Final Primary Analysis

Tianshu Liu, Lincole Jiang, Jiong Ma

Contents

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

1 Model Training

1.1 Primary Analysis

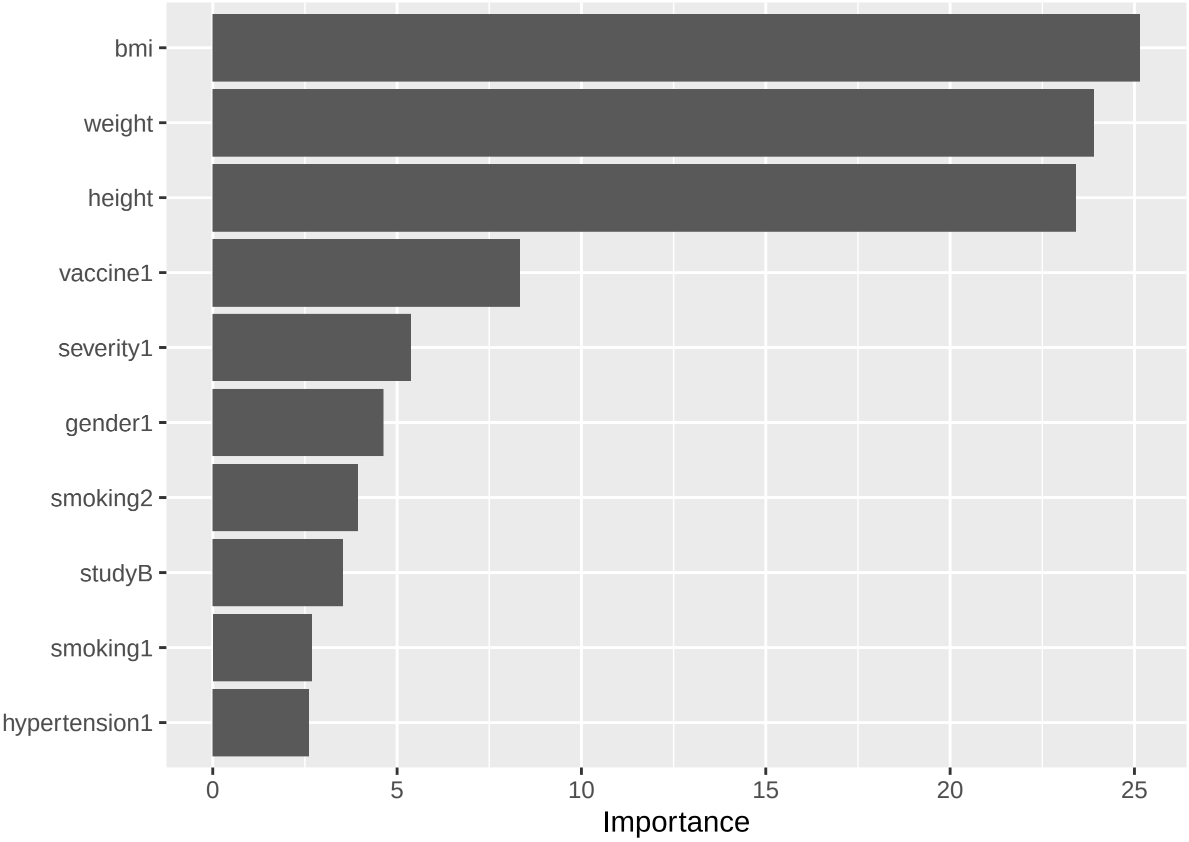
```r
ctrl1 <- trainControl(method = "cv", number = 10)
```

1.1.1 Linear Model

```r
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
```

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



1.1.2 LASSO

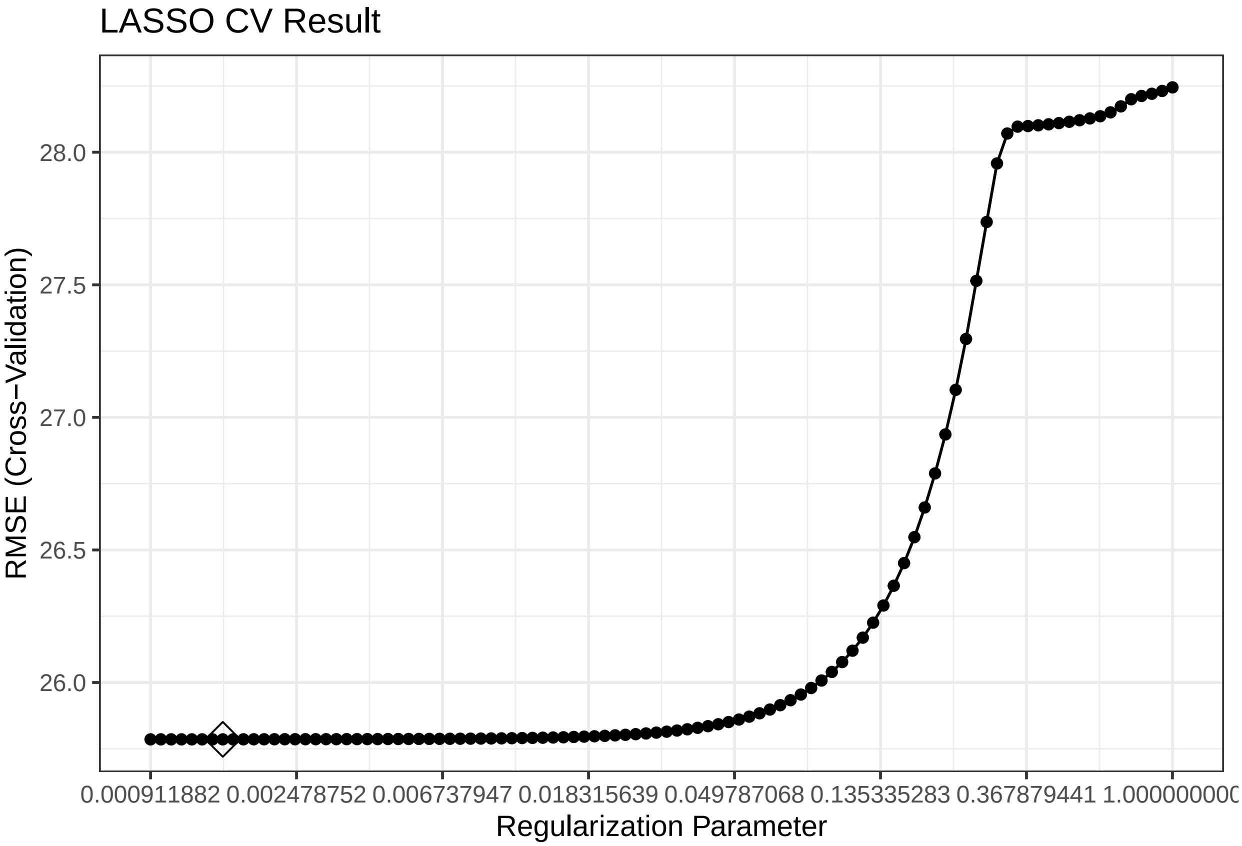
```
set.seed(2023)
lasso.fit <- train(train.x, train.y,
                   method = "glmnet",
                   tuneGrid = expand.grid(
                     alpha = 1,
                     lambda = exp(seq(0, -7, length=100))),
                   trControl = ctrl1)

lasso.fit$bestTune
```

```
##   alpha      lambda
## 8     1 0.001495865
```

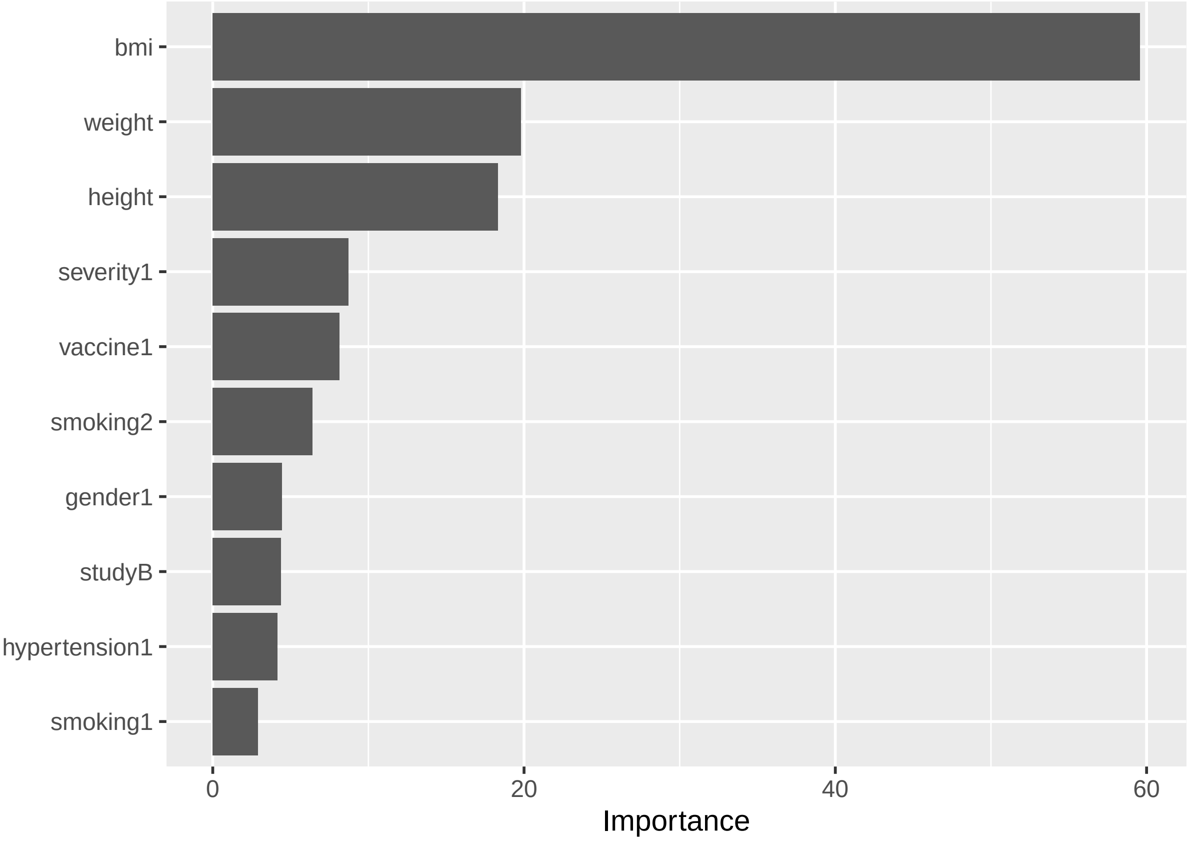
```
coef(lasso.fit$finalModel, s = lasso.fit$bestTune$lambda)
```

```
## 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)
```



