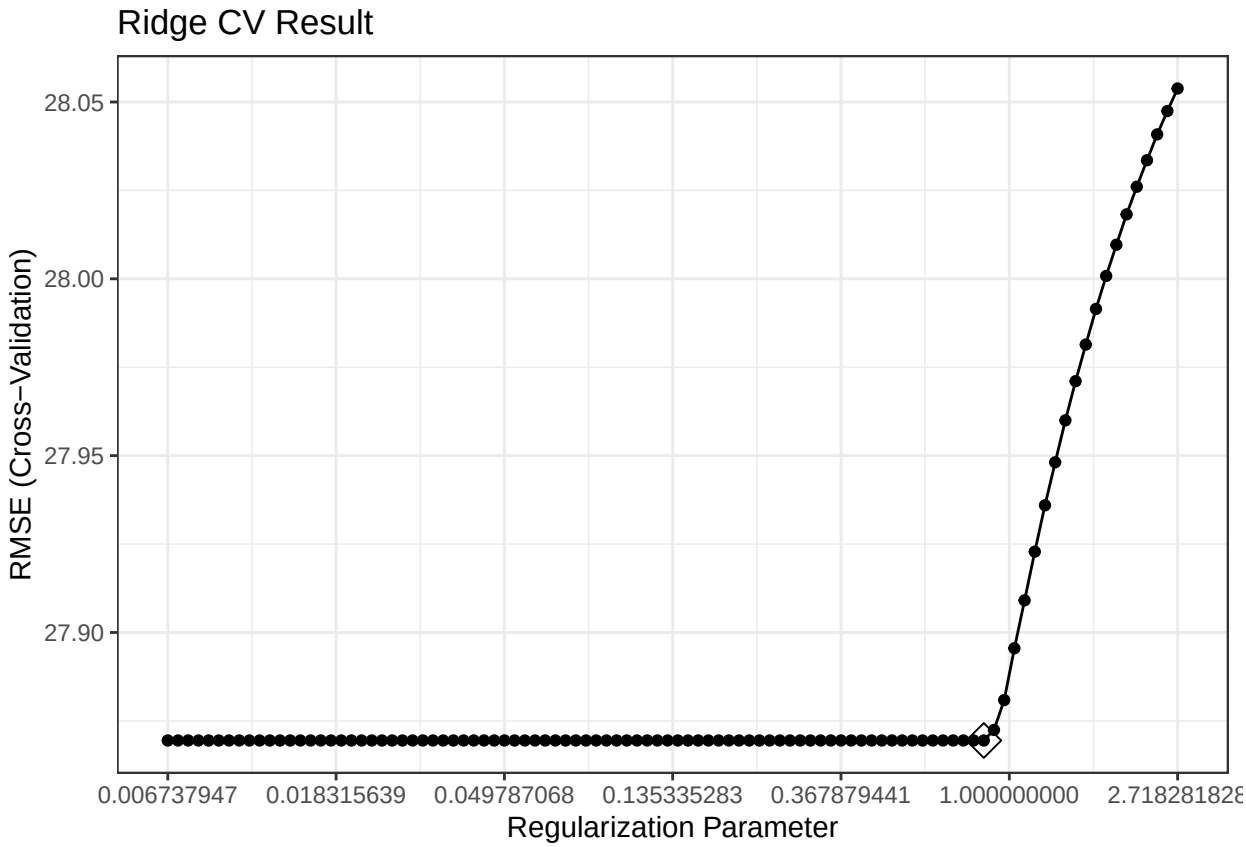
```
##    alpha    lambda
## 81     0 0.8594049
```

```
coef(ridge.fit$finalModel, s = ridge.fit$bestTune$lambda)
```

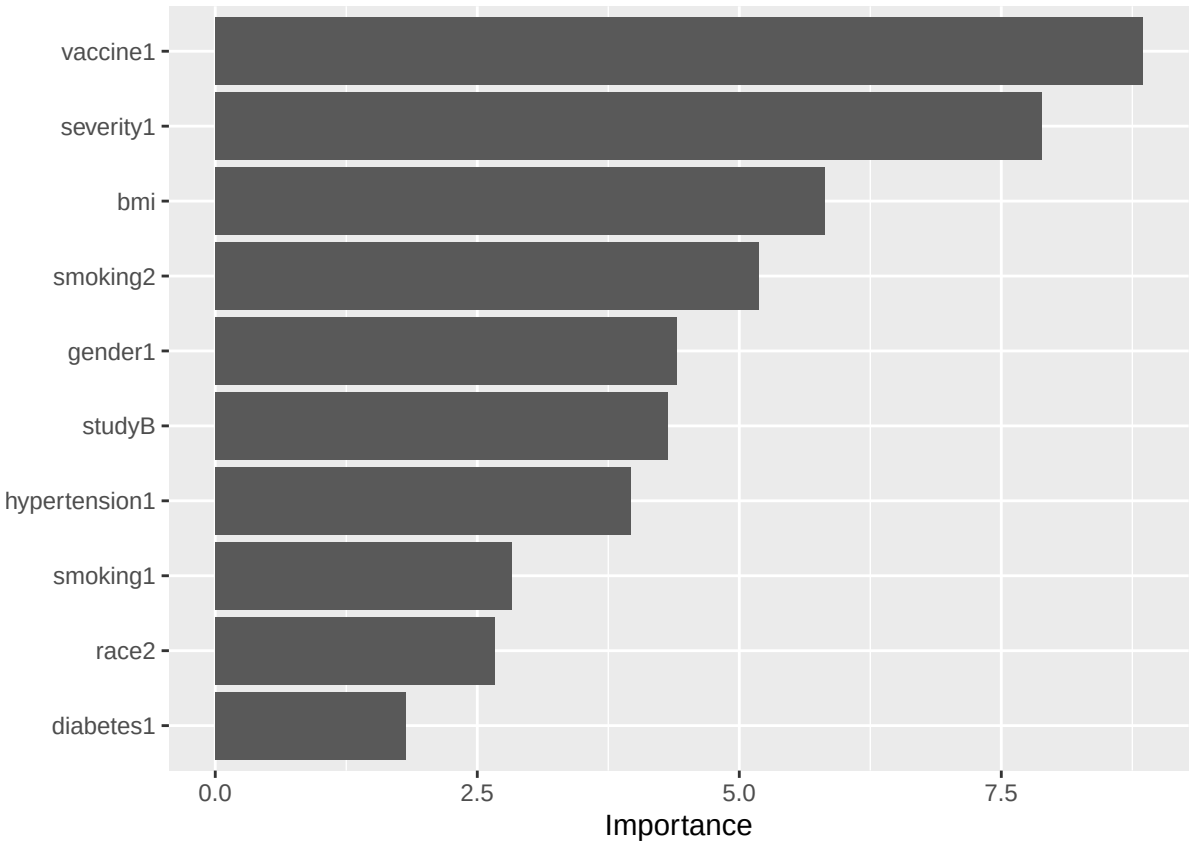
```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)   -131.33806374
## age              0.09731228
## gender1         -4.40320528
## race2            2.66527141
## race3           -1.32710400
## race4           -1.12570977
## smoking1         2.82624366
## smoking2         5.18400128
## height           0.60404463
```

```
## weight          -1.01341715
## bmi              5.81922510
## hypertension1    3.96367066
## diabetes1       -1.81677375
## SBP             -0.06303616
## LDL             -0.04440780
## vaccine1        -8.84608080
## severity1        7.88676978
## studyB           4.32156225
## studyC          -0.51357417
```

```
ggplot(ridge.fit,highlight = TRUE) +
  scale_x_continuous(trans='log', n.breaks = 6) +
  labs(title="Ridge CV Result") +
  theme_bw()
```




```
ggsave("./figure/ridge_cv.jpeg", dpi = 500)

vip(ridge.fit$finalModel)
```



3.2.4 Elastic Net

```
set.seed(2023)
enet.fit <- train(train.x, train.y,
                  method = "glmnet",
                  tuneGrid = expand.grid(
                    alpha = seq(0, 1, length = 21),
                    lambda = exp(seq(0, -8, length = 100))),
                  trControl = ctrl1)

enet.fit$bestTune
```

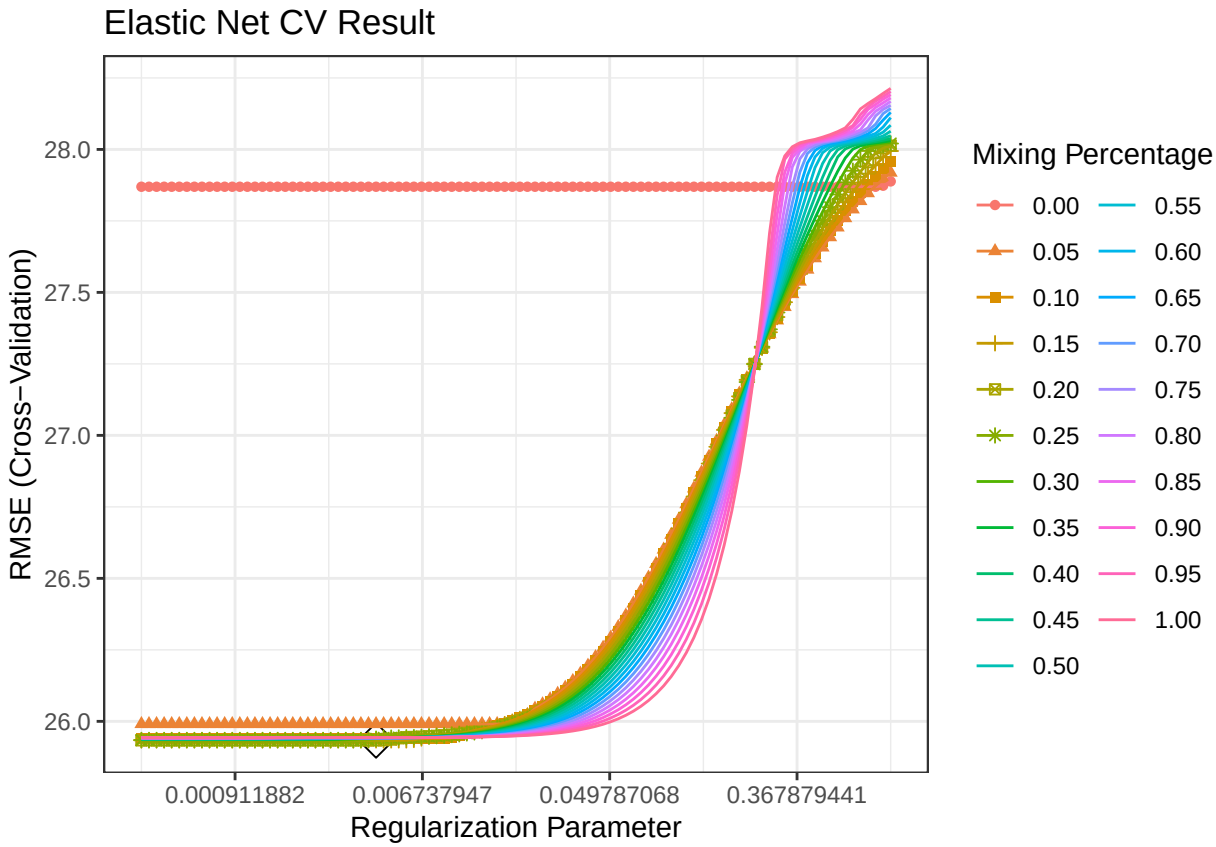
```
##     alpha      lambda
## 432   0.2 0.004107464
```

```
coef(enet.fit$finalModel, enet.fit$bestTune$lambda)
```

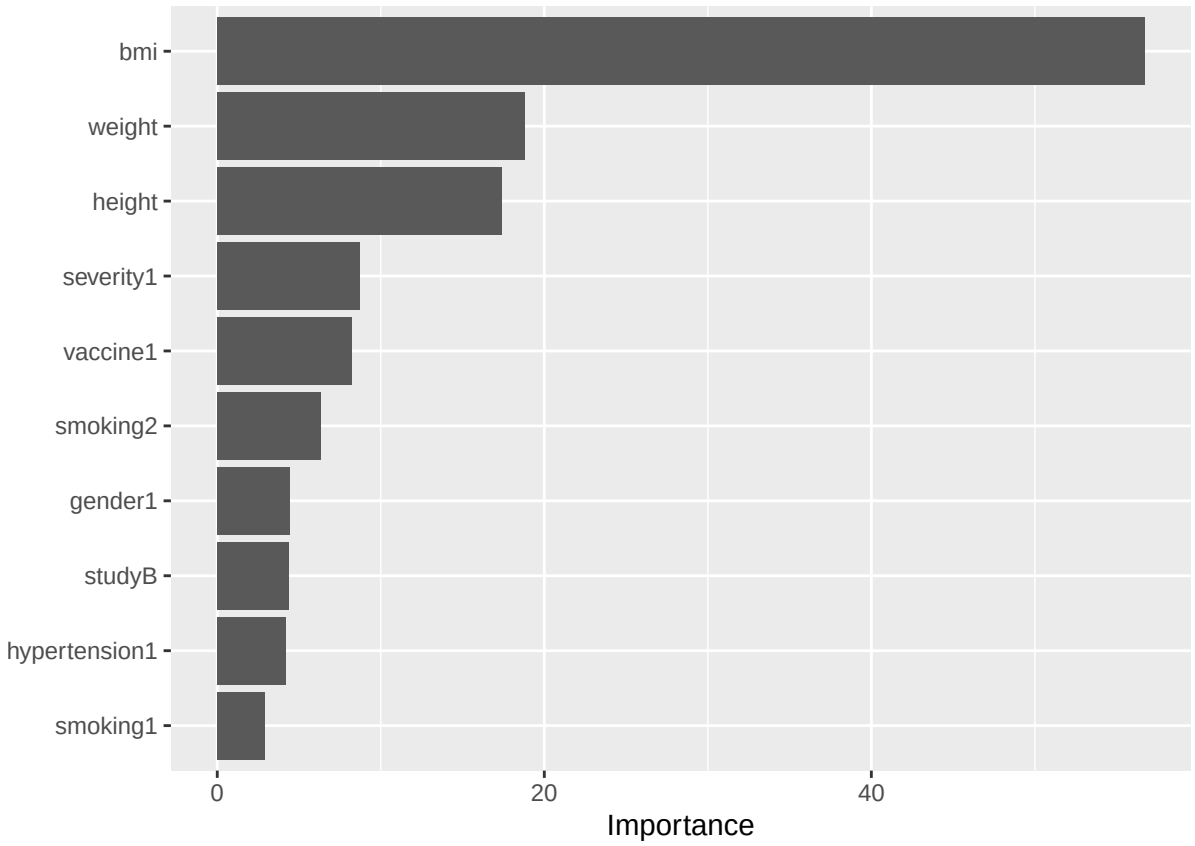
```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                               s1
## (Intercept)   -2.973365e+03
## age            1.150907e-01
## gender1       -4.447763e+00
## race2          2.221991e+00
## race3         -7.070056e-01
## race4         -1.153498e+00
## smoking1       2.906092e+00
## smoking2       6.349675e+00
```

```
## height          1.738293e+01
## weight         -1.878818e+01
## bmi             5.669617e+01
## hypertension1   4.168257e+00
## diabetes1      -1.199482e+00
## SBP            -7.836707e-02
## LDL            -4.236166e-02
## vaccine1       -8.201080e+00
## severity1       8.701657e+00
## studyB          4.377894e+00
## studyC         -6.636602e-01
```

```
ggplot(enet.fit, highlight = TRUE) +
  scale_x_continuous(trans='log', n.breaks = 6) +
  labs(title ="Elastic Net CV Result") +
  theme_bw()
```

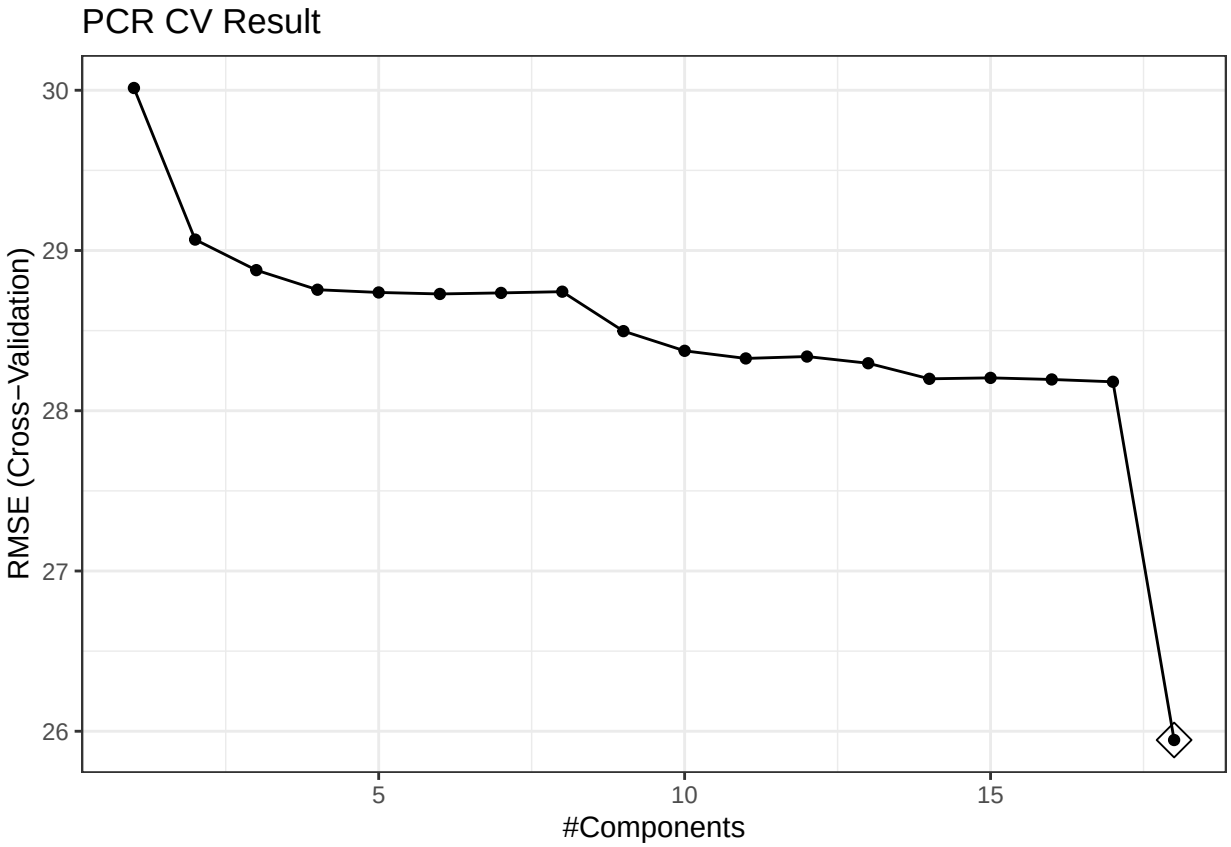



```
ggsave("./figure/enet_cv.jpeg", dpi = 500)

vip(enet.fit$finalModel)
```