1.1.3 Ridge

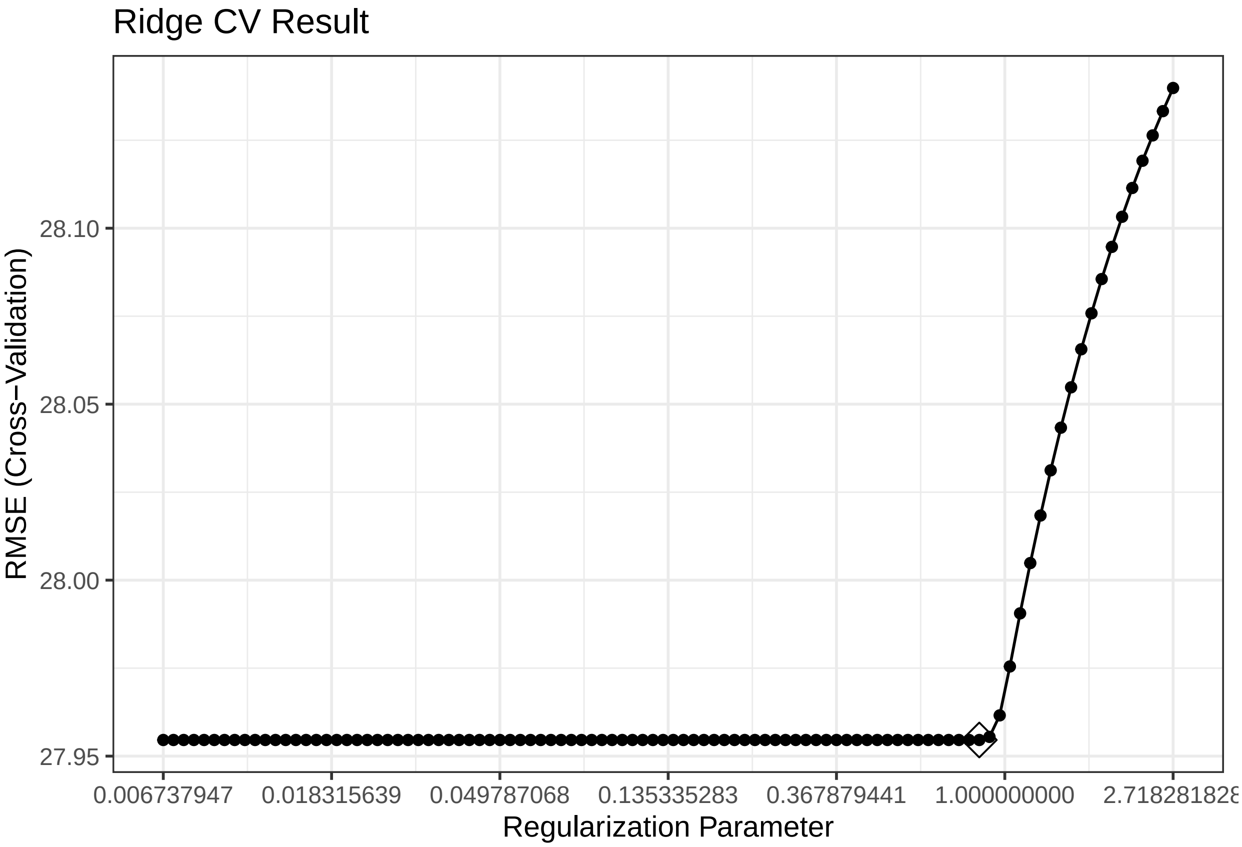
```
set.seed(2023)
ridge.fit <- train(train.x, train.y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 0,
                                          lambda = exp(seq(1, -5, length=100))),
                   trControl = ctrl1)

ridge.fit$bestTune
```

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

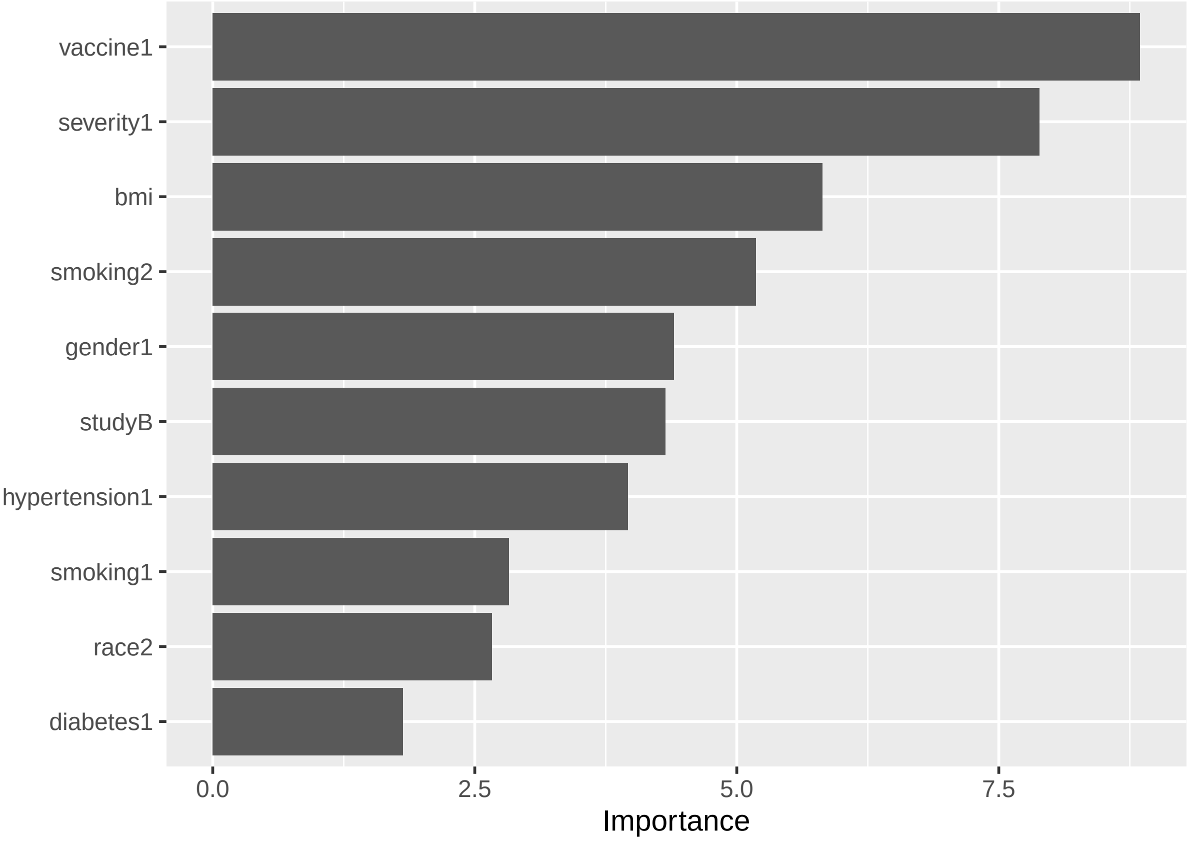
```
coef(ridge.fit$finalModel, s = ridge.fit$bestTune$lambda)
```

```
## 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)
```



1.1.4 Elastic Net

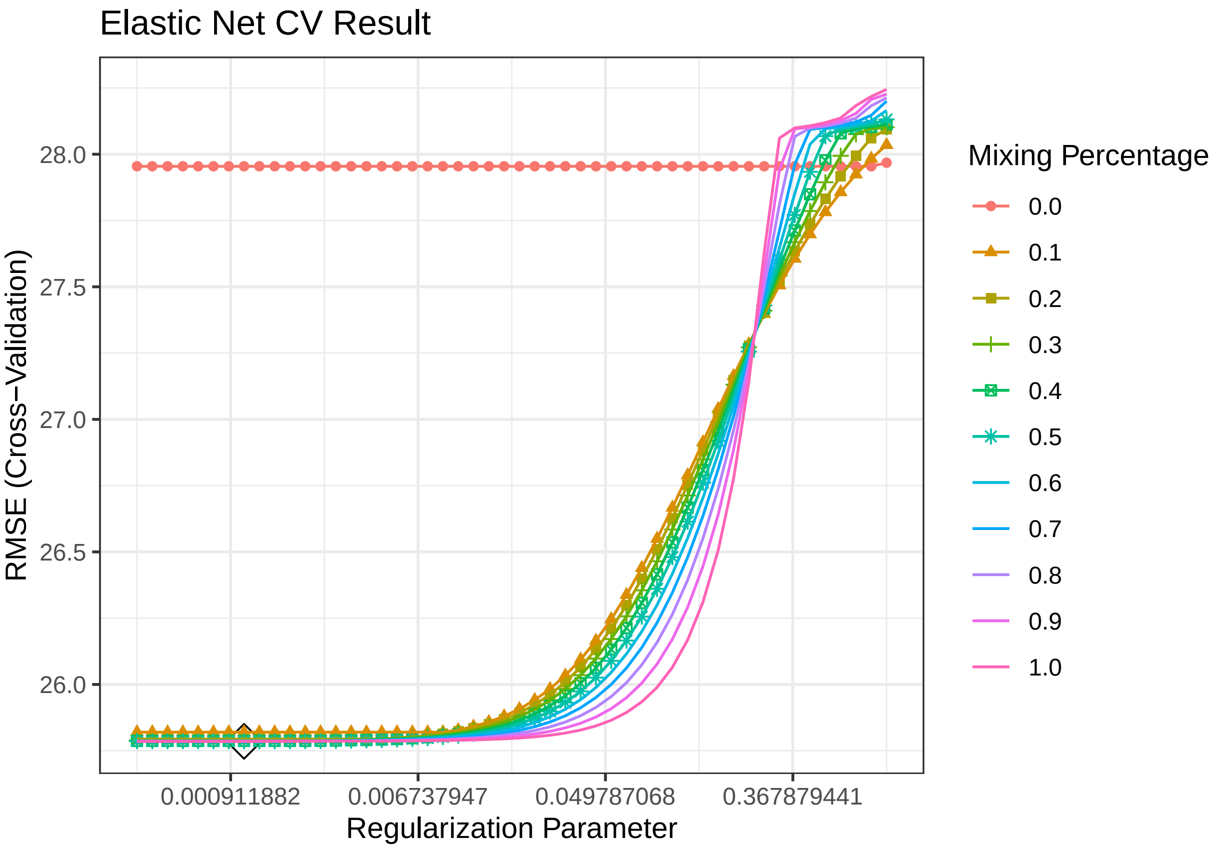
```
set.seed(2023)
enet.fit <- train(train.x, train.y,
                  method = "glmnet",
                  tuneGrid = expand.grid(
                    alpha = seq(0, 1, length = 11),
                    lambda = exp(seq(0, -8, length = 50))),
                  trControl = ctrl1)

enet.fit$bestTune
```

```
##     alpha      lambda
## 458   0.9 0.001051915
```

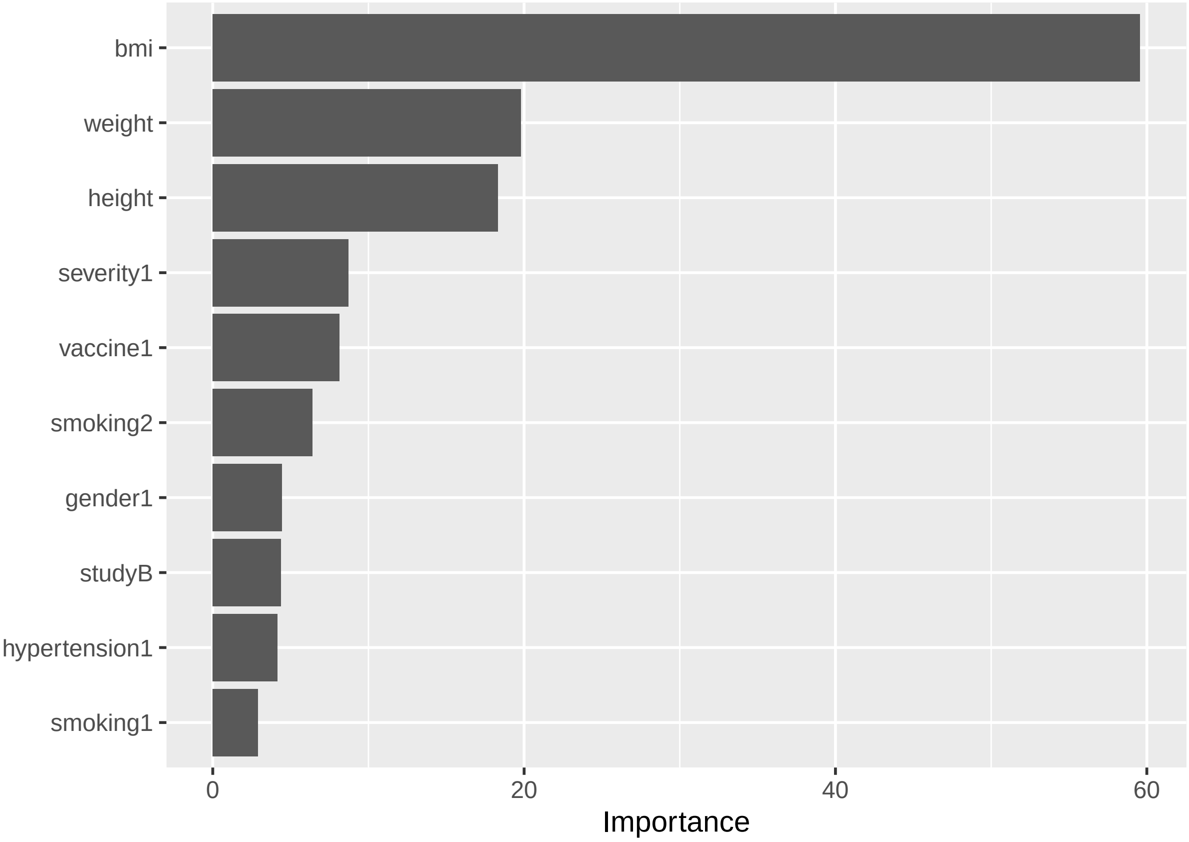
```
coef(enet.fit$finalModel, enet.fit$bestTune$lambda)
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)   -3.133363e+03
## age            1.156446e-01
## gender1       -4.443015e+00
## race2          2.194049e+00
## race3         -6.697538e-01
## race4         -1.151993e+00
## smoking1       2.902929e+00
## smoking2       6.403008e+00
## height          1.832705e+01
## weight         -1.978780e+01
## bmi             5.955488e+01
## hypertension1   4.156169e+00
## diabetes1      -1.161920e+00
## SBP            -7.786025e-02
## LDL            -4.215546e-02
## vaccine1       -8.149202e+00
## severity1       8.732536e+00
## studyB          4.370077e+00
## studyC         -6.790033e-01
```

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

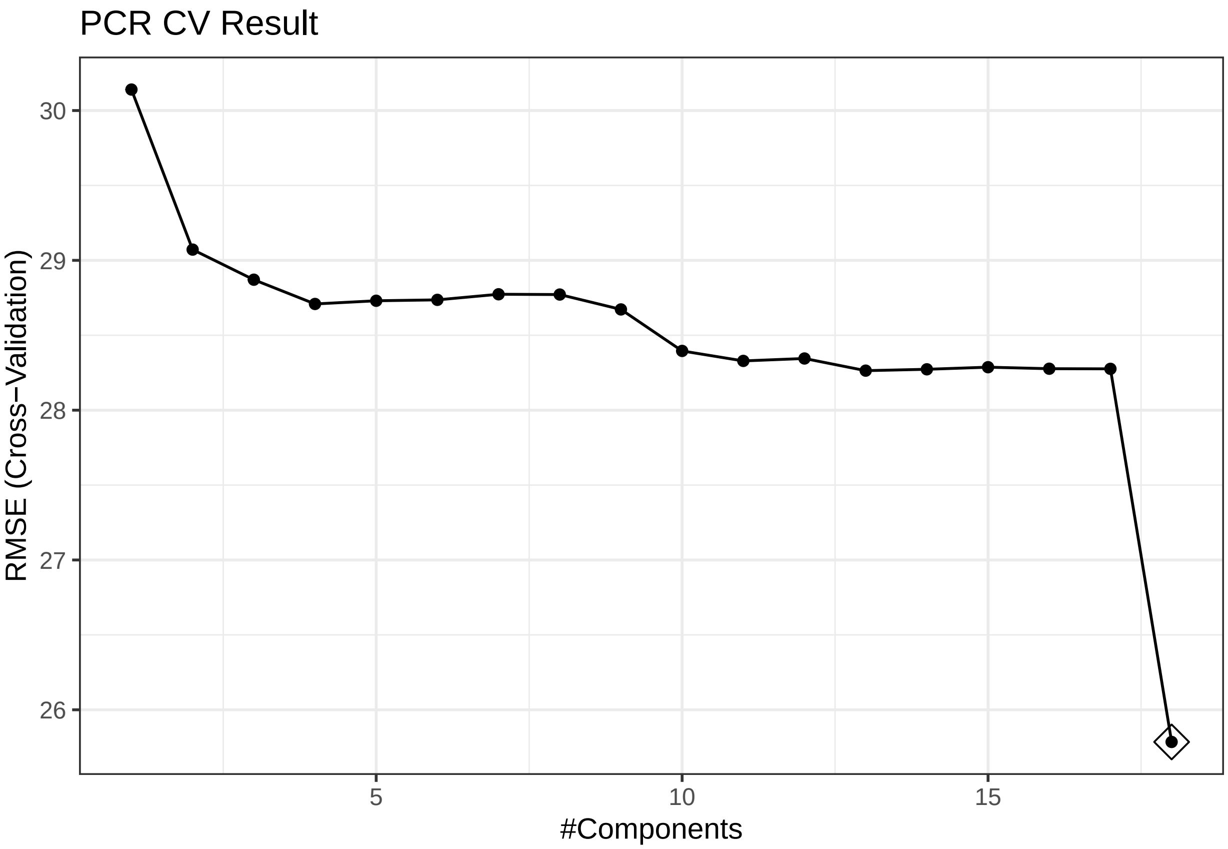


```
ggsave("./figure/enet_cv.jpeg", dpi = 500)

vip(enet.fit$finalModel)
```