3.2.5 Principal components regression (PCR)

```
set.seed(2023)
pcr.fit <- train(train.x,
                 train.y,
                 method = "pcr",
                 tuneGrid  = data.frame(ncomp = 1:ncol(train.x)),
                 trControl = ctrl1,
                 preProcess = c("center", "scale"))
ggplot(pcr.fit, highlight = TRUE) +
  labs(title  ="PCR CV Result") +
  theme_bw()
```




```
ggsave("./figure/pcr_cv.jpeg", dpi = 500)

pcr.fit$bestTune
```

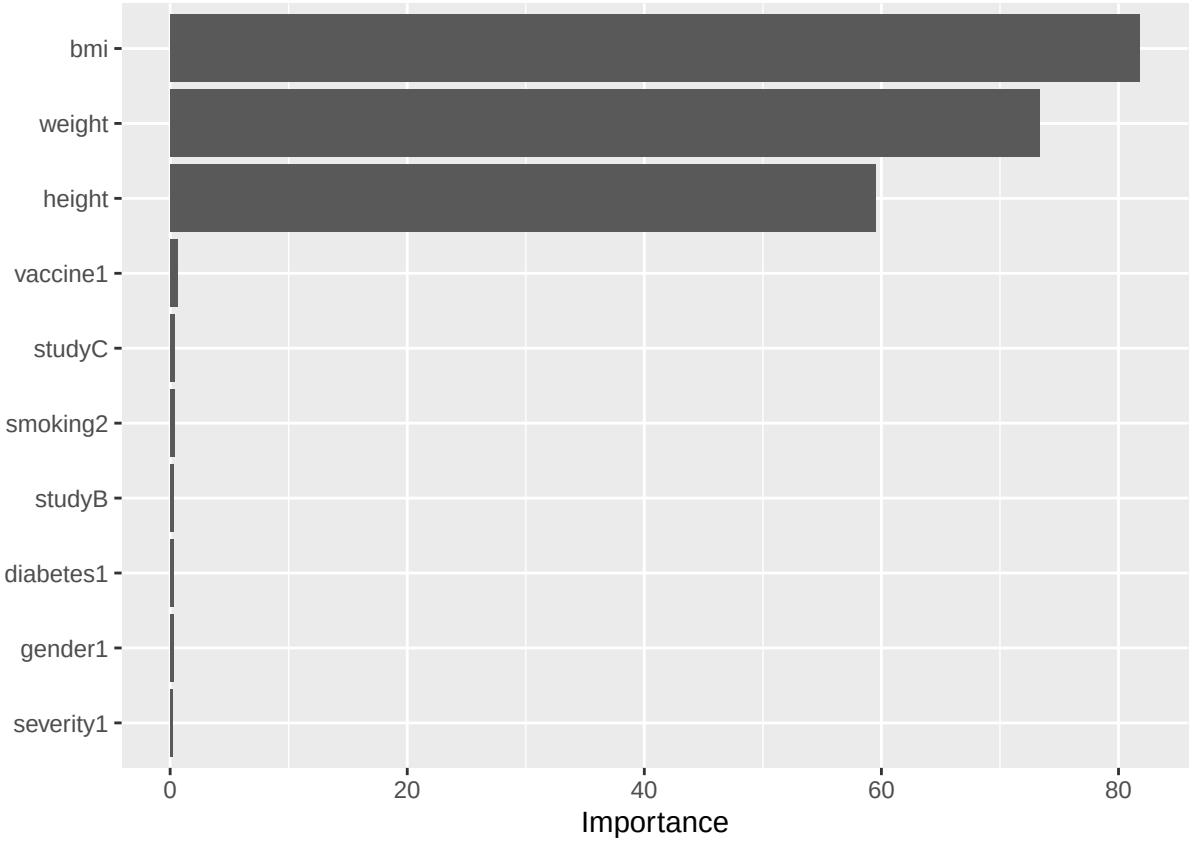
```
##    ncomp
## 18    18
```

```
coef(pcr.fit$finalModel)
```

```
## , , 18 comps
##
##                    .outcome
## age               0.5252538
## gender1          -2.2221586
## race2             0.4563464
## race3            -0.2619635
## race4            -0.3476329
## smoking1          1.3205684
## smoking2          1.9344423
## height          112.6936931
## weight         -141.0001175
## bmi             165.1518985
## hypertension1     2.0811234
## diabetes1        -0.4188178
## SBP              -0.6356938
## LDL              -0.8376686
## vaccine1         -4.0025673
## severity1         2.5879846
```

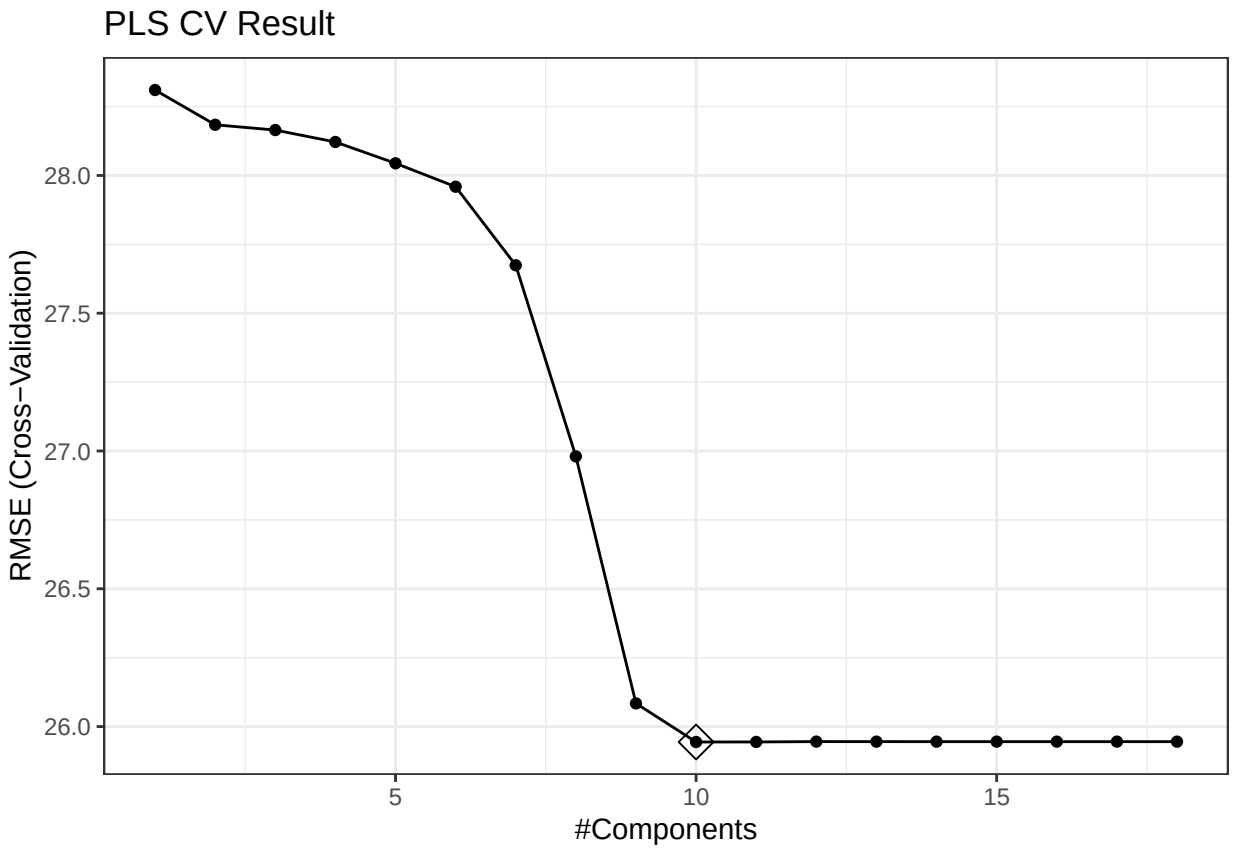
```
## studyB            2.1374000
## studyC           -0.2730416
```

```
vip(pcr.fit$finalModel)
```



3.2.6 Partial Least Squares (PLS)

```
set.seed(2023)
pls.fit <- train(train.x,
                 train.y,
                 method = "pls",
                 tuneGrid = data.frame(ncomp = 1:ncol(train.x)),
                 trControl = ctrl1,
                 preProcess = c("center", "scale"))
ggplot(pls.fit, highlight = TRUE) +
  labs(title  ="PLS CV Result") +
  theme_bw()
```




```
ggsave("./figure/pls_cv.jpeg", dpi = 500)

pls.fit$bestTune
```

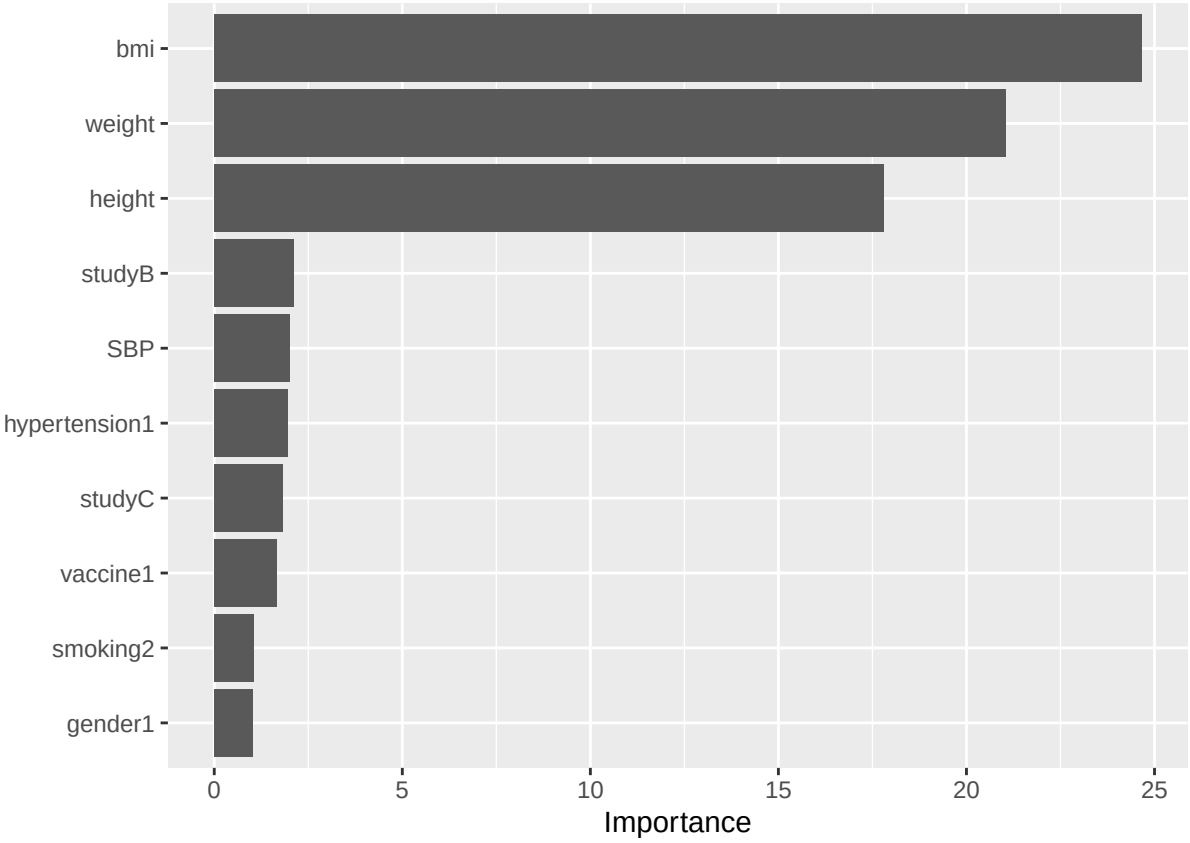
```
##    ncomp
## 10    10
```

```
coef(pls.fit$finalModel)
```

```
## , , 10 comps
##
##                    .outcome
## age               0.2557655
## gender1          -2.2288535
## race2             0.4448161
## race3            -0.1161982
## race4            -0.4146309
## smoking1          1.3626057
## smoking2          1.8793373
## height          112.5738962
## weight         -140.9034395
## bmi             165.0235495
## hypertension1     2.1927025
## diabetes1        -0.4588377
## SBP              -0.6092501
## LDL              -0.7129796
## vaccine1         -4.0284909
## severity1         2.5664367
```

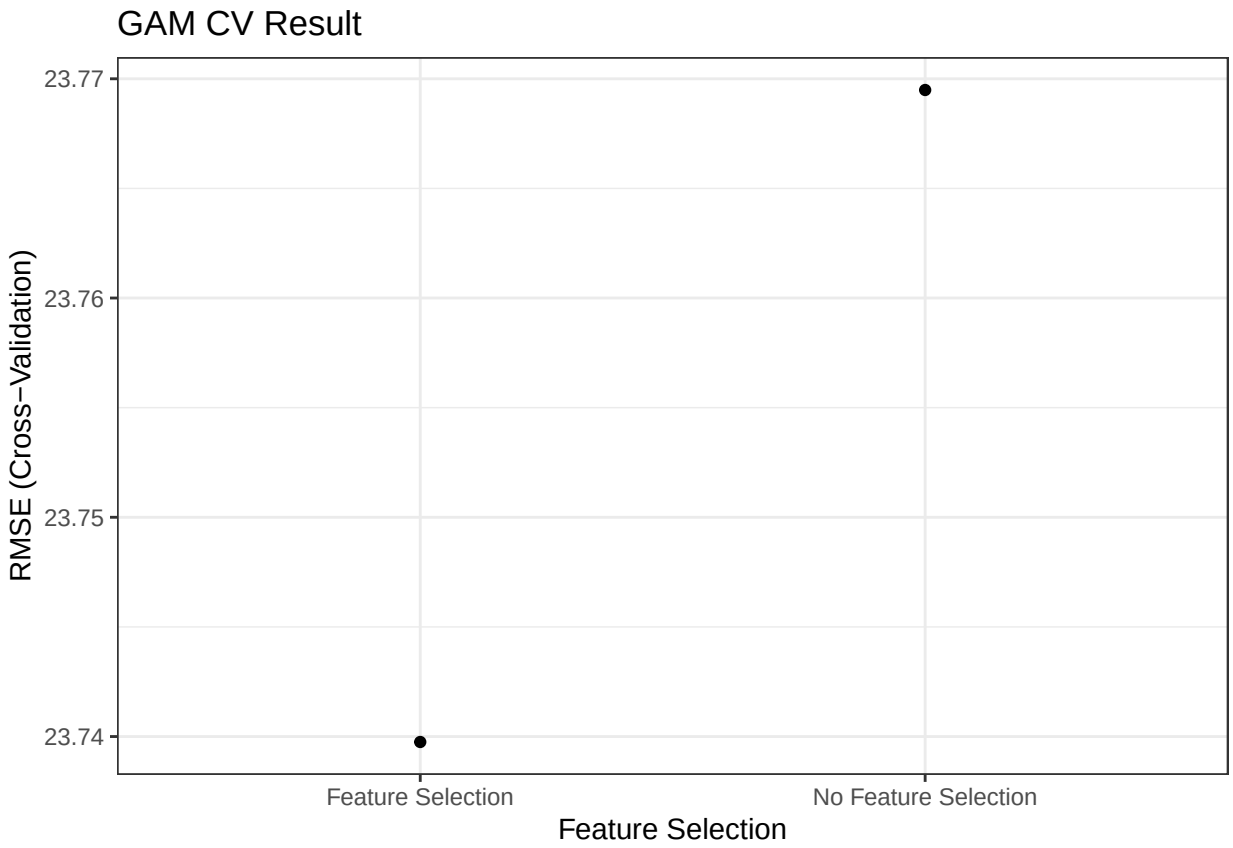
```
## studyB            2.1234056
## studyC           -0.2961257
```

```
vip(pls.fit$finalModel)
```



3.2.7 Generalized Additive Model (GAM)

```
set.seed(2023)
gam.fit <- train(train.x,
                 train.y,
                 method = "gam",
                 tuneGrid = data.frame(select = c(TRUE, FALSE),
                                       method = "GCV.Cp"),
                 trControl = ctrl1)


ggplot(gam.fit) +
  labs(title = "GAM CV Result") +
  theme_bw()
```