1.1.5 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()
```



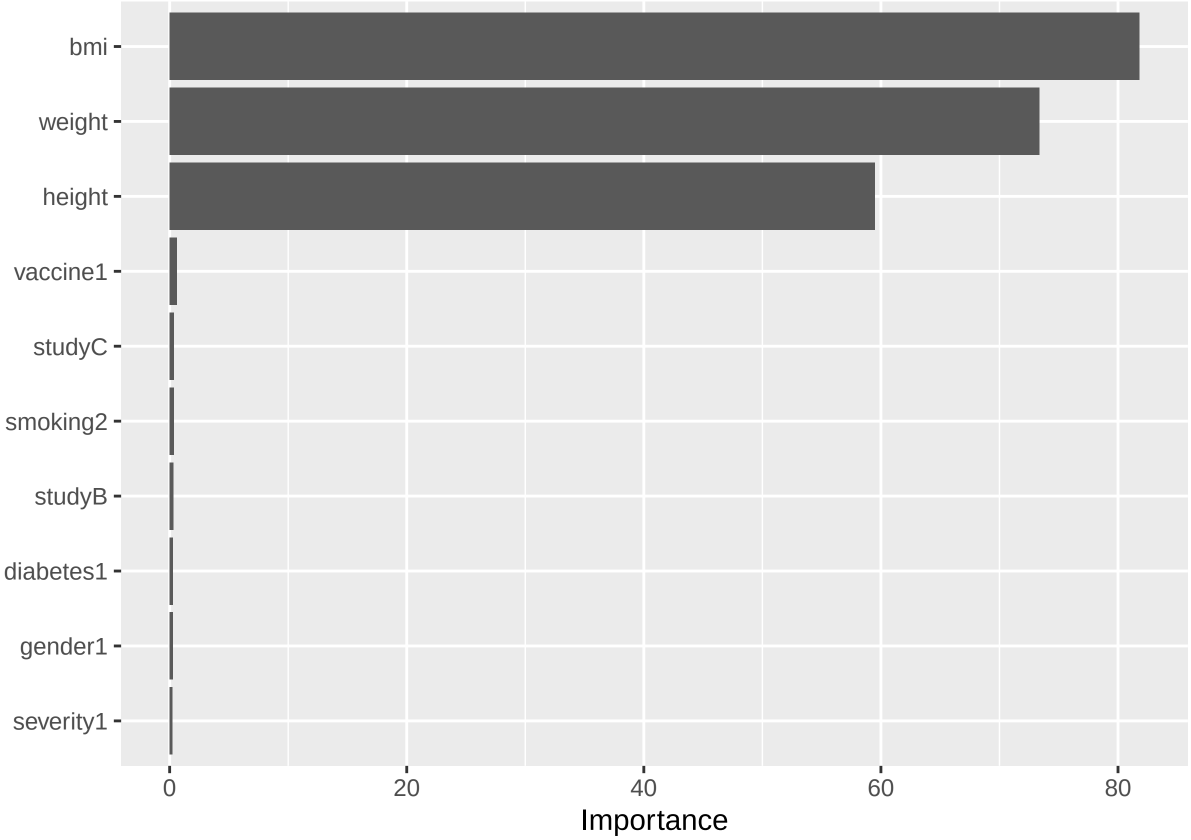
```
ggsave("./figure/pcr_cv.jpeg", dpi = 500)

pcr.fit$bestTune
```

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

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

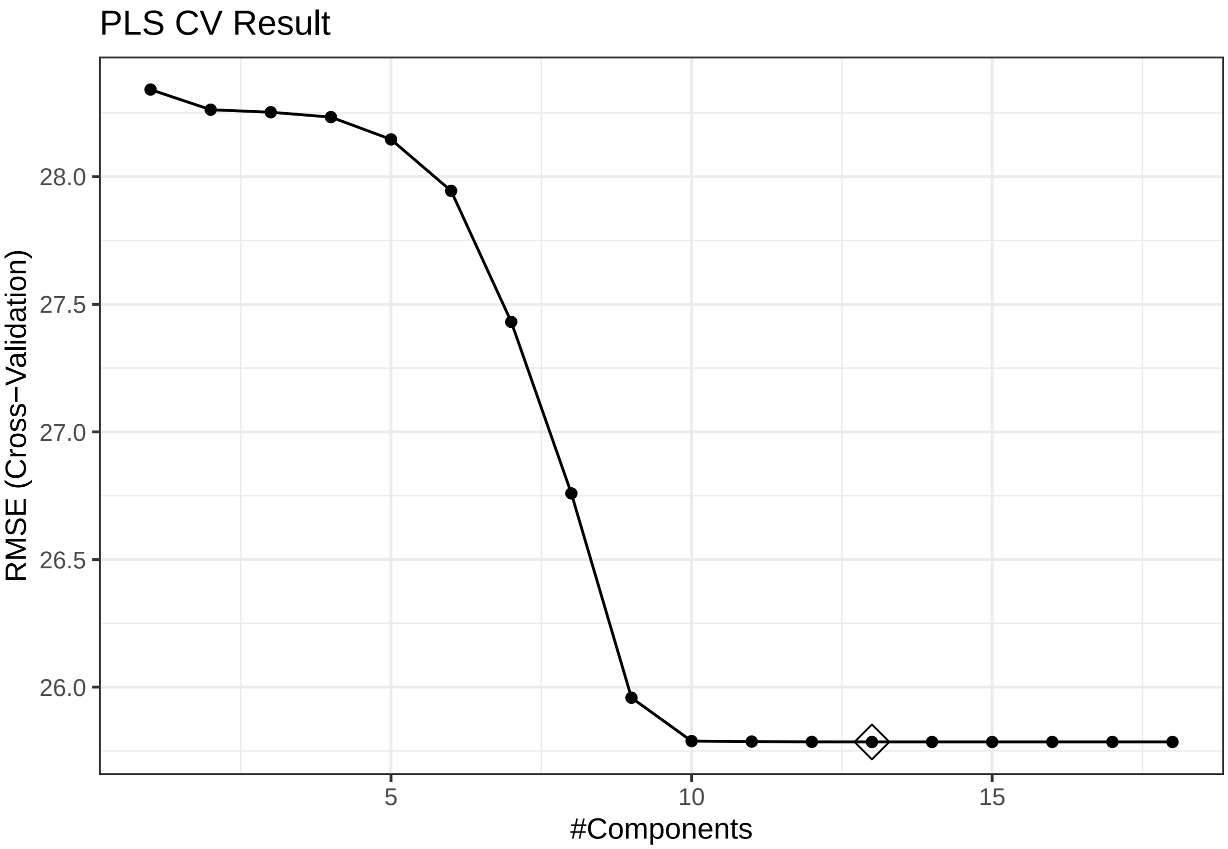
```
## , , 18 comps
##
##                    .outcome
## age               0.5252538
## gender1          -2.2221586
## race2             0.4563464
## race3            -0.2619635
## race4            -0.3476329
## smoking1          1.3205684
## smoking2          1.9344423
## height          112.6936931
## 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)
```



1.1.6 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()
```



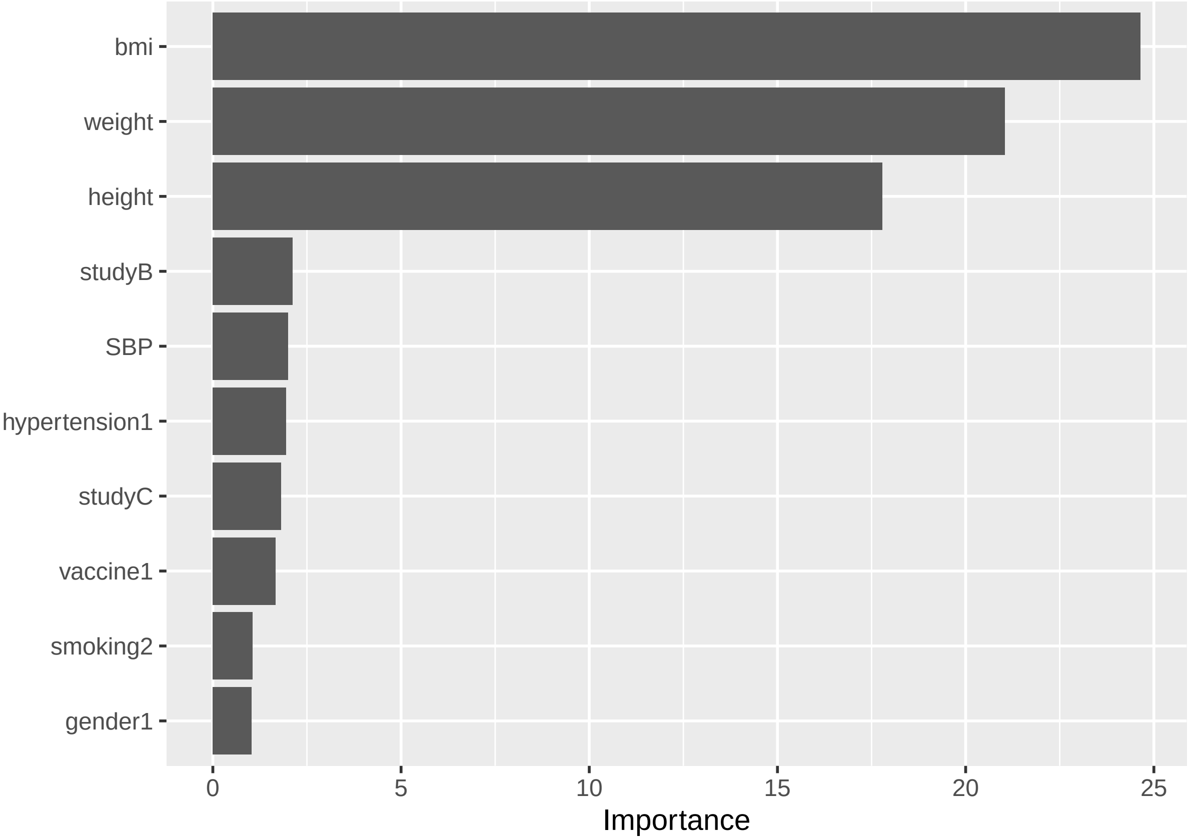
```
ggsave("./figure/pls_cv.jpeg", dpi = 500)

pls.fit$bestTune
```

```
##    ncomp
## 13    13
```

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

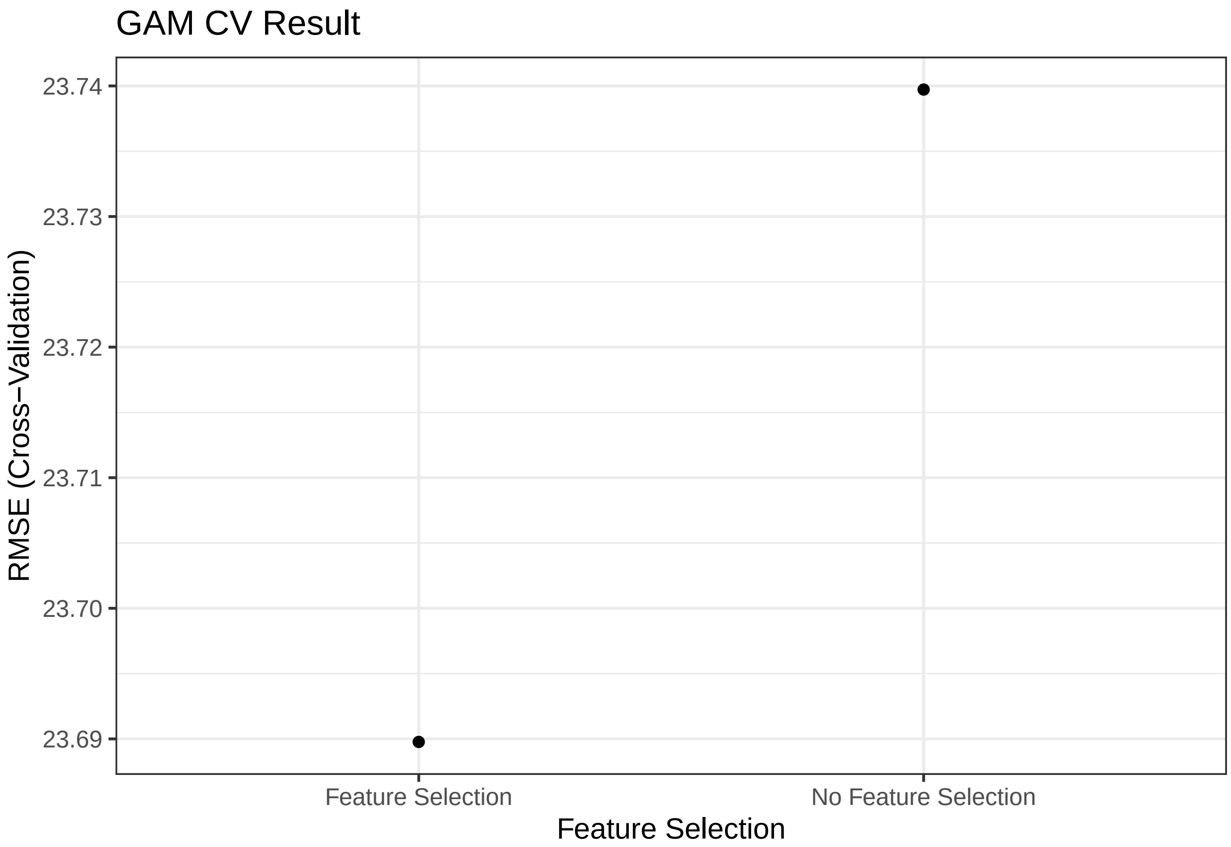
```
## , , 13 comps
##
##                    .outcome
## age               0.5253162
## gender1          -2.2224171
## race2             0.4564699
## race3            -0.2616135
## race4            -0.3472528
## smoking1          1.3206873
## smoking2          1.9344789
## height          112.6936914
## weight         -141.0001239
## bmi             165.1518926
## hypertension1     2.0811255
## diabetes1        -0.4187817
## SBP              -0.6356784
## LDL              -0.8377705
## vaccine1         -4.0025291
## severity1         2.5877989
## studyB           2.1374098
## studyC          -0.2730417
```

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



1.1.7 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()
```



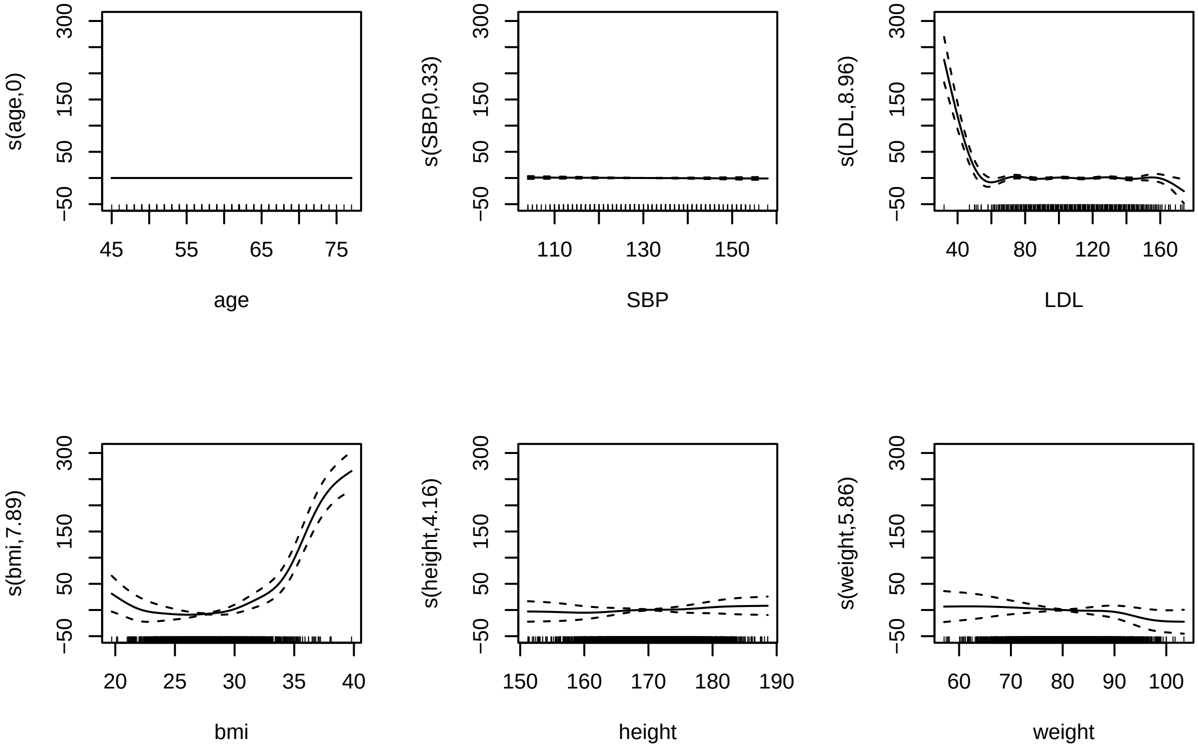
```r
ggsave("./figure/gam_cv.jpeg", dpi = 500)