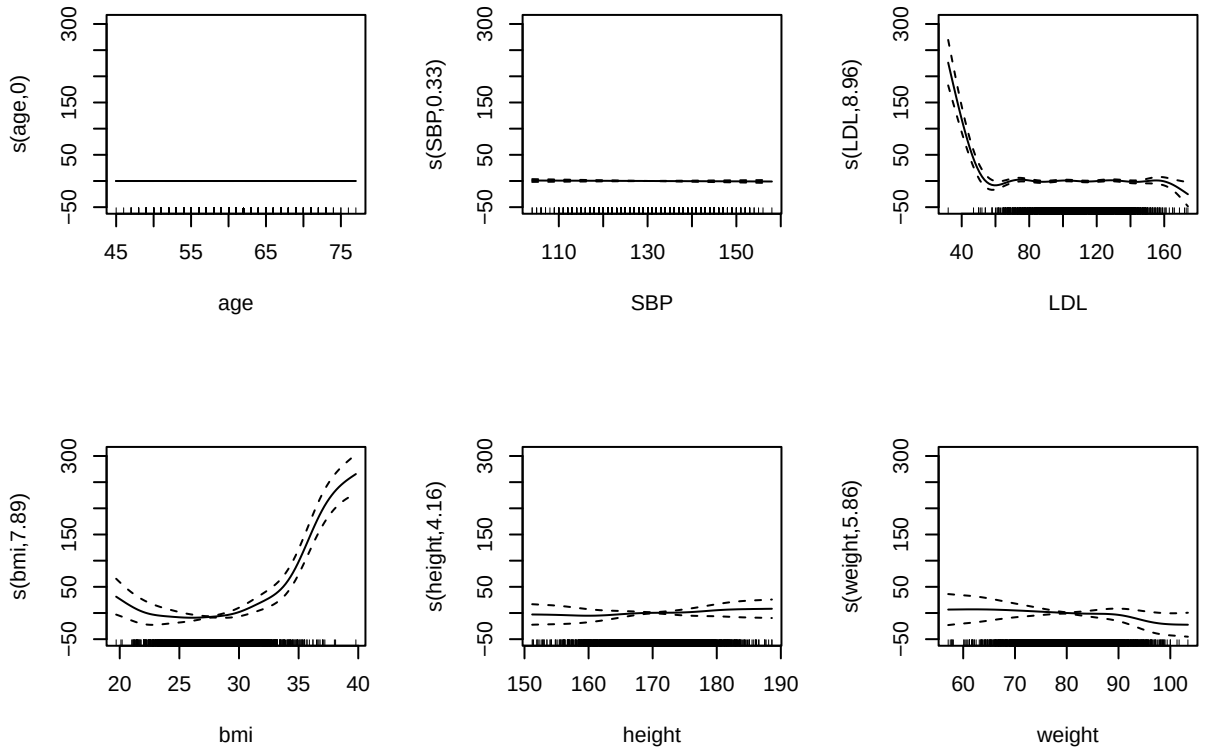

```
ggsave("./figure/gam_cv.jpeg", dpi = 500)

gam.fit$bestTune
```

```
##   select method
## 2   TRUE GCV.Cp
```

```
# coef(gam.fit$finalModel)
gam.fit$finalModel
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender1 + race3 + race4 + smoking1 + smoking2 + hypertension1 +
##     diabetes1 + vaccine1 + severity1 + studyB + studyC + s(age) +
##     s(SBP) + s(LDL) + s(bmi) + s(height) + s(weight)
##
## Estimated degrees of freedom:
## 0.000 0.329 8.959 7.893 4.163 5.856  total = 39.2
##
## GCV score: 524.051
```

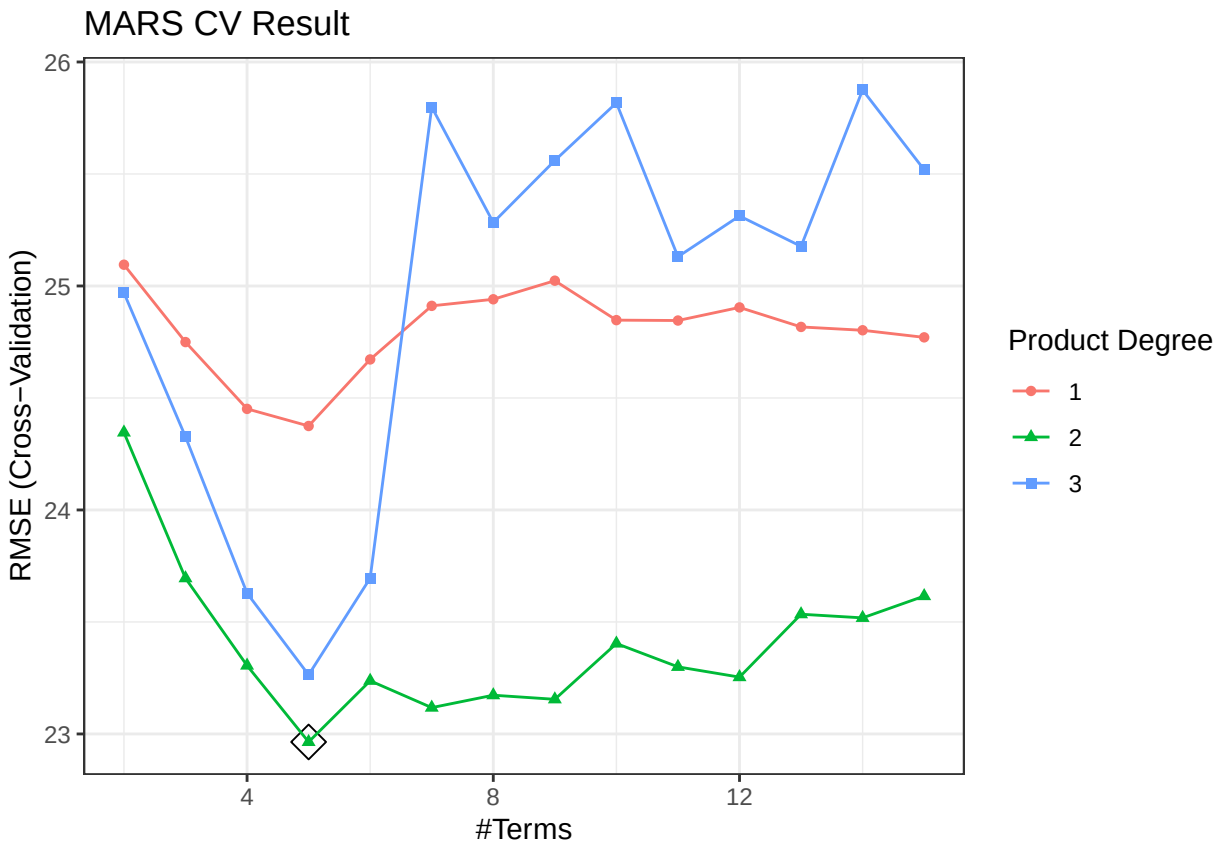
```
par(mfrow=c(2, 3))
plot(gam.fit$finalModel)
```




```
par(mfrow=c(1, 1))
```

3.2.8 Multivariate Adaptive Regression Splines (MARS)

```
mars.grid <- expand.grid(degree = 1:3,
                         nprune = 2:15)
set.seed(2023)
mars.fit <- train(train.x,
                  train.y,
                  method = "earth",
                  tuneGrid = mars.grid,
                  trControl = ctrl1)

ggplot(mars.fit, highlight = TRUE)+
  labs(title  ="MARS CV Result") +
  theme_bw()
```




```
ggsave("./figure/mars_cv.jpeg", dpi = 500)

mars.fit$bestTune
```

```
##    nprune degree
## 18      5      2
```

```
coef(mars.fit$finalModel)
```

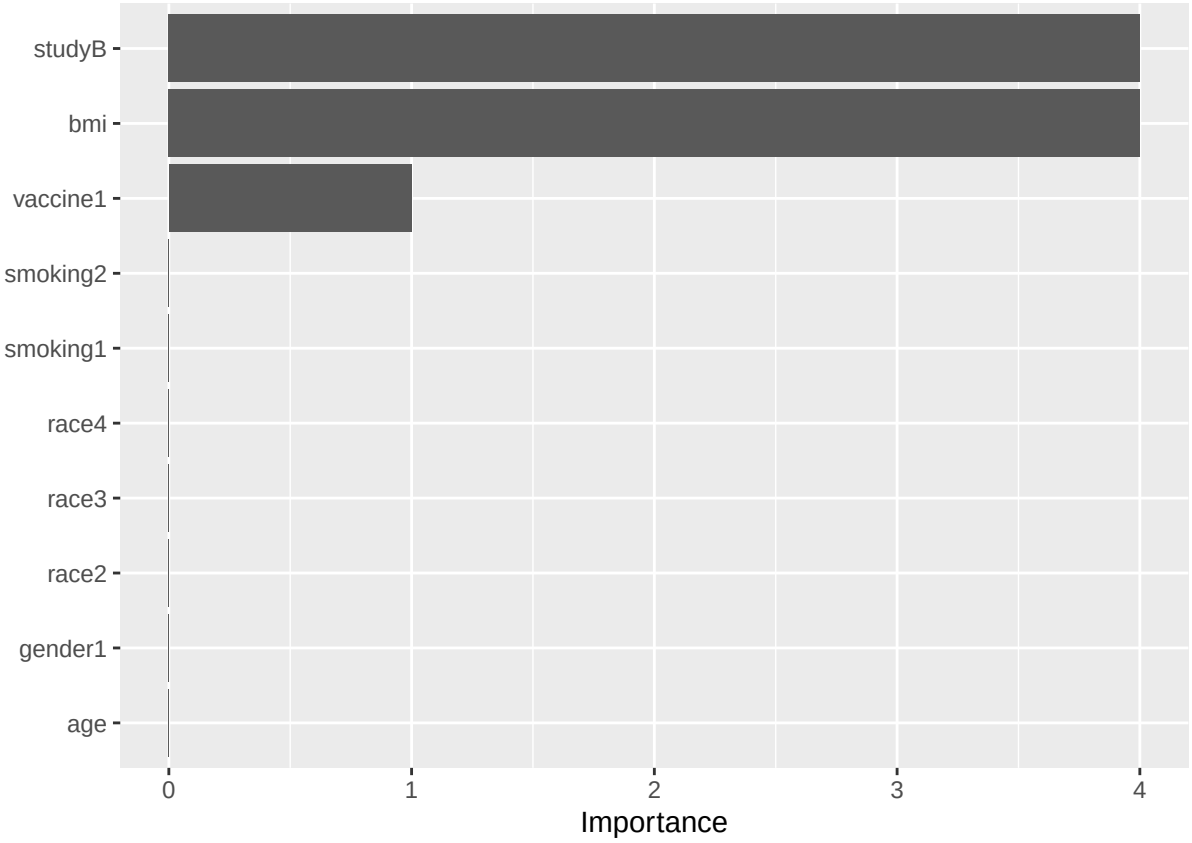
```
##          (Intercept)          h(31.7-bmi) h(bmi-31.7) * studyB
##            19.366730             3.705371            34.383832
##          h(bmi-26.8)             vaccine1
##             6.695655            -7.788338
```

```
summary(mars.fit$finalModel)
```

```
## Call: earth(x=matrix[2900,18], y=c(40,34,31,50,3...), keepxy=TRUE, degree=2,
##             nprune=5)
##
##                      coefficients
## (Intercept)             19.366730
## vaccine1                -7.788338
## h(bmi-26.8)              6.695655
## h(31.7-bmi)              3.705371
## h(bmi-31.7) * studyB    34.383832
##
## Selected 5 of 25 terms, and 3 of 18 predictors (nprune=5)
## Termination condition: Reached nk 37
## Importance: bmi, studyB, vaccine1, age-unused, gender1-unused, ...
```

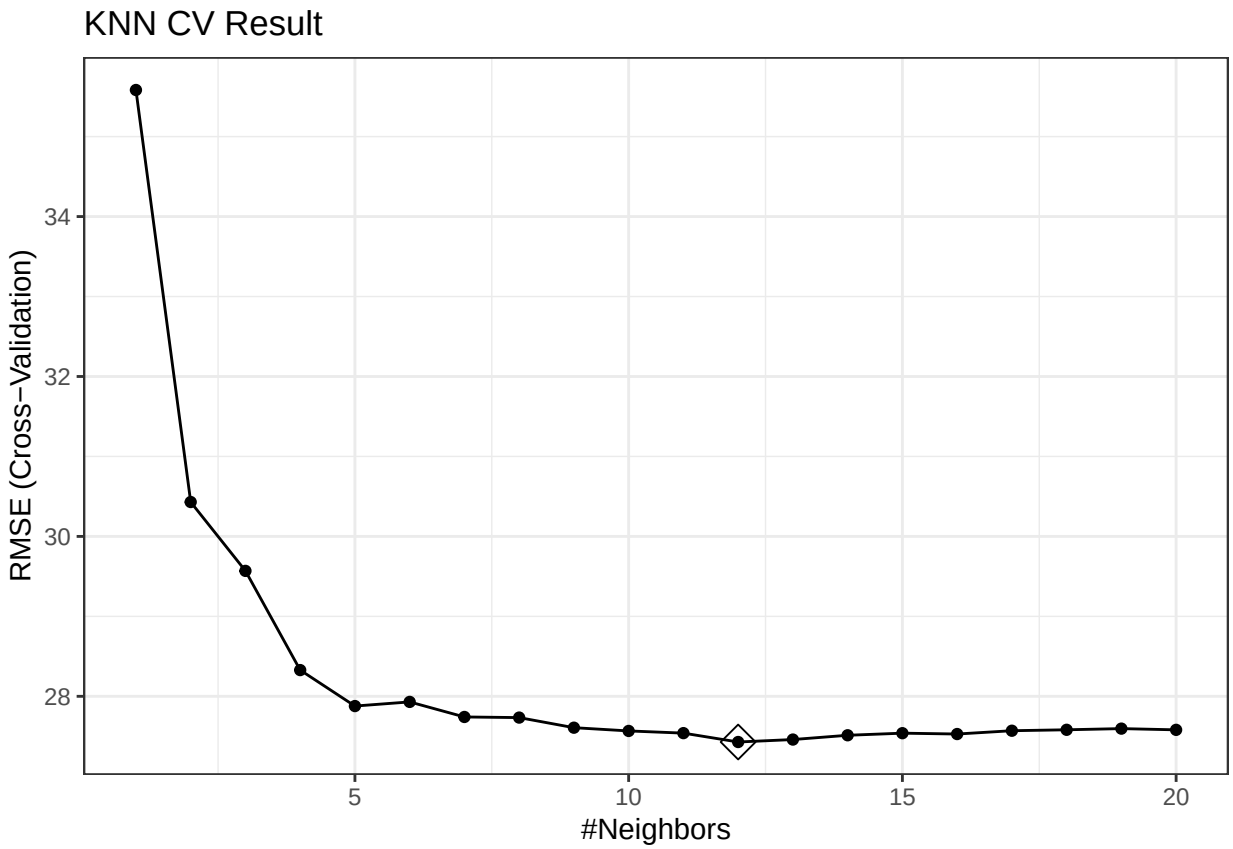
```
## Number of terms at each degree of interaction: 1 3 1
## GCV 491.1694    RSS 1413606    GRSq 0.4723714    RSq 0.4760052
```

```
vip(mars.fit$finalModel)
```



3.2.9 K-Nearest Neighbour (KNN)

```
set.seed(2023)
knn.fit <- train(train.x,
                 train.y,
                 tuneGrid  = data.frame(k = 1:20),
                 method = "knn",
                 trControl = ctrl1)

ggplot(knn.fit, highlight = TRUE) +
  labs(title  ="KNN CV Result") +
  theme_bw()
```




```
ggsave("./figure/knn_cv.jpeg", dpi = 500)

knn.fit$bestTune
```