gam.fit$bestTune
```

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

```r
# 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
```

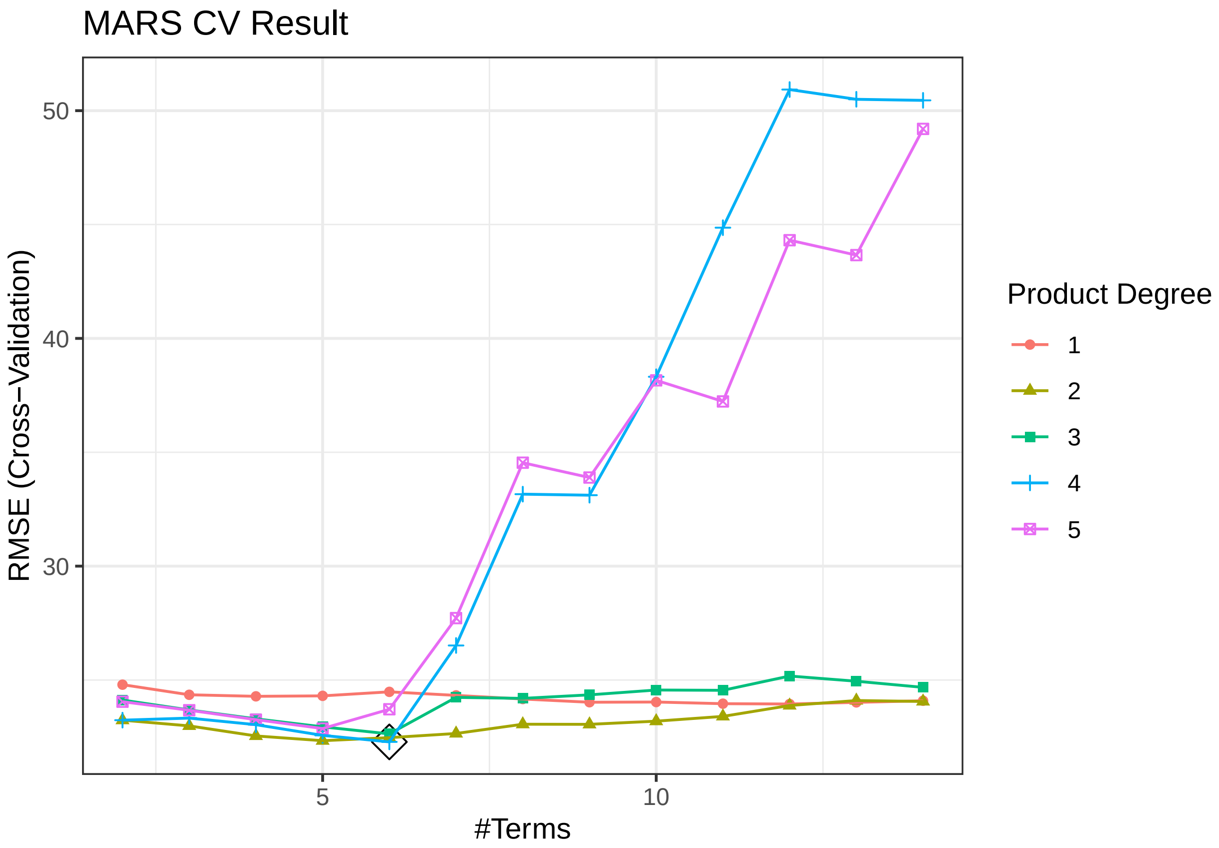
```r
par(mfrow=c(2, 3))
plot(gam.fit$finalModel)
```



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

1.1.8 Multivariate Adaptive Regression Splines (MARS)

```
mars.grid <- expand.grid(degree = 1:5,
                         nprune = 2:14)
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
## 44      6      4
```

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

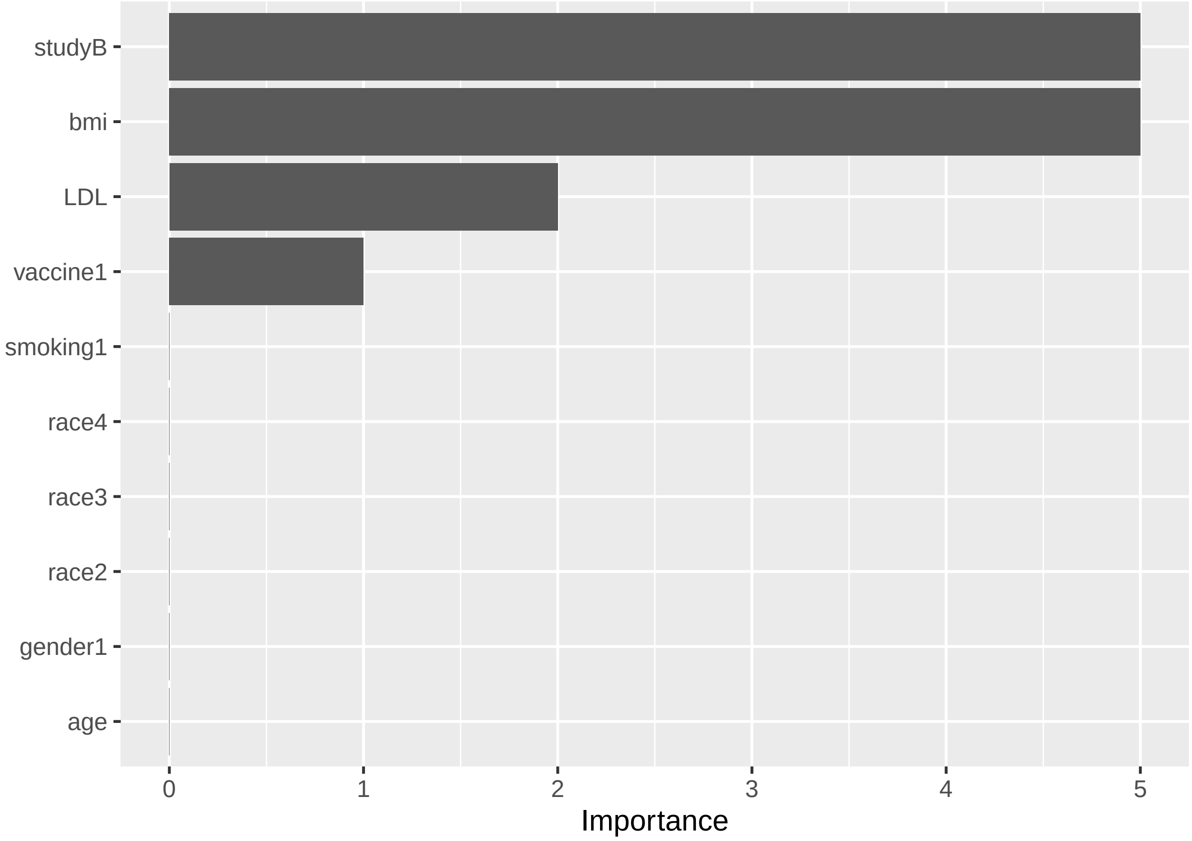
```
##                        (Intercept)                       h(31.7-bmi)
##                          19.332680                          3.707079
##                h(bmi-31.7) * studyB                       h(bmi-26.8)
##                          39.317768                          6.812493
## h(bmi-31.7) * h(LDL-115) * studyB                          vaccine1
##                          -1.309903                         -7.855865
```

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

```
## Call: earth(x=matrix[2900,18], y=c(40,34,31,50,3...), keepxy=TRUE, degree=4,
##             nprune=6)
##
##                                    coefficients
## (Intercept)                           19.332680
## vaccine1                              -7.855865
## h(bmi-26.8)                            6.812493
## h(31.7-bmi)                            3.707079
## h(bmi-31.7) * studyB                  39.317768
## h(bmi-31.7) * h(LDL-115) * studyB     -1.309903
##
```

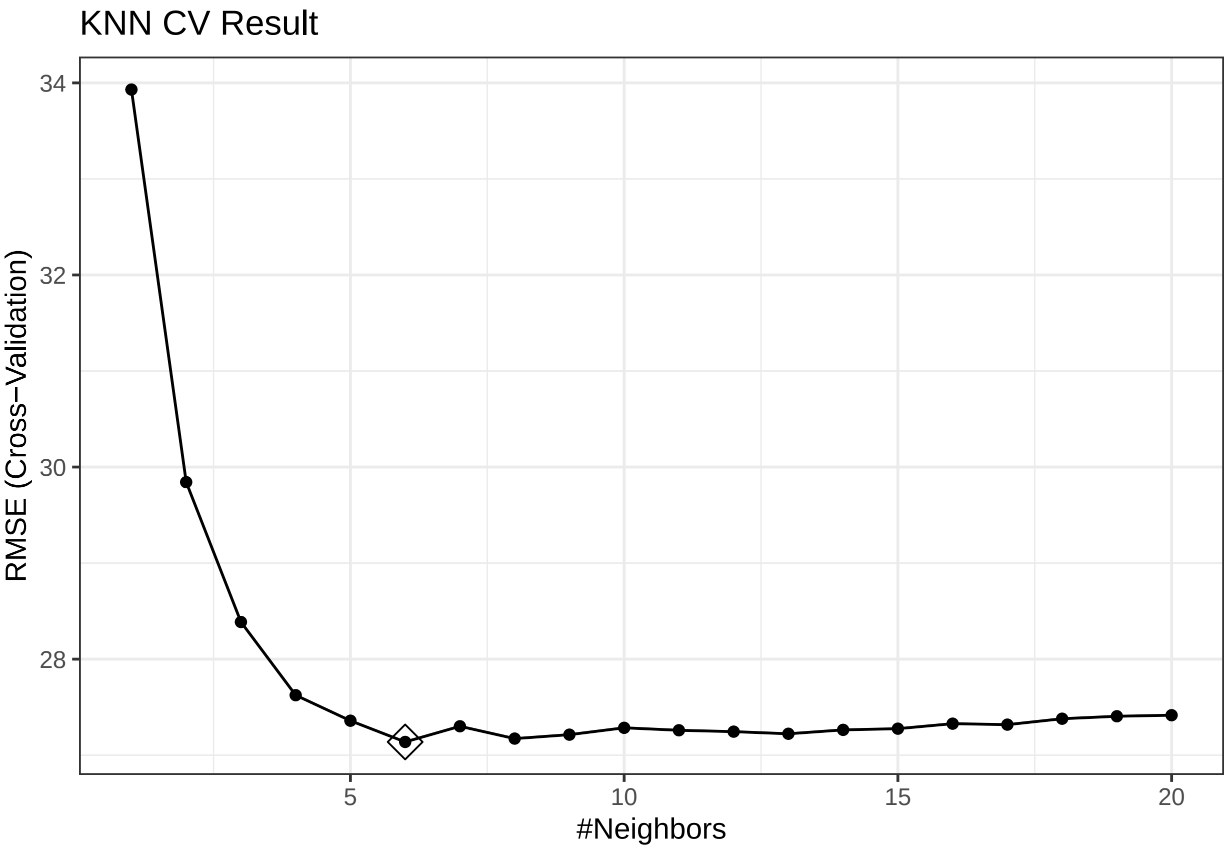
```
## Selected 6 of 26 terms, and 4 of 18 predictors (nprune=6)
## Termination condition: Reached nk 37
## Importance: bmi, studyB, LDL, vaccine1, age-unused, gender1-unused, ...
## Number of terms at each degree of interaction: 1 3 1 1
## GCV 474.177    RSS 1362340    GRSq 0.4906252    RSq 0.4950084
```

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



1.1.9 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
```