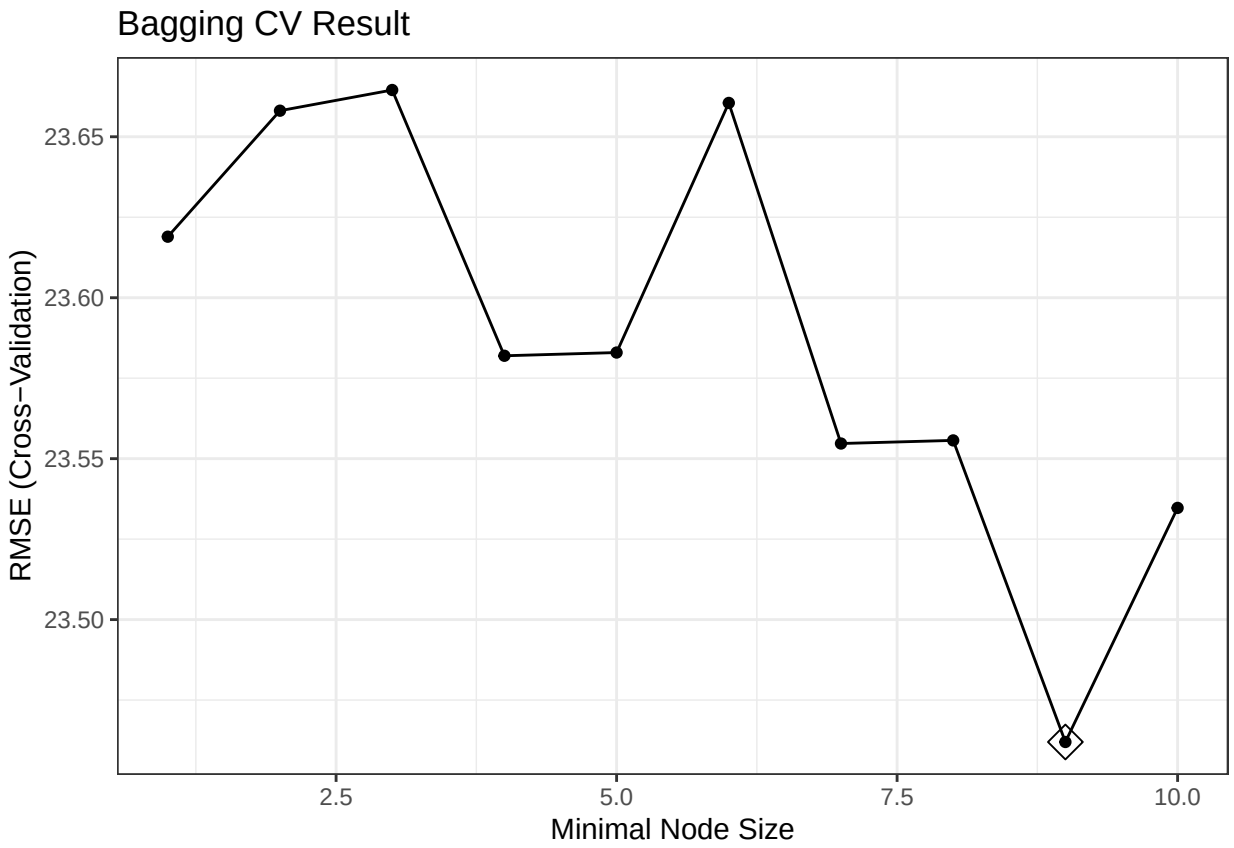
```
##     k
## 12 12
```

3.2.10 Bagging

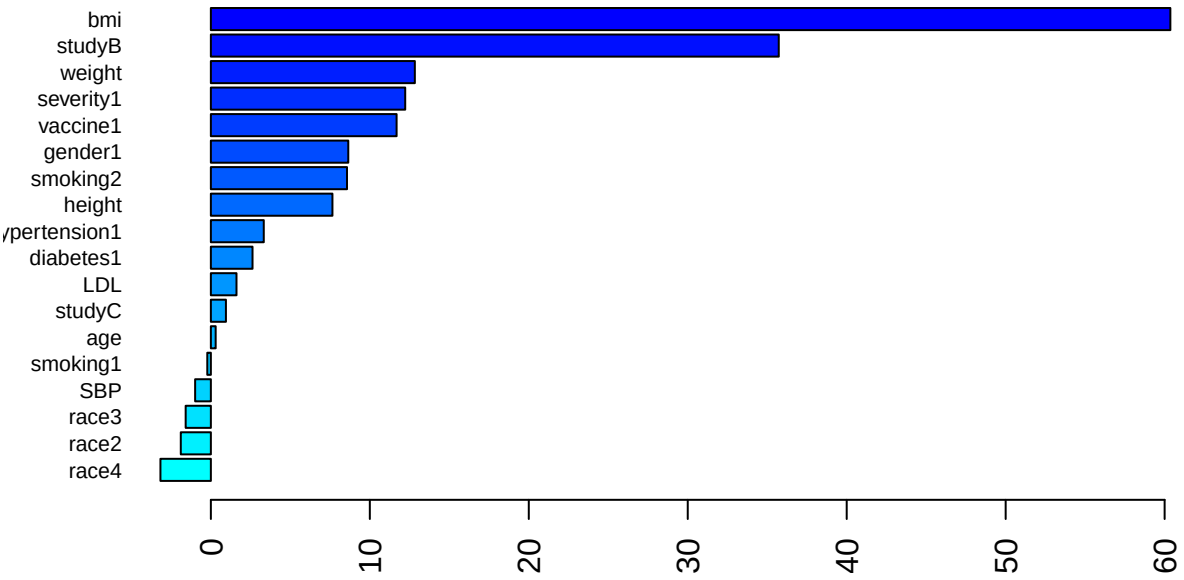
```
bag.grid <- expand.grid(mtry = ncol(train.x),
                        splitrule = "variance",
                        min.node.size = 1:10)
set.seed(2023)
bag.fit <- train(train.x,
                 train.y,
                 method = "ranger",
                 tuneGrid = bag.grid,
                 trControl = ctrl1)

bag.fit$bestTune
```

```
##   mtry splitrule min.node.size
## 9   18  variance             9
```

```
ggplot(bag.fit, highlight = TRUE) +
  labs(title = "Bagging CV Result") +
  theme_bw()
```




```
ggsave("./figure/bagging_cv.jpeg", dpi = 500)

bag.final.per <- ranger(recovery_time ~ .,
                        data = train.dat.matrix,
                        mtry = ncol(train.x),
                       splitrule = "variance",
                       min.node.size = bag.fit$bestTune[[3]],
                       importance = "permutation",
                       scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(bag.final.per),
             decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(ncol(train.x)))
```



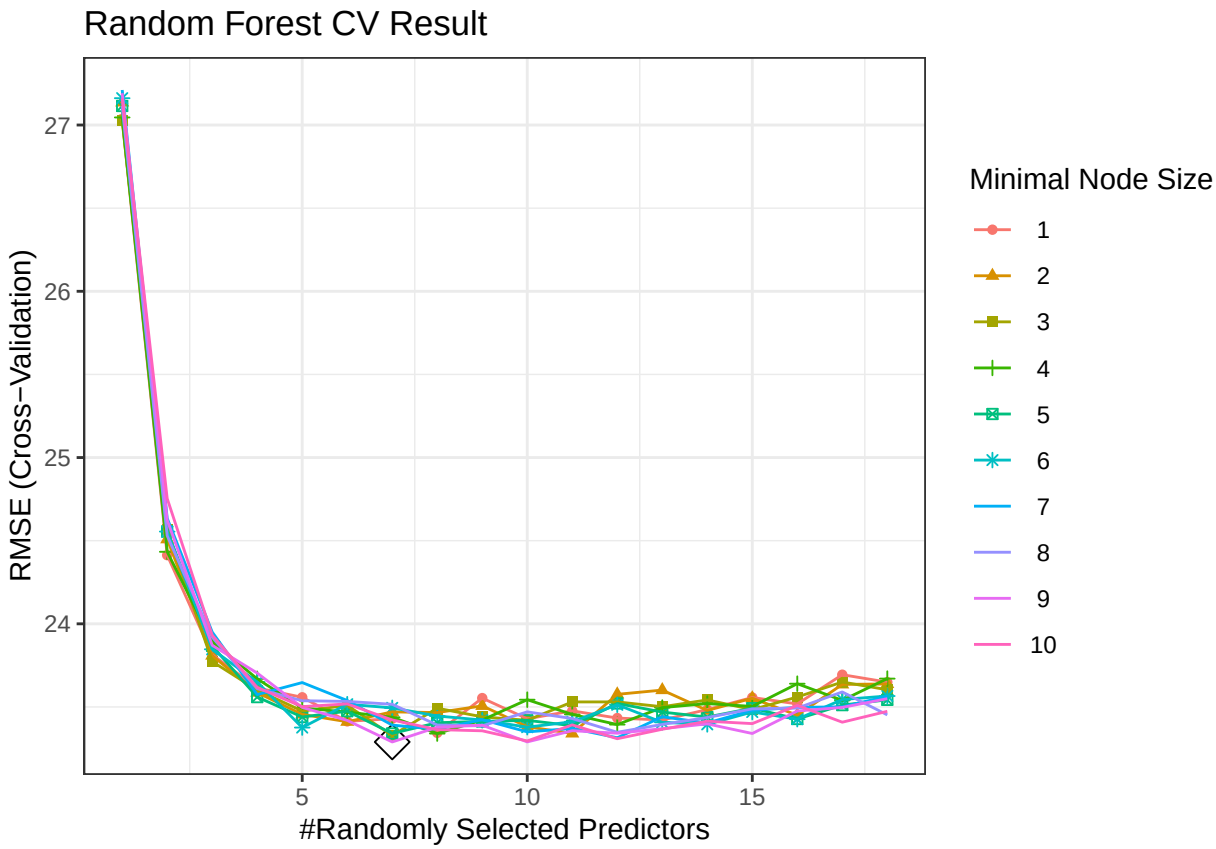

```
# p1 <- pdp::partial(
#   bag.fit,
#   pred.var = "Lot_Area",
#   grid.resolution = 20
#   ) %>%
#   autoplot()
# p2 <- pdp::partial(
#   bag.fit,
#   pred.var = "Lot_Frontage",
#   grid.resolution = 20
#   ) %>%
#   autoplot()
# gridExtra::grid.arrange(p1, p2, nrow = 1)
```

3.2.11 Random Forest

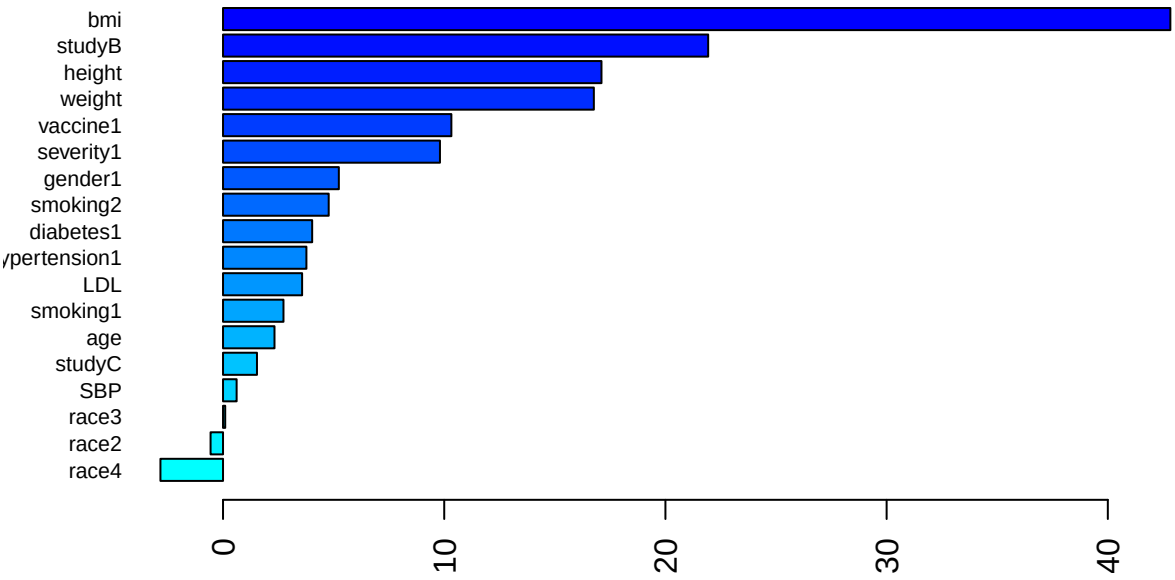
```
rf.grid <- expand.grid(mtry = 1:ncol(train.x),
                       splitrule = "variance",
                       min.node.size = 1:10)
set.seed(2023)
rf.fit <- train(train.x,
                train.y,
                method = "ranger",
                tuneGrid = rf.grid,
                trControl = ctrl1)

rf.fit$bestTune
```

```
##    mtry splitrule min.node.size
## 69    7  variance             9
```

```
ggplot(rf.fit, highlight = TRUE) +
  labs(title = "Random Forest CV Result") +
  theme_bw()
```




```
ggsave("./figure/rf_cv.jpeg", dpi = 500)

rf.final.per <- ranger(recovery_time ~ .,
                       data = train.dat.matrix,
                       mtry = rf.fit$bestTune[[1]],
                       splitrule = "variance",
                       min.node.size = rf.fit$bestTune[[3]],
                       importance = "permutation",
                       scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(rf.final.per), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(ncol(train.x)))
```

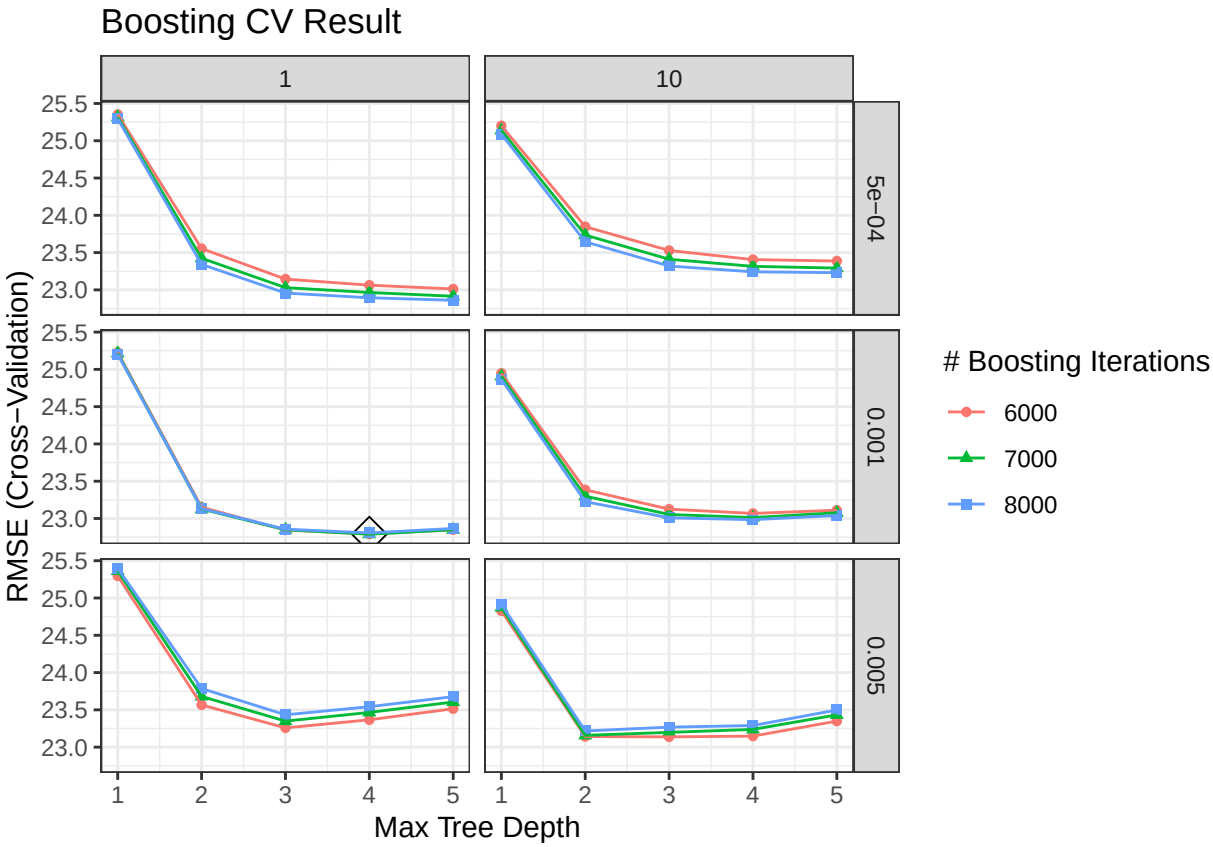


3.2.12 Boosting

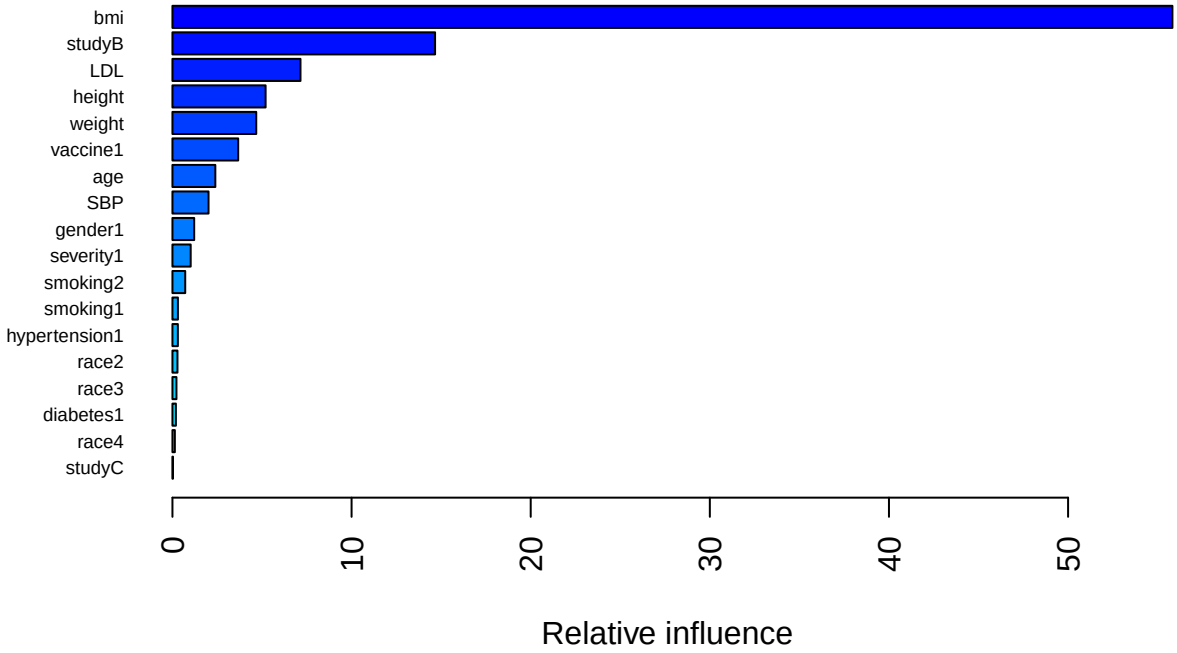
```
set.seed(2023)
bst.grid <- expand.grid(n.trees = c(6000, 7000, 8000),
                        interaction.depth = 1:5,
                        shrinkage = c(0.0005, 0.001,0.005),
                        n.minobsinnode = c(1,10))

bst.fit <- train(train.x,
                 train.y,
                 method = "gbm",
                 tuneGrid = bst.grid,
                 trControl = ctrl1,
                 verbose = FALSE)

bst.fit$bestTune
```

```
##    n.trees interaction.depth shrinkage n.minobsinnode
## 50    7000                 4     0.001              1
```

```
ggplot(bst.fit, highlight = TRUE) +
  labs(title = "Boosting CV Result") +
  theme_bw()
```




```
ggsave("./figure/boosting_cv.jpeg", dpi = 500)

# Variable Importance
summary(bst.fit$finalModel, las = 2, cBars = ncol(train.x), cex.names = 0.6)
```

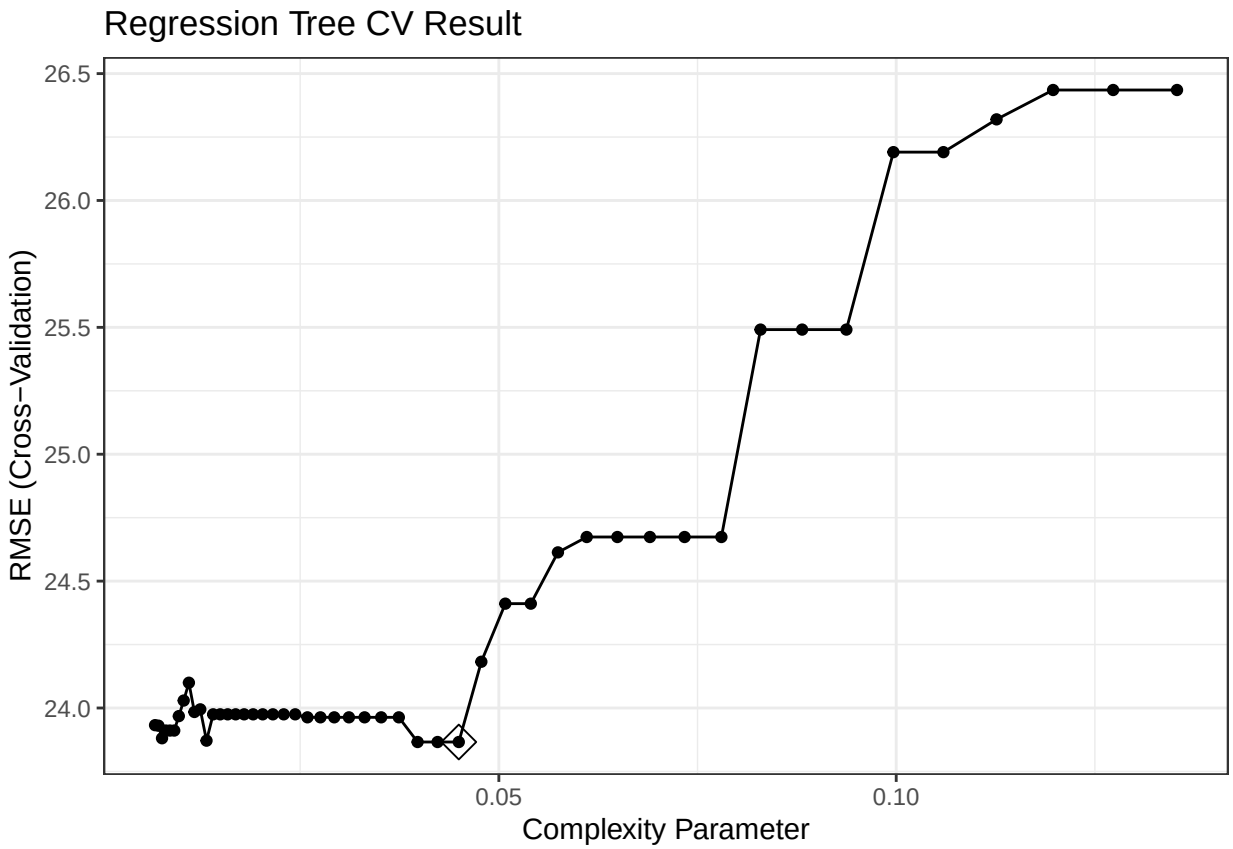



```
##                        var     rel.inf
```

```
## bmi                    bmi 55.82435579
## studyB              studyB 14.66082079
## LDL                    LDL  7.14824322
## height              height  5.20099563
## weight              weight  4.67820774
## vaccine1          vaccine1  3.66997663
## age                    age  2.39369807
## SBP                    SBP  2.01391839
## gender1            gender1  1.21406824
## severity1        severity1  1.01825404
## smoking2          smoking2  0.71254465
## smoking1          smoking1  0.30551973
## hypertension1 hypertension1  0.30244396
## race2                race2  0.28011065
## race3                race3  0.22397720
## diabetes1        diabetes1  0.19176667
## race4                race4  0.13327971
## studyC              studyC  0.02781888
```

3.2.13 Regression Trees

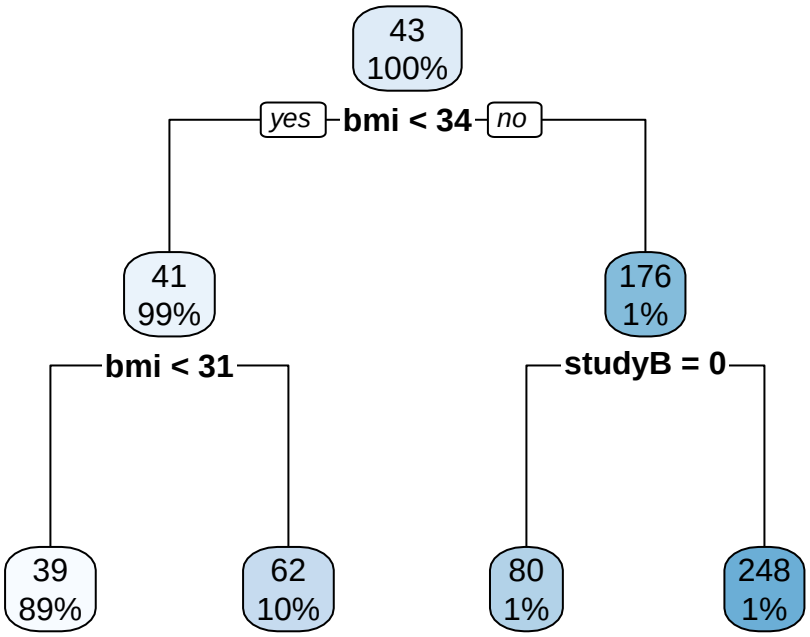
```
rpart.grid <- expand.grid(cp = exp(seq(-5,-2, length = 50)))
set.seed(2023)
rpart.fit1 <- train(train.x,
                    train.y,
                    method = "rpart",
                    tuneGrid = rpart.grid,
                    trControl = ctrl1)

ggplot(rpart.fit1, highlight = TRUE) +
  labs(titlem = "Regression Tree CV Result") +
  theme_bw()
```




```
ggsave("./figure/rpart1_cv.jpeg", dpi = 500)

rpart.fit1$bestTune
```

```
##            cp
## 32 0.04495736
```

```
rpart.plot(rpart.fit1$finalModel)
```




```
jpeg("./figure/rpart1.jpeg", width = 8, height = 6, units="in", res=500)
rpart.plot(rpart.fit1$finalModel)
dev.off()
```

```
## pdf
##   2
```

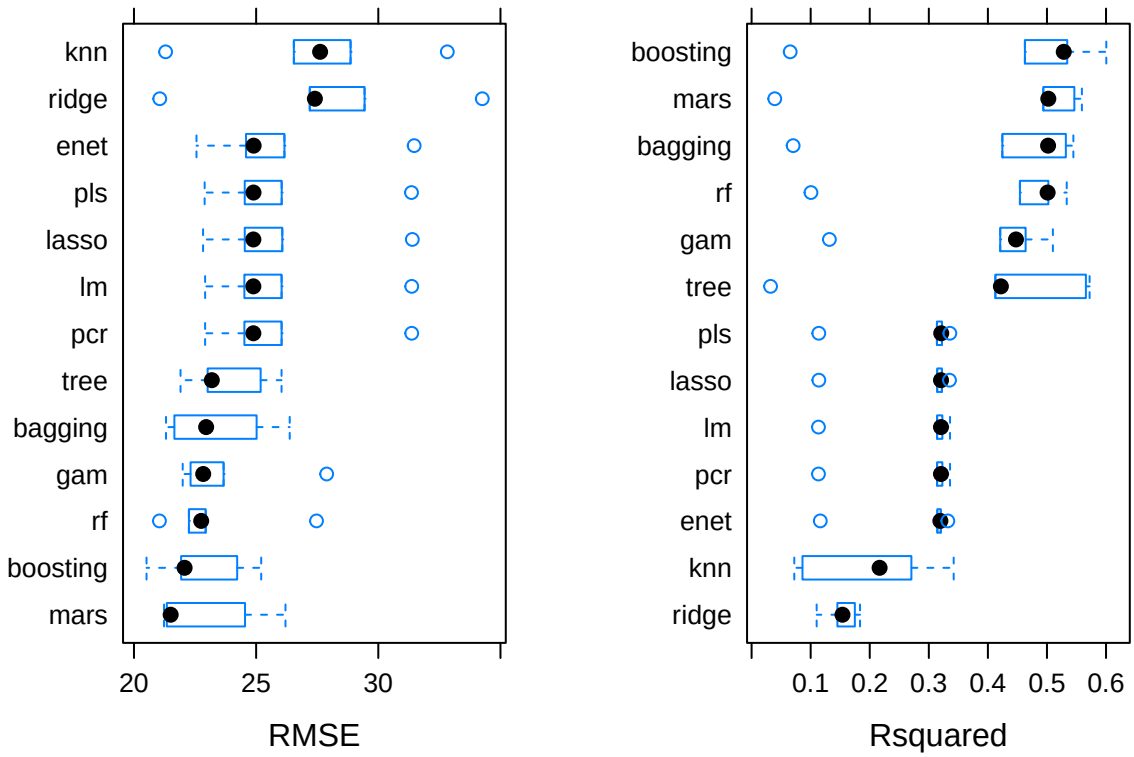
3.3 Model Selection

```
set.seed(2023)
resamp <- resamples(list(lm = lm.fit,
                         lasso = lasso.fit,
                         ridge = ridge.fit,
                         enet = enet.fit,
                         pcr = pcr.fit,
                         pls = pls.fit,
                         gam = gam.fit,
                         mars = mars.fit,
                         knn = knn.fit,
                         bagging = bag.fit,
                         rf = rf.fit,
                         boosting = bst.fit,
                         tree = rpart.fit1))

summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: lm, lasso, ridge, enet, pcr, pls, gam, mars, knn, bagging, rf, boosting, tree
## Number of resamples: 5
##
## MAE
```

```
##              Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lm       16.23173 16.30828 16.52868 16.61900 16.59544 17.43085    0
## lasso    16.21044 16.29219 16.49725 16.58977 16.53867 17.41030    0
## ridge    15.98311 16.49651 16.70285 16.85431 17.12329 17.96581    0
## enet     16.16042 16.24960 16.36222 16.51089 16.42530 17.35693    0
## pcr      16.23173 16.30828 16.52868 16.61900 16.59544 17.43085    0
## pls      16.21851 16.31098 16.52109 16.60977 16.57408 17.42418    0
## gam      14.96149 15.31743 15.35347 15.37421 15.37746 15.86122    0
## mars     14.48861 14.54724 14.58285 15.06219 15.81591 15.87632    0
## knn      15.16105 15.73142 15.79086 16.13709 16.38626 17.61583    0
## bagging  14.58743 14.76587 15.07453 15.19909 15.69039 15.87722    0
## rf       14.26815 14.67895 14.92547 14.98461 15.22321 15.82726    0
## boosting 14.09554 14.53527 14.69015 14.71165 15.03535 15.20193    0
## tree     14.91705 15.21864 15.45248 15.51302 15.91231 16.06462    0
##
## RMSE
##              Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lm       22.91150 24.51831 24.88597 25.94526 26.04242 31.36808    0
## lasso    22.82812 24.53373 24.88628 25.94219 26.07031 31.39252    0
## ridge    21.05565 27.19445 27.40378 27.86946 29.43813 34.25528    0
## enet     22.55834 24.58644 24.89736 25.93351 26.15765 31.46776    0
## pcr      22.91150 24.51831 24.88597 25.94526 26.04242 31.36808    0
## pls      22.89214 24.53226 24.89142 25.94392 26.04558 31.35821    0
## gam      21.99363 22.31575 22.83368 23.73975 23.66851 27.88718    0
## mars     21.22944 21.34499 21.50208 22.96420 24.54460 26.19988    0
## knn      21.29395 26.54338 27.61785 27.42866 28.86421 32.82391    0
## bagging  21.31058 21.65291 22.95364 23.46197 25.01913 26.37361    0
## rf       21.04171 22.24775 22.75182 23.28937 22.93423 27.47134    0
## boosting 20.51403 21.92962 22.07191 22.78917 24.22341 25.20691    0
## tree     21.90520 23.01458 23.18530 23.86591 25.18408 26.04041    0
##
## Rsquared
##                Min.    1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm       0.11319560 0.31401617 0.3206204 0.2813453 0.3229742 0.3359202    0
## lasso    0.11386516 0.31421662 0.3206286 0.2811877 0.3221851 0.3350431    0
## ridge    0.11015851 0.14523321 0.1538663 0.1535189 0.1747868 0.1835499    0
## enet     0.11628655 0.31442226 0.3195815 0.2805479 0.3202588 0.3321902    0
## pcr      0.11319560 0.31401617 0.3206204 0.2813453 0.3229742 0.3359202    0
## pls      0.11377107 0.31374985 0.3210112 0.2812813 0.3222148 0.3356594    0
## gam      0.13171077 0.42087488 0.4475492 0.3948083 0.4638793 0.5100272    0
## mars     0.03921635 0.49381643 0.5023136 0.4281672 0.5463545 0.5591351    0
## knn      0.07225895 0.08613868 0.2166700 0.1975252 0.2704826 0.3420757    0
## bagging  0.07033870 0.42448137 0.5019016 0.4146791 0.5319576 0.5447161    0
## rf       0.10049299 0.45424922 0.5011111 0.4183419 0.5024069 0.5334491    0
## boosting 0.06532432 0.46267491 0.5284672 0.4382426 0.5343027 0.6004439    0
## tree     0.03210038 0.41264371 0.4220056 0.4009682 0.5659027 0.5721886    0
```

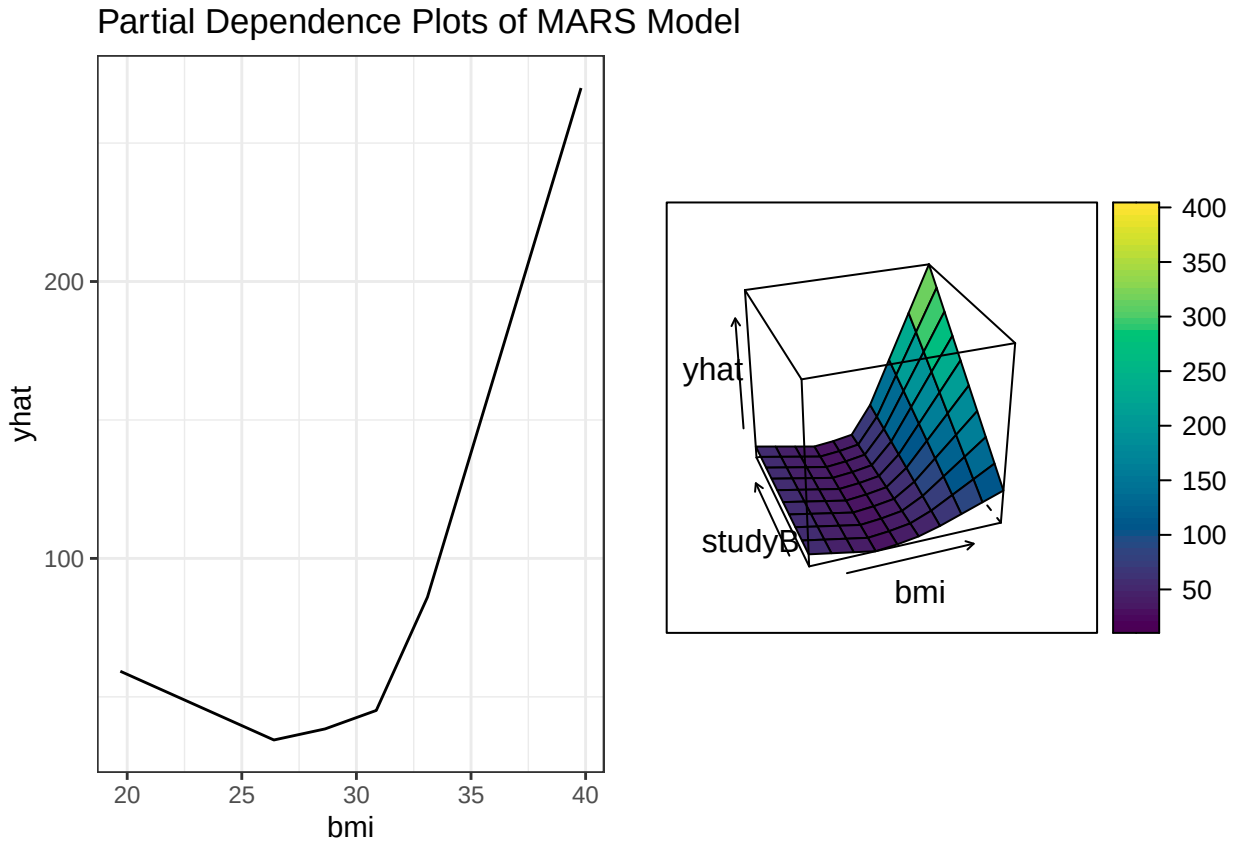
```
p1=bwplot(resamp, metric = "RMSE")
p2=bwplot(resamp, metric = "Rsquared")
grid.arrange(p1, p2 ,ncol=2)
```




```
jpeg("./figure/resample1.jpeg", width = 8, height=6, units="in", res=500)
p1=bwplot(resamp, metric = "RMSE")
p2=bwplot(resamp, metric = "Rsquared")
grid.arrange(p1, p2, ncol=2)
dev.off()
```

```
## pdf
##   2
```

```
p1<- pdp::partial(mars.fit, pred.var = c("bmi"), grid.resolution = 10) %>% autoplot() +
  theme_bw()+
  labs(title = "Partial Dependence Plots of MARS Model")

p2 <-pdp::partial(mars.fit, pred.var = c("bmi", "studyB"),
                  grid.resolution = 10) %>%
      pdp::plotPartial(levelplot = FALSE, zlab = "yhat", drape = TRUE,
                       screen = list(z = 20, x = -60))

# jpeg("./figure/partial_dependence.jpeg", width = 8, height=6, units="in", res=500)
gridExtra::grid.arrange(p1, p2, ncol = 2)
```




```
# dev.off()

# Important variables
varImp(mars.fit$finalModel)
```

```
##             Overall
## bmi      100.00000
## studyB   100.00000
## vaccine1  17.78457
```

3.4 Training / Testing Error

```
# training error
mars.train.pred = predict(mars.fit, newdata = train.x)
RMSE(train.y, mars.train.pred)
```

```
## [1] 22.07828
```

```
# testing error
mars.pred = predict(mars.fit, newdata = test.x)
RMSE(test.y, mars.pred)
```

```
## [1] 22.1712
```

4 Secondary Analysis

4.1 Exploratory analysis and data visualization

```
# data summary
st_options(plain.ascii = FALSE,
           style = "rmarkdown",
           dfSummary.silent = TRUE,
           footnote = NA,
           subtitle.emphasis = FALSE)
dfSummary(train.bin.dat)
```