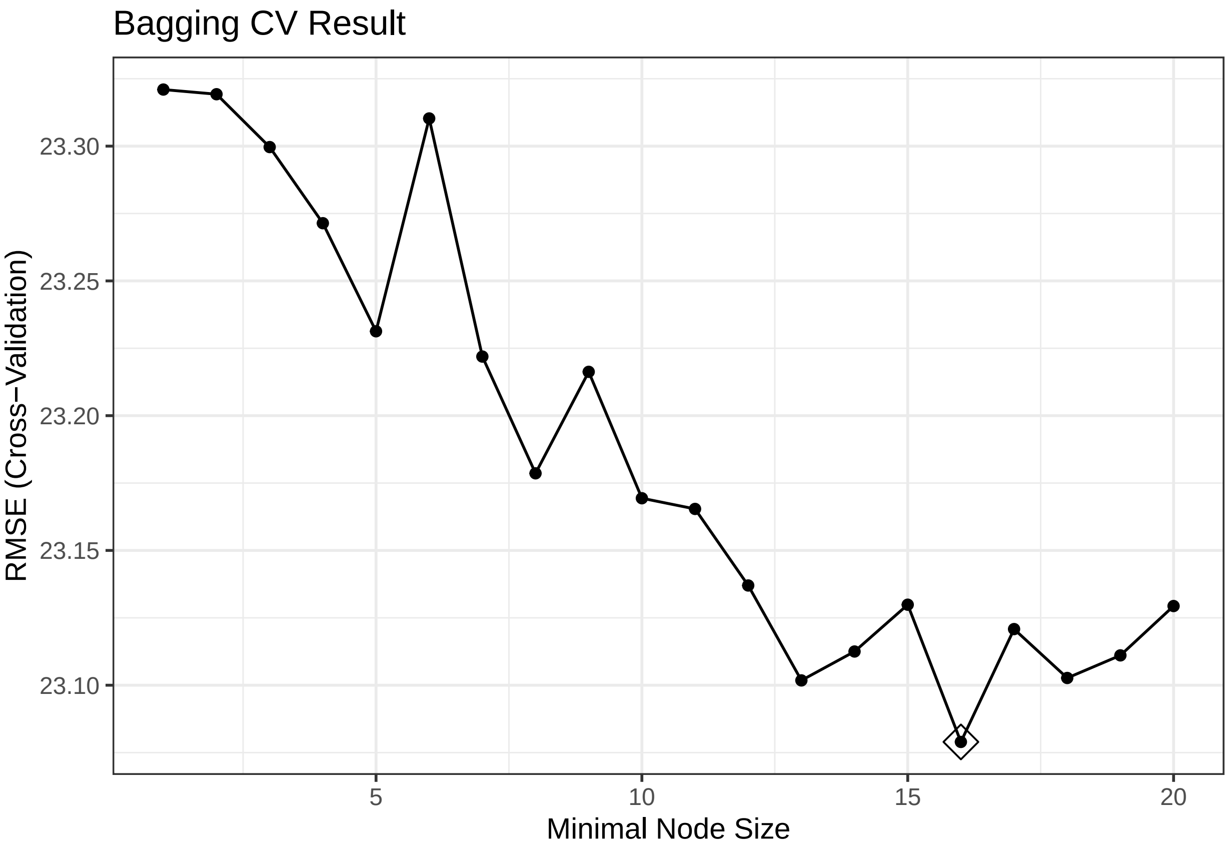
```
##   k
## 6 6
```

1.1.10 Bagging

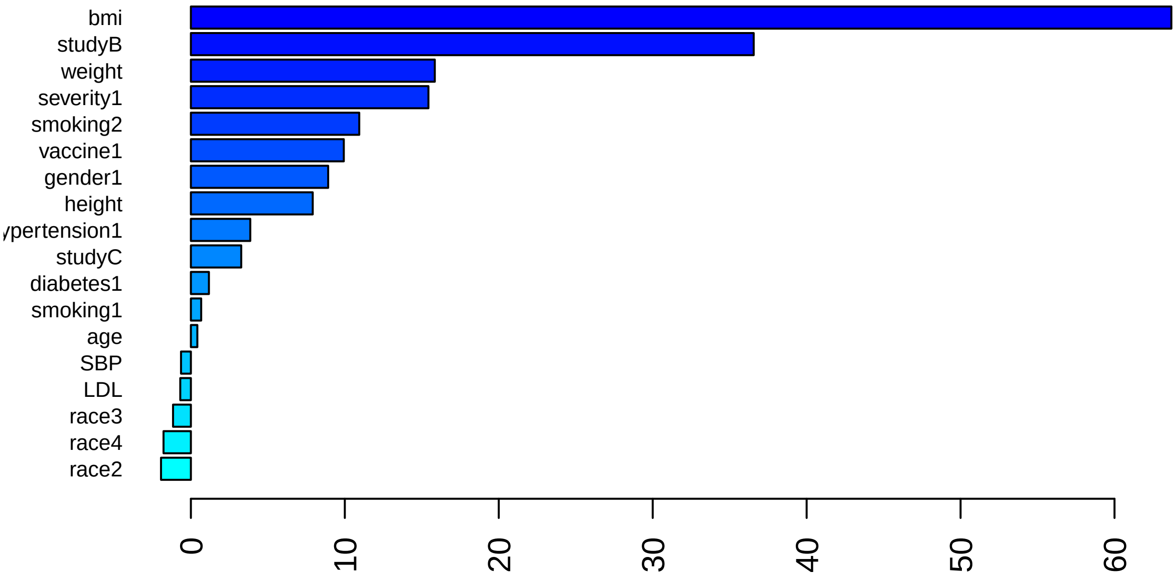
```
bag.grid <- expand.grid(mtry = ncol(train.x),
                        splitrule = "variance",
                        min.node.size = 1:20)

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
## 16   18  variance            16
```

```
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)))
```



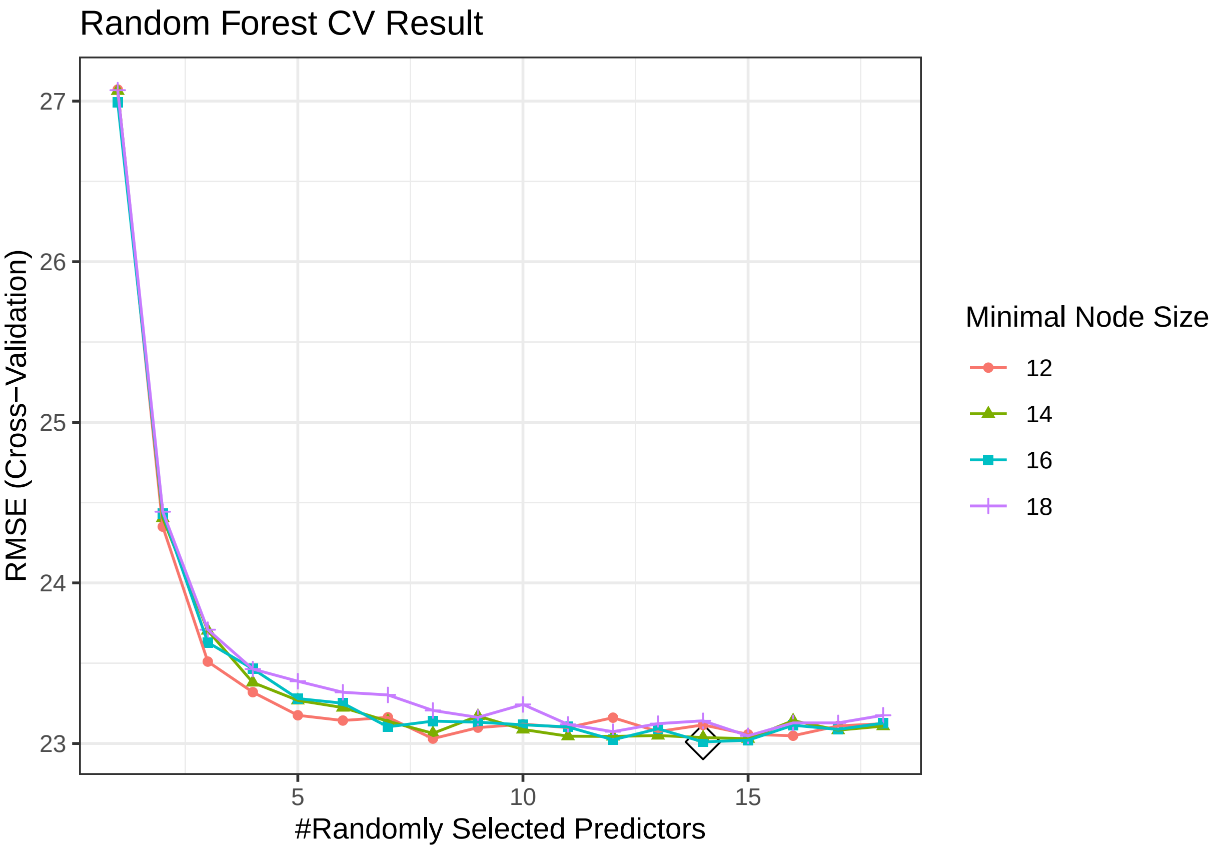
```
# p1 <- pdp::partial(
#   bag.fit,
#   pred.var = "Lot_Area",
#   grid.resolution = 20
#   ) %>%
#   autoplot()
# p2 <- pdp::partial(
#   bag.fit,
#   pred.var = "Lot_Frontage",
#   grid.resolution = 20
#   ) %>%
#   autoplot()
# gridExtra::grid.arrange(p1, p2, nrow = 1)
```