4.1.1 Data Frame Summary

train.bin.dat
Dimensions: 2900 x 15
Duplicates: 0

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	age [numeric]	Mean (sd) : 60.1 (4.5) min < med < max: 45 < 60 < 77 IQR (CV) : 6 (0.1)	33 distinct values	: : . : : : : . . : : : : : : .	2900 (100.0%)	0 (0.0%)
2	gender [factor]	1. 0 2. 1	1468 (50.6%) 1432 (49.4%)	IIIIIIIIII IIIIIIIII	2900 (100.0%)	0 (0.0%)
3	race [factor]	1. 1 2. 2 3. 3 4. 4	1909 (65.8%) 132 (4.6%) 568 (19.6%) 291 (10.0%)	IIIIIIIIIIII III II	2900 (100.0%)	0 (0.0%)
4	smoking [factor]	1. 0 2. 1 3. 2	1763 (60.8%) 845 (29.1%) 292 (10.1%)	IIIIIIIIIII IIIII II	2900 (100.0%)	0 (0.0%)
5	height [numeric]	Mean (sd) : 170.2 (6) min < med < max: 151.2 < 170.1 < 188.6 IQR (CV) : 8 (0)	312 distinct values	: : : : . : : : : : : : .	2900 (100.0%)	0 (0.0%)
6	weight [numeric]	Mean (sd) : 80.2 (7) min < med < max: 57.1 < 80.3 < 103.4 IQR (CV) : 9.5 (0.1)	361 distinct values	. : . : : : : : : . . : : : : : : : .	2900 (100.0%)	0 (0.0%)
7	bmi [numeric]	Mean (sd) : 27.8 (2.7) min < med < max: 19.7 < 27.7 < 39.8 IQR (CV) : 3.6 (0.1)	160 distinct values	: . : : : : . : : : : : : : .	2900 (100.0%)	0 (0.0%)
8	hypertension [factor]	1. 0 2. 1	1514 (52.2%) 1386 (47.8%)	IIIIIIIIII IIIIIIIII	2900 (100.0%)	0 (0.0%)
9	diabetes [factor]	1. 0 2. 1	2446 (84.3%) 454 (15.7%)	IIIIIIIIIIIIIII III	2900 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
10	SBP [numeric]	Mean (sd) : 130.2 (8.1) min < med < max: 104 < 130 < 158 IQR (CV) : 11 (0.1)	54 distinct values	: : . : : : . . : : : : : . : : : : : : : :	2900 (100.0%)	0 (0.0%)
11	LDL [numeric]	Mean (sd) : 110.3 (19.9) min < med < max: 32 < 110 < 174 IQR (CV) : 27 (0.2)	116 distinct values	. : : : : : : : : : : : : : : : : : : : .	2900 (100.0%)	0 (0.0%)
12	vaccine [factor]	1. 0 2. 1	1192 (41.1%) 1708 (58.9%)	IIIIIIII IIIIIIIIII	2900 (100.0%)	0 (0.0%)
13	severity [factor]	1. 0 2. 1	2619 (90.3%) 281 (9.7%)	IIIIIIIIIIIIIIIII I	2900 (100.0%)	0 (0.0%)
14	study [factor]	1. A 2. B 3. C	580 (20.0%) 1750 (60.3%) 570 (19.7%)	IIII IIIIIIIIIII III	2900 (100.0%)	0 (0.0%)
15	recovery_time [factor]	1. lt30 2. gt30	887 (30.6%) 2013 (69.4%)	IIIIII IIIIIIIIIIII	2900 (100.0%)	0 (0.0%)

```
skimr::skim_without_charts(train.bin.dat)
```

Table 6: Data summary

Name	train.bin.dat
Number of rows	2900
Number of columns	15
Column type frequency:	
factor	9
numeric	6
Group variables	None

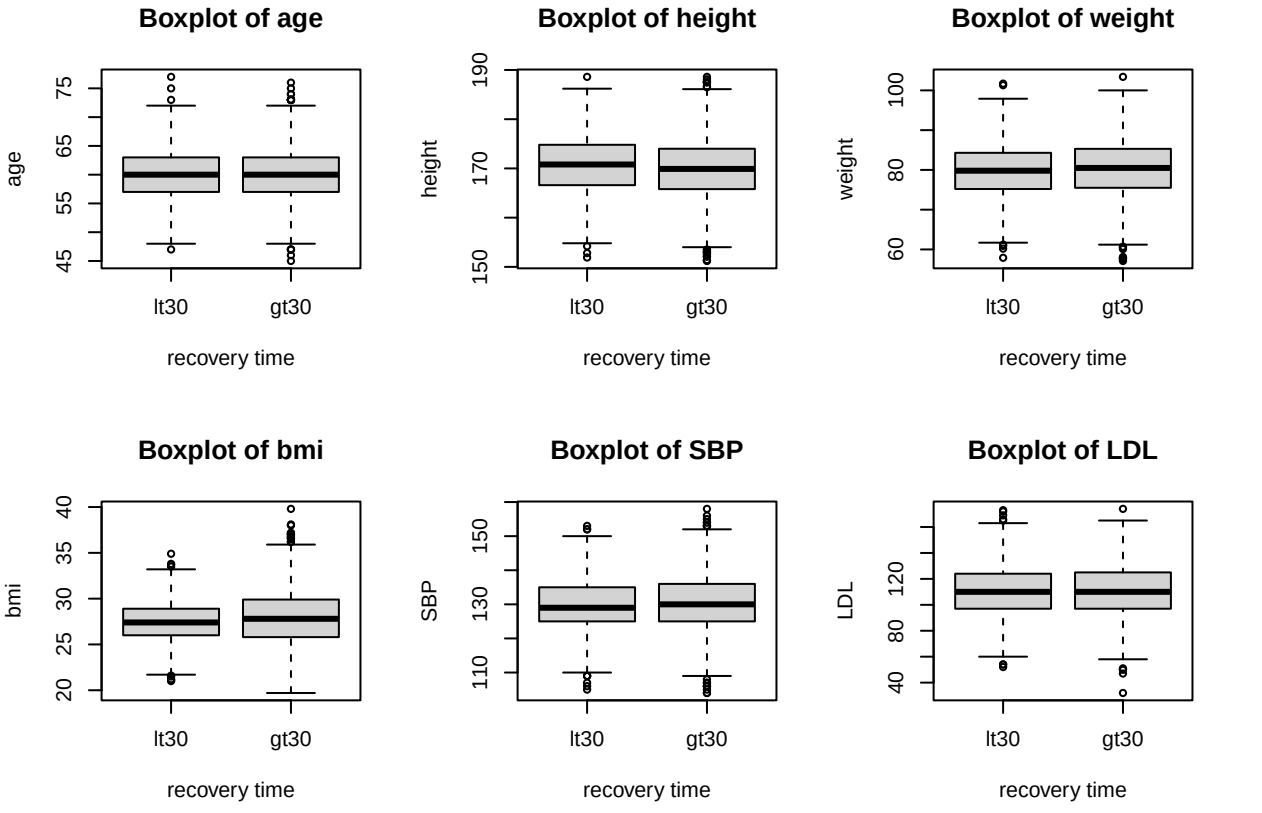
Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gender	0	1	FALSE	2	0: 1468, 1: 1432
race	0	1	FALSE	4	1: 1909, 3: 568, 4: 291, 2: 132
smoking	0	1	FALSE	3	0: 1763, 1: 845, 2: 292
hypertension	0	1	FALSE	2	0: 1514, 1: 1386
diabetes	0	1	FALSE	2	0: 2446, 1: 454
vaccine	0	1	FALSE	2	1: 1708, 0: 1192
severity	0	1	FALSE	2	0: 2619, 1: 281
study	0	1	FALSE	3	B: 1750, A: 580, C: 570
recovery_time	0	1	FALSE	2	gt3: 2013, lt3: 887

Variable type: numeric

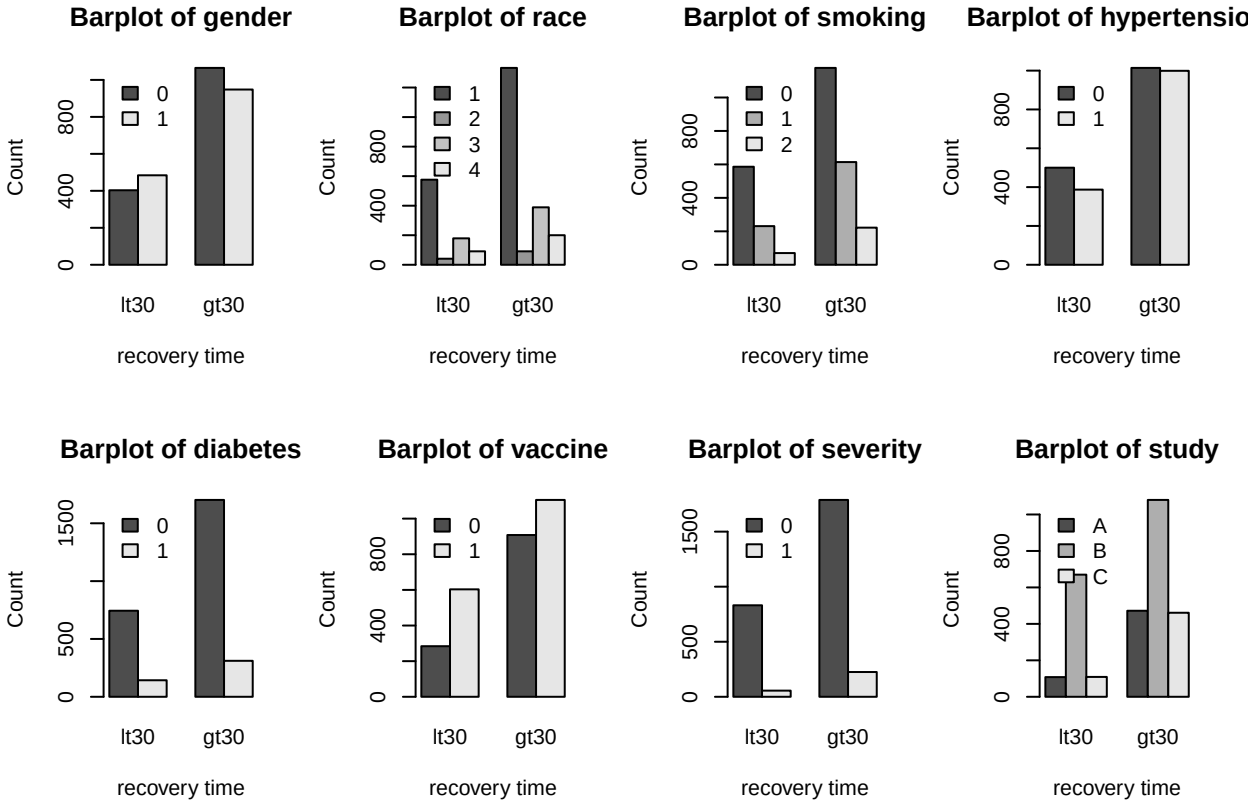
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
age	0	1	60.07	4.51	45.0	57.0	60.00	63.0	77.0
height	0	1	170.17	6.04	151.2	166.1	170.15	174.1	188.6
weight	0	1	80.20	7.00	57.1	75.4	80.30	84.9	103.4
bmi	0	1	27.76	2.73	19.7	25.9	27.70	29.5	39.8
SBP	0	1	130.19	8.08	104.0	125.0	130.00	136.0	158.0
LDL	0	1	110.27	19.87	32.0	97.0	110.00	124.0	174.0

```
#####################################################################
## Remember to edit the next chunk if you do any modification here:)
#####################################################################
# EDA

# boxplot of continuous predictors
par(mfrow=c(2, 3))
for (i in 1:length(cts_var)){
  var = cts_var[i]
  boxplot(train.bin.dat[,var]~recovery_time,
        data = train.bin.dat,
        xlab = "recovery time",
        ylab = var,
        main = str_c("Boxplot of ", var))
}
```

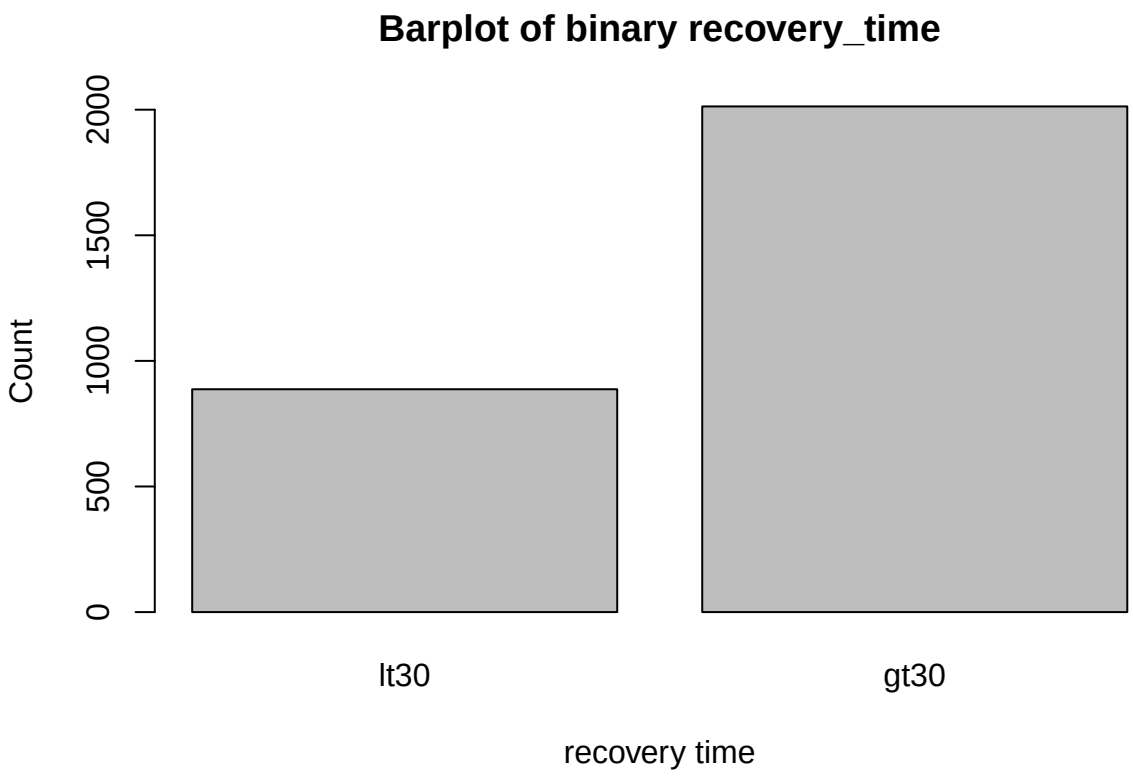



```
# barplot of categorical predictors
par(mfrow=c(2, 4))
for (i in 1:length(fct_var)){
```

```
  var <- fct_var[i]
  counts <- table(train.bin.dat[,var], train.bin.y)
  barplot(counts, beside = TRUE, legend.text = TRUE,
        xlab = "recovery time",
        ylab = "Count",
        main = str_c("Barplot of ", var),
        args.legend = list(bty = 'n', x = 'topleft'))
}
```




```
# barplot of response
par(mfrow=c(1, 1))
counts <- table(train.bin.y)
barplot(counts,
        xlab = "recovery time",
        ylab = "Count",
        main = "Barplot of binary recovery_time")
```



4.2 Model Training

```
contrasts(train.bin.y)
```

```
##      gt30
## lt30    0
## gt30    1
```

```
ctrl2 <- trainControl(method = "cv",
                      number = 5,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)
```

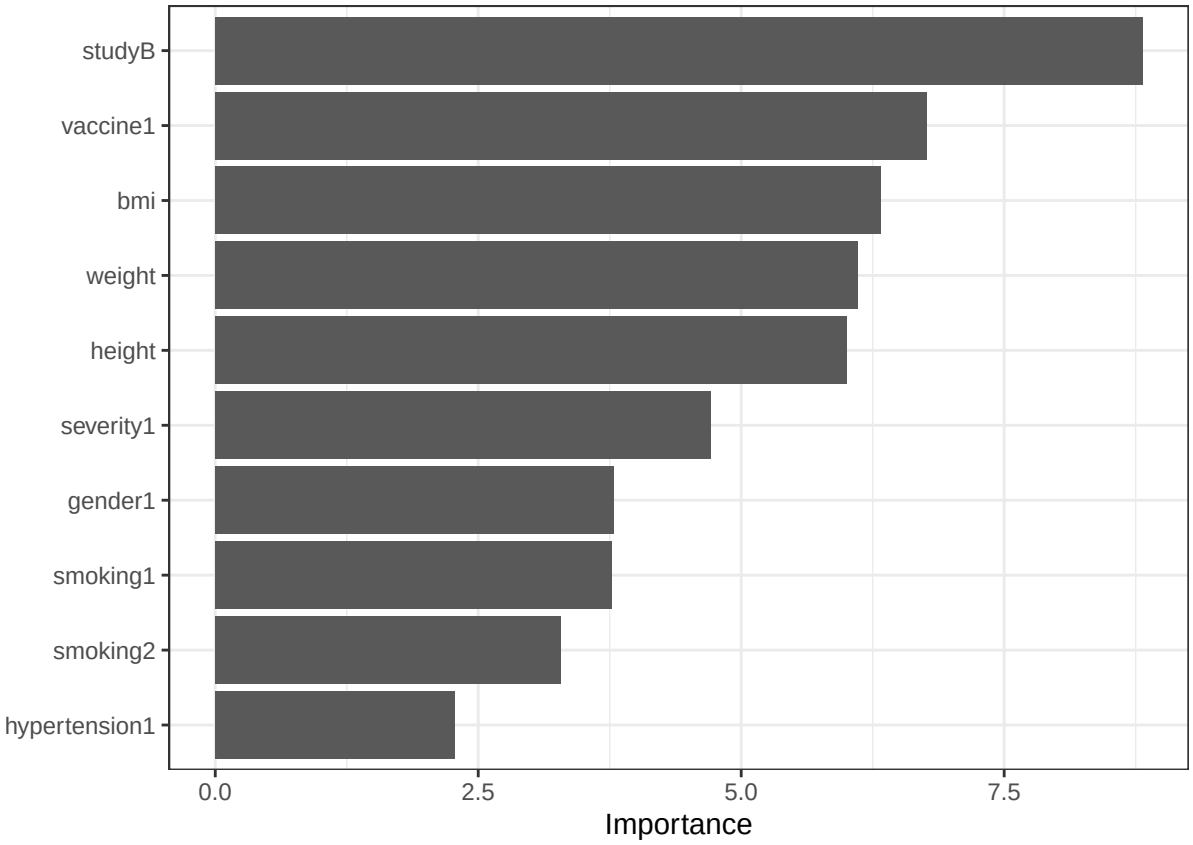
4.2.1 Logistic Regression

```
glm.fit <- train(x = train.x,
                 y = train.bin.y,
                 method = 'glm',
                 metric = 'ROC',
                 trControl = ctrl2)
coef(glm.fit$finalModel)
```

```
##   (Intercept)           age       gender1         race2         race3
## -85.313351100   0.014271893  -0.323467202  -0.104376256  -0.039351201
##         race4      smoking1      smoking2        height        weight
##   0.003550318   0.367190652   0.502782618   0.502637208  -0.545510559
##           bmi hypertension1     diabetes1           SBP           LDL
##   1.634689792   0.325697328  -0.070567897  -0.005832901  -0.001829020
##      vaccine1     severity1        studyB        studyC
```

```
##  -0.600151829   0.761039467  -1.066825060  -0.031460504
```

```
vip(glm.fit$finalModel) + theme_bw()
```

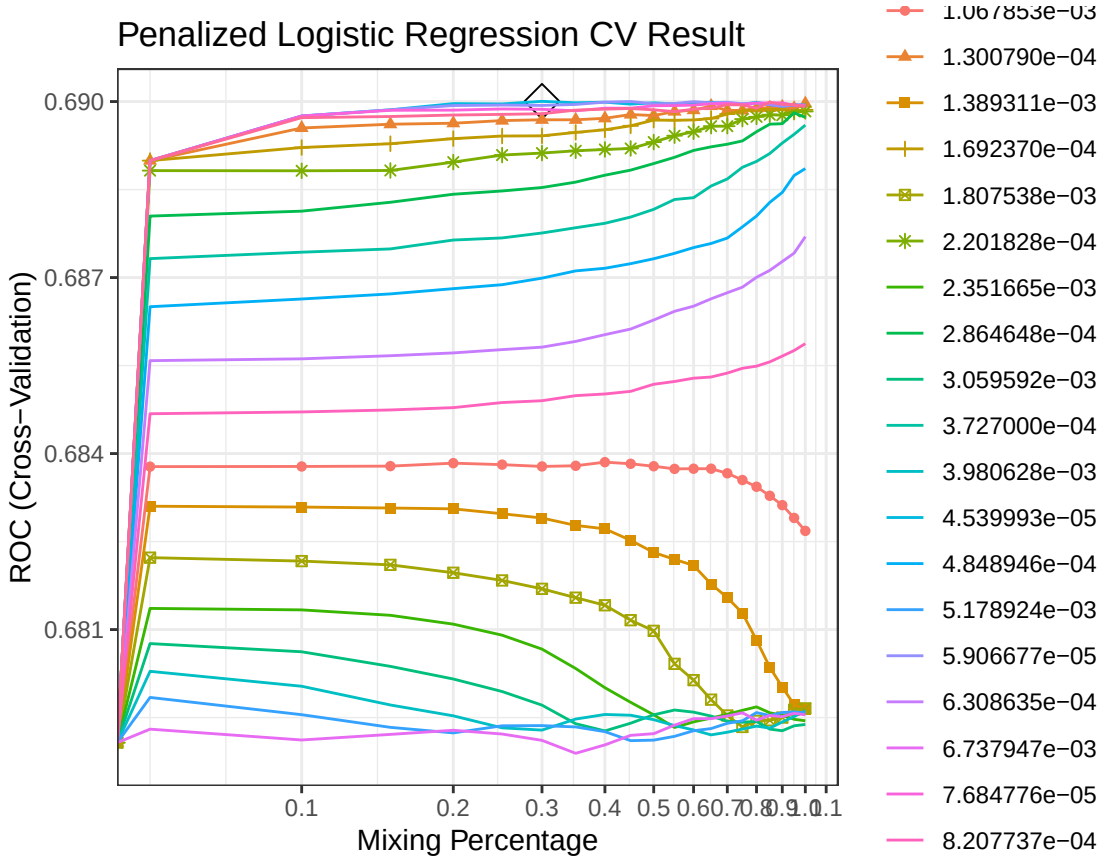


4.2.2 Penalized Logistic Regression

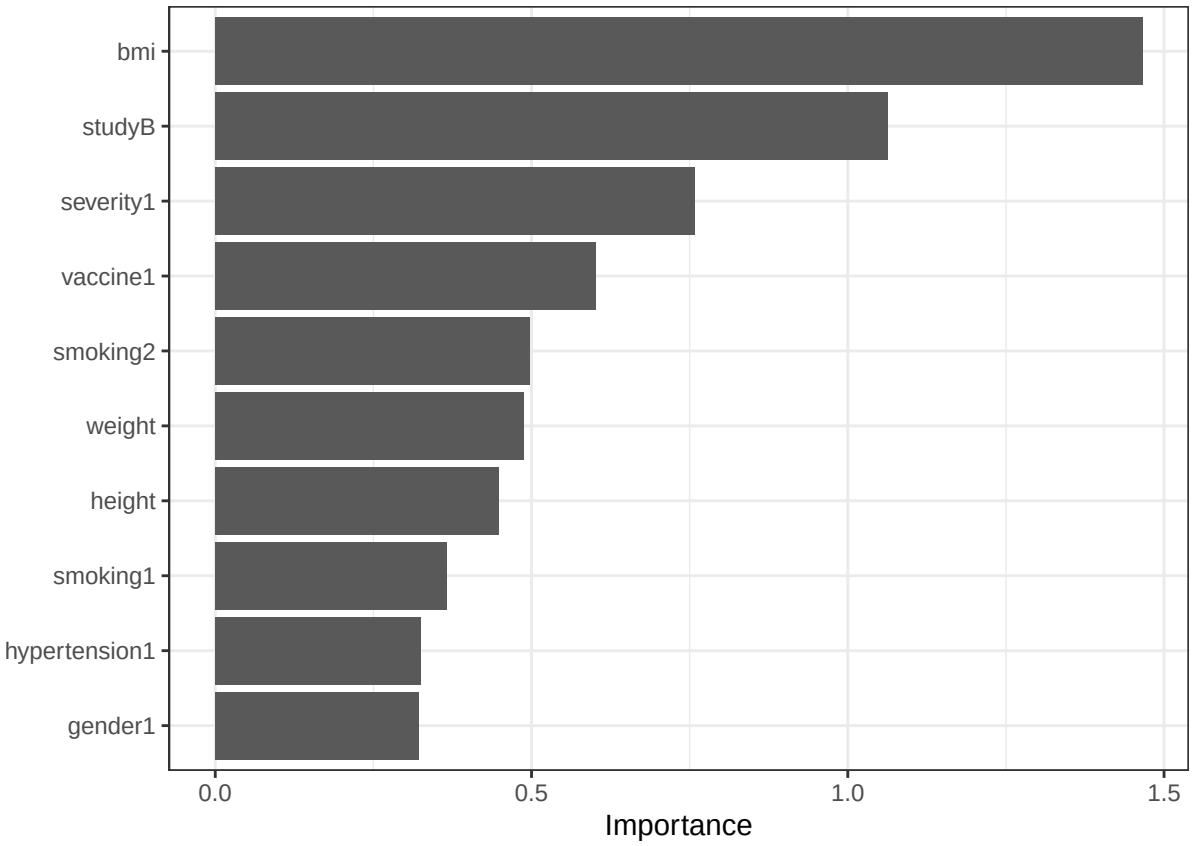
```
glmnGrid <- expand.grid(.alpha = seq(0, 1, length = 21),
                        .lambda = exp(seq(-10, -5, length = 20)))
set.seed(2023)
glmn.fit <- train(train.x,
                  train.bin.y,
                  method = 'glmnet',
                  tuneGrid = glmnGrid,
                  metric = 'ROC',
                  trControl = ctrl2)

glmn.fit$bestTune
```

```
##     alpha       lambda
## 121   0.3 4.539993e-05
```

```
myCol<- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
              superpose.line = list(col = myCol))
ggplot(glmn.fit, highlight = TRUE) +
  labs(title="Penalized Logistic Regression CV Result") +
  scale_x_continuous(trans='log',n.breaks = 10) +
  theme_bw()
```

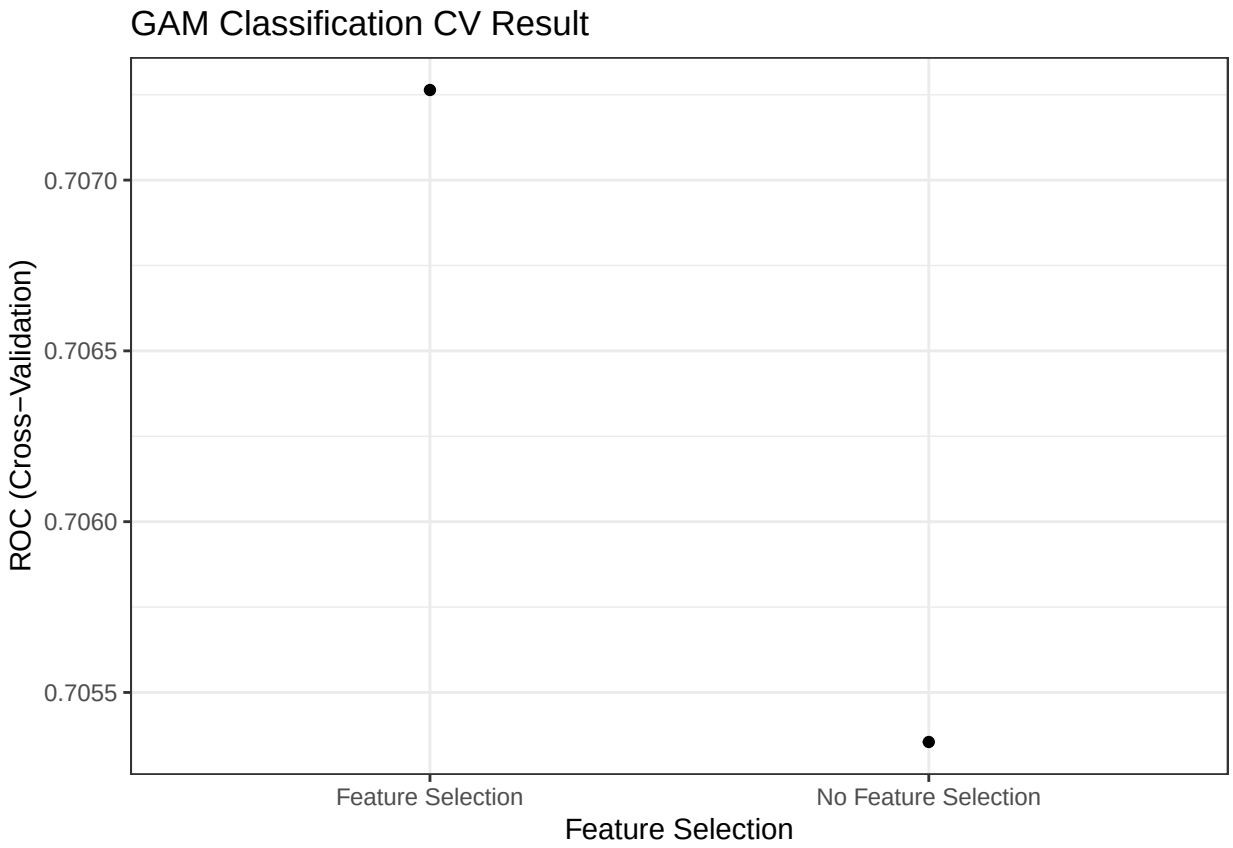



```
ggsave("./figure/penal_logi_cv.jpeg", dpi = 500)

#coef(glmn.fit$finalModel)
vip(glmn.fit$finalModel) + theme_bw()
```



4.2.3 Generalized Additive Model (GAM) for classification

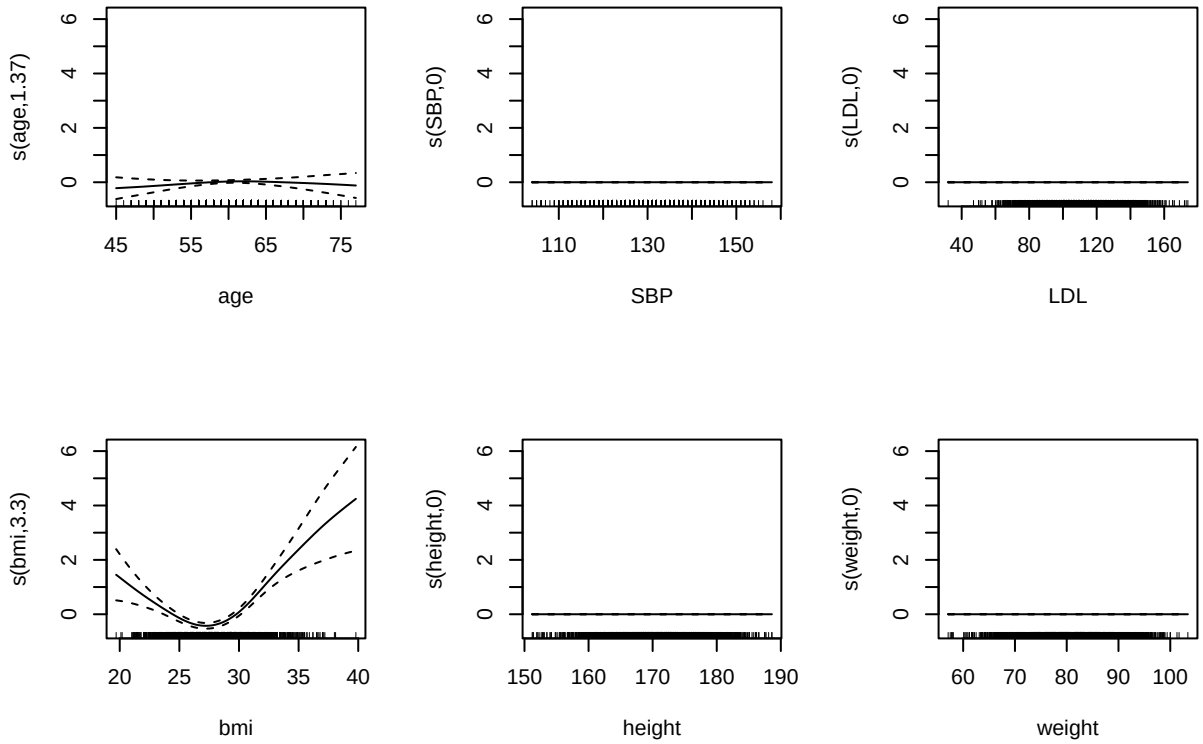
```
set.seed(2023)
gam.bin.fit <- train(train.x,
                     train.bin.y,
                     method = "gam",
                     metric = "ROC",
                     trControl = ctrl2)

ggplot(gam.bin.fit) +
  labs(title = "GAM Classification CV Result") +
  theme_bw()
```




```
ggsave("./figure/gam_binned_cv.jpeg", dpi = 500)

gam.bin.fit$bestTune
```

```
##   select method
## 2   TRUE GCV.Cp
```

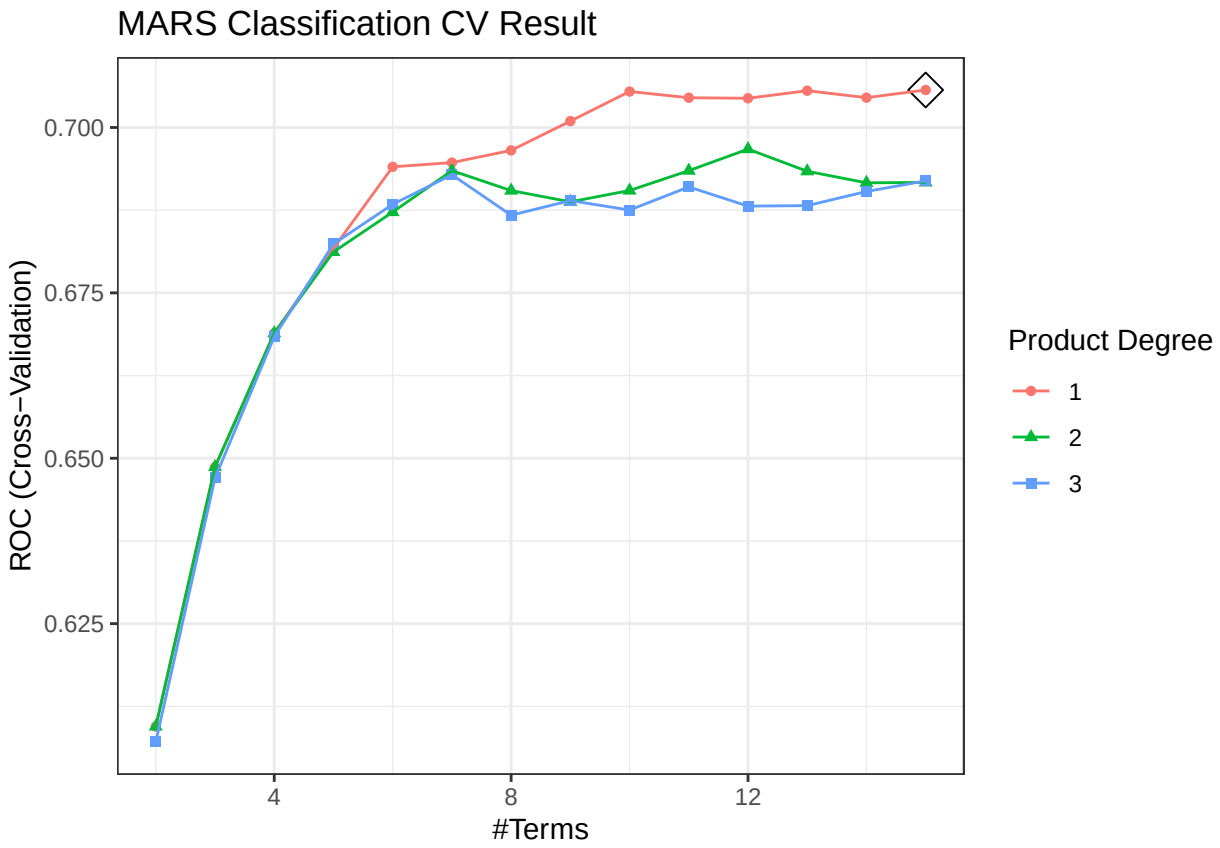
```
# coef(gam.fit$finalModel)

par(mfrow=c(2, 3))
plot(gam.bin.fit$finalModel)
```




```
par(mfrow=c(1, 1))
```

4.2.4 Multivariate Adaptive Regression Splines (MARS) for classification

```
set.seed(2023)
mars.bin.fit <- train(train.x,
                      train.bin.y,
                      method = "earth",
                      tuneGrid = expand.grid(degree = 1:3,
                                             nprune = 2:15),
                      metric = "ROC",
                      trControl = ctrl2)
ggplot(mars.bin.fit, highlight = TRUE)+
  labs(title  ="MARS Classification CV Result") +
  theme_bw()
```




```
ggsave("./figure/mars_binned_cv.jpeg", dpi = 500)

mars.bin.fit$bestTune
```

```
##    nprune degree
## 14     15      1
```

```
coef(mars.bin.fit$finalModel) %>%
  broom::tidy() %>%
  knitr::kable()
```

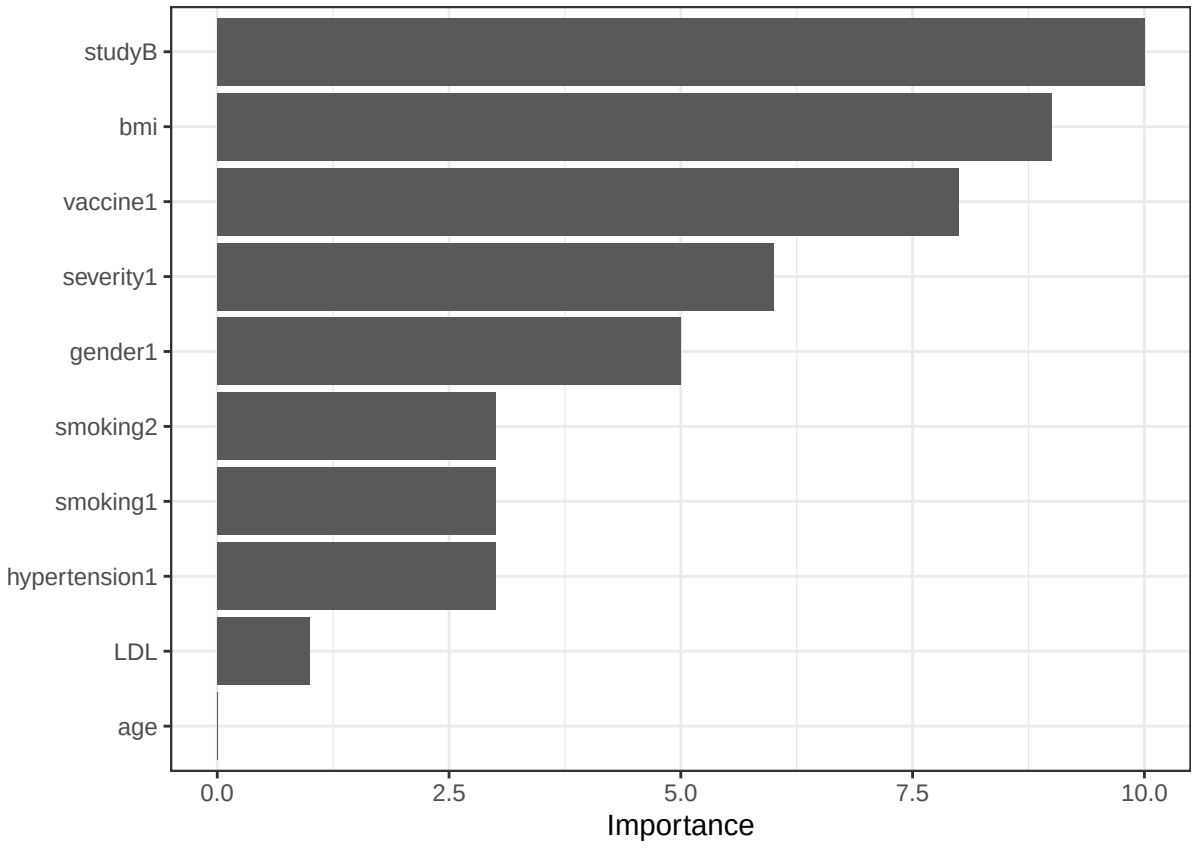
names	x
(Intercept)	1.1011705
studyB	-1.0779091
h(bmi-26.9)	0.2900212
h(26.9-bmi)	0.2935615
vaccine1	-0.6217928
severity1	0.7969230
gender1	-0.3261333
hypertension1	0.3099788
smoking1	0.3912885
smoking2	0.5358382
h(LDL-157)	-0.1512777

```
summary(mars.bin.fit$finalModel)
```

```
## Call: earth(x=matrix[2900,18], y=factor.object, keepxy=TRUE,
```

```
##             glm=list(family=function.object, maxit=100), degree=1, nprune=15)
##
## GLM coefficients
##                      gt30
## (Intercept)     1.1011705
## gender1        -0.3261333
## smoking1        0.3912885
## smoking2        0.5358382
## hypertension1   0.3099788
## vaccine1       -0.6217928
## severity1       0.7969230
## studyB         -1.0779090
## h(26.9-bmi)     0.2935615
## h(bmi-26.9)     0.2900212
## h(LDL-157)     -0.1512777
##
## GLM (family binomial, link logit):
##  nulldev   df       dev   df   devratio     AIC iters converged
##  3571.35 2899   3204.42 2889      0.103    3226     4         1
##
## Earth selected 11 of 14 terms, and 9 of 18 predictors (nprune=15)
## Termination condition: RSq changed by less than 0.001 at 14 terms
## Importance: studyB, bmi, vaccine1, severity1, gender1, smoking1, smoking2, ...
## Number of terms at each degree of interaction: 1 10 (additive model)
## Earth GCV 0.1906834    RSS 545.0022    GRSq 0.1024844    RSq 0.1148255
```

```
vip(mars.bin.fit$finalModel) + theme_bw()
```



4.2.5 Linear Discriminant Analysis (LDA)

4.2.6 Quadratic Discriminant Analysis (QDA)

4.2.7 Naive Bayes (NB)

4.2.8 Bagging

4.2.9 Random Forest

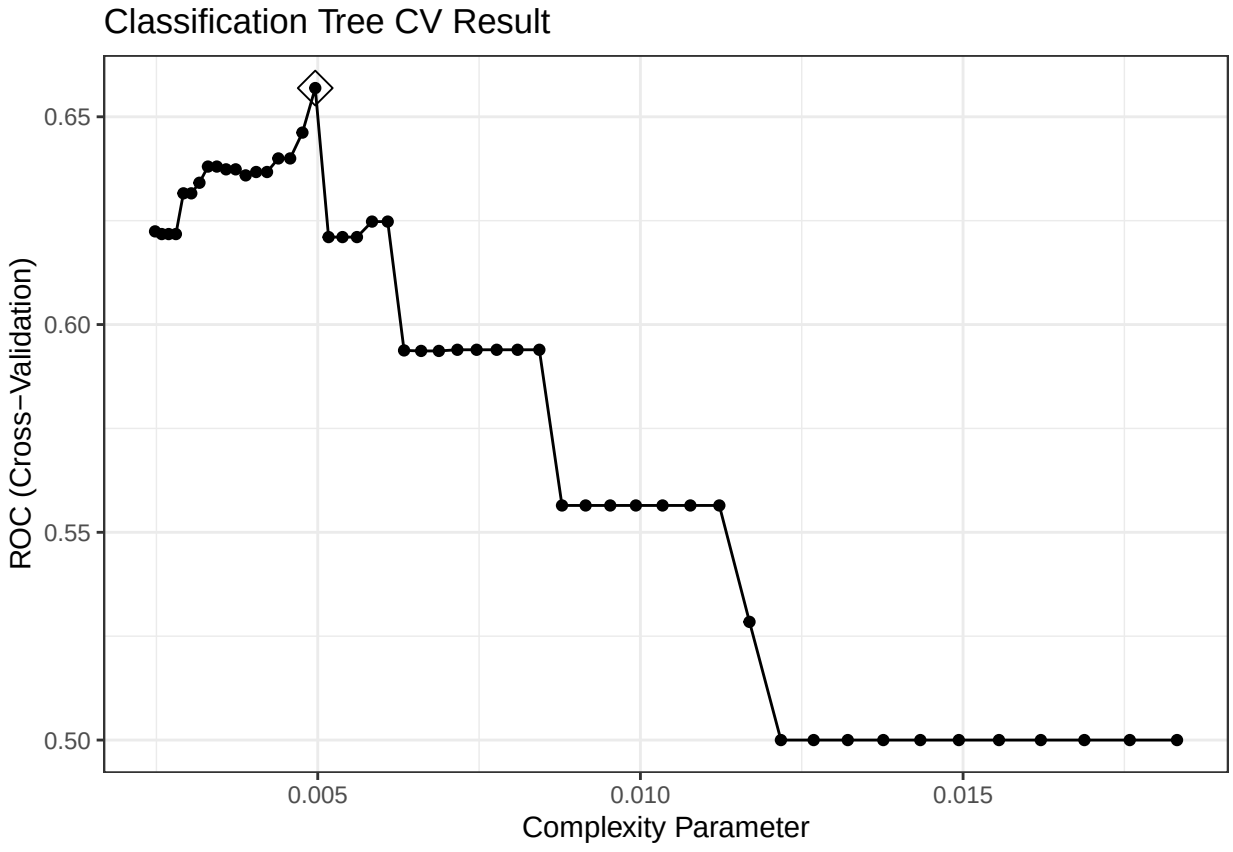
4.2.10 Boosting

4.2.11 Classification Trees

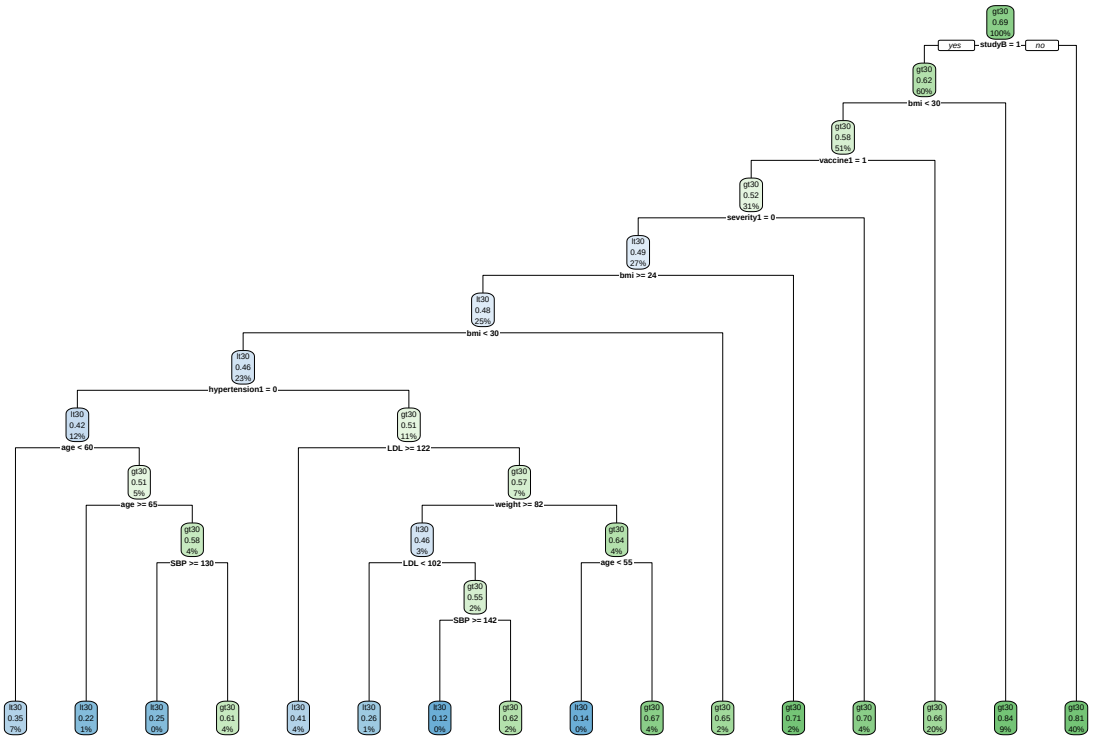
```
rpart.grid = expand.grid(cp = exp(seq(-6,-4, len = 50)))
set.seed(2023)
rpart.fit2 <- train(train.x,
                    train.bin.y,
                    method = "rpart",
                    tuneGrid = rpart.grid,
                    trControl = ctrl2,
                    metric = "ROC")
rpart.fit2$bestTune
```

```
##             cp
## 18 0.004961126
```

```
ggplot(rpart.fit2, highlight = TRUE) +
  labs(title = "Classification Tree CV Result") +
  theme_bw()
```



```
ggsave("./figure/rpart2_cv.jpeg", dpi = 500)

rpart.plot(rpart.fit2$finalModel)
```




```
jpeg("./figure/rpart2.jpeg", width = 8, height = 6, units="in", res=500)
rpart.plot(rpart.fit2$finalModel)
dev.off()
```

```
## pdf
##   2
```

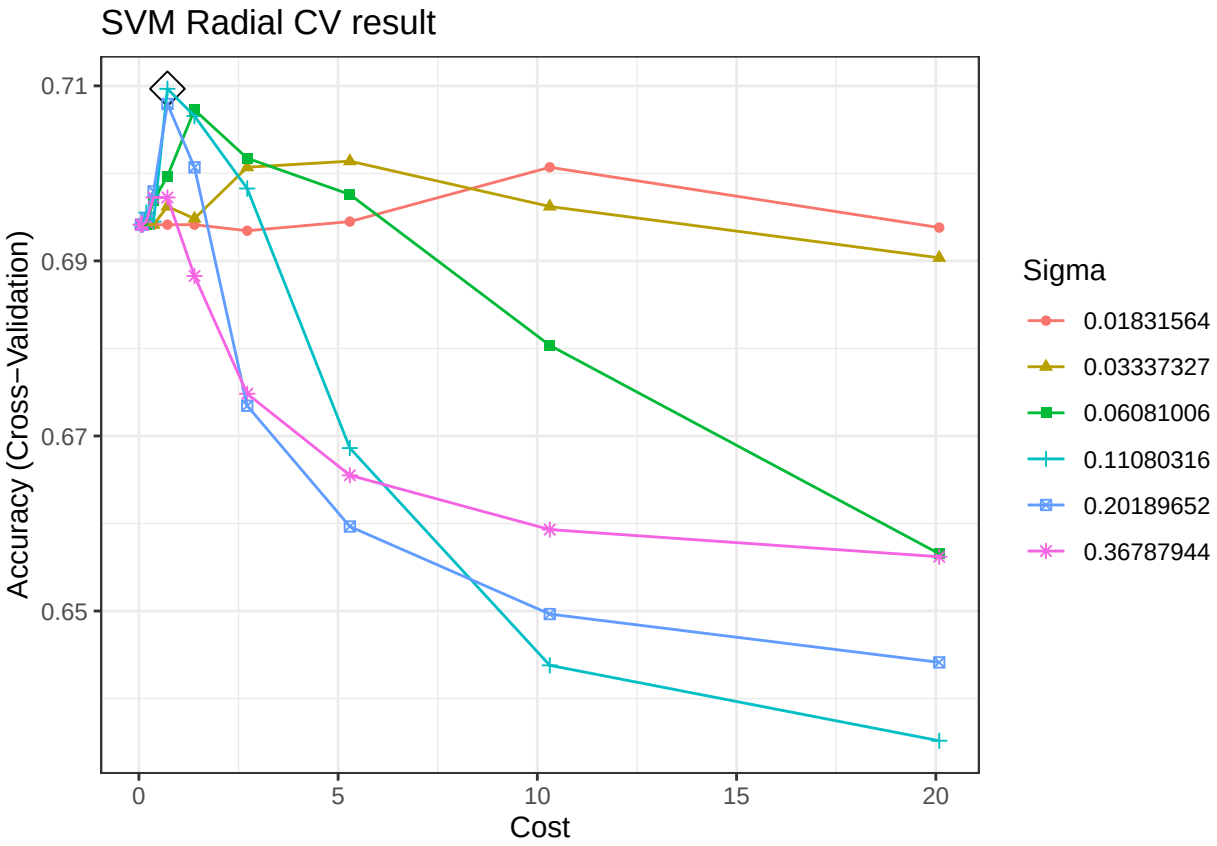
4.2.12 Support Vector Machine (SVM)

```
svmr.grid <- expand.grid(C = exp(seq(-3, 3, len = 10)),
                         sigma = exp(seq(-4, -1, len = 6)))

set.seed(2023)
svmr.fit <- train(train.x,
                  train.bin.y,
                  method = "svmRadialSigma",
                  tuneGrid = svmr.grid,
                  trControl = ctrl1)

svmr.fit$bestTune
```

```
##        sigma         C
## 28 0.1108032 0.7165313
```

```
myCol<- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
              superpose.line = list(col = myCol))
```

```
ggplot(svmr.fit, highlight = TRUE, par.settings = myPar) +
  labs(title = "SVM Radial CV result") +
  theme_bw()
```




```
ggsave("./figure/svmr_cv.jpeg", dpi = 500)
```

4.3 Model Selection

4.4 Training / Testing Error