1.1.11 Random Forest

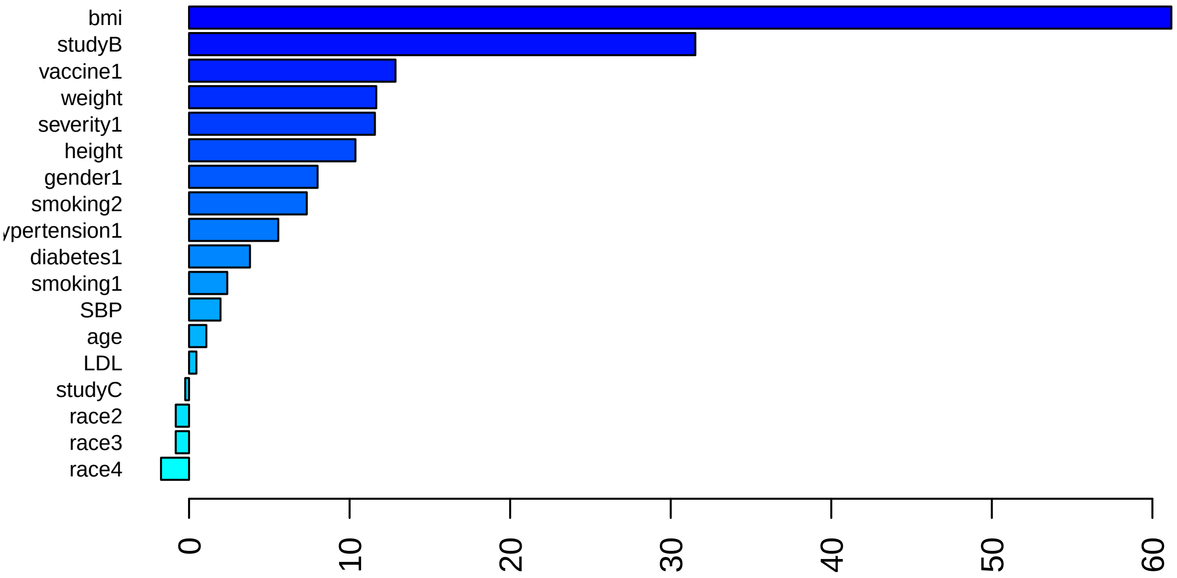
```
rf.grid <- expand.grid(mtry = 1:ncol(train.x),
                       splitrule = "variance",
                       min.node.size = seq(12, 18, by = 2))
set.seed(2023)
rf.fit <- train(train.x,
                train.y,
                method = "ranger",
                tuneGrid = rf.grid,
                trControl = ctrl1)

rf.fit$bestTune
```

```
##    mtry splitrule min.node.size
## 55   14  variance            16
```

```
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)))
```



1.1.12 Boosting

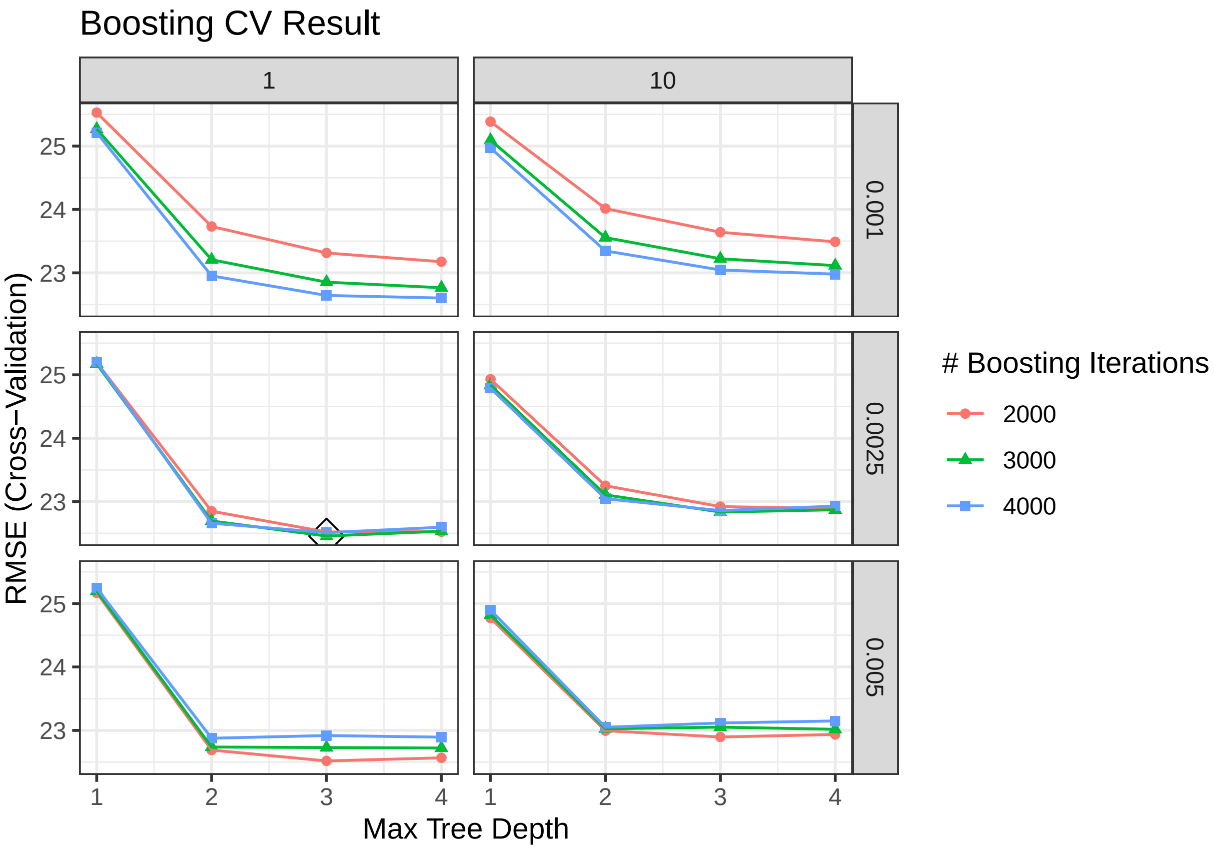
```
set.seed(2023)
bst.grid <- expand.grid(n.trees = c(2000, 3000, 4000),
                        interaction.depth = 1:4,
                        shrinkage = c(0.001, 0.0025, 0.005),
                        n.minobsinnode = c(1, 10))

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
## 38    3000                 3    0.0025              1
```

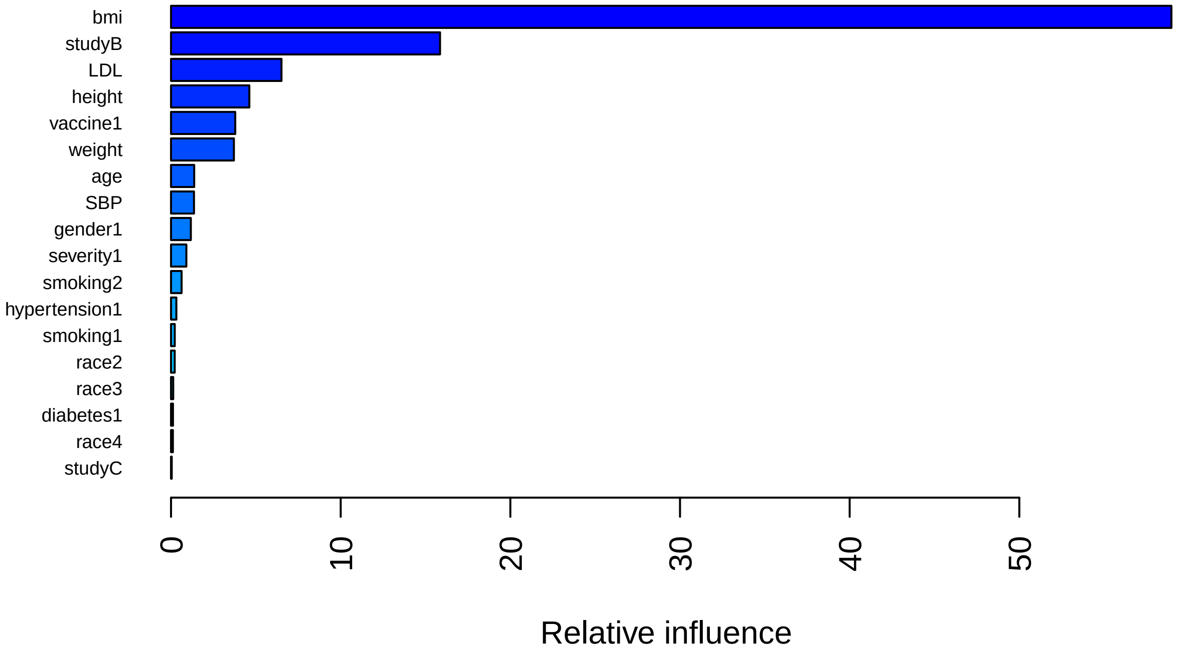
```
bst.fit$finalModel
```

```
## A gradient boosted model with gaussian loss function.
## 3000 iterations were performed.
## There were 18 predictors of which 18 had non-zero influence.
```

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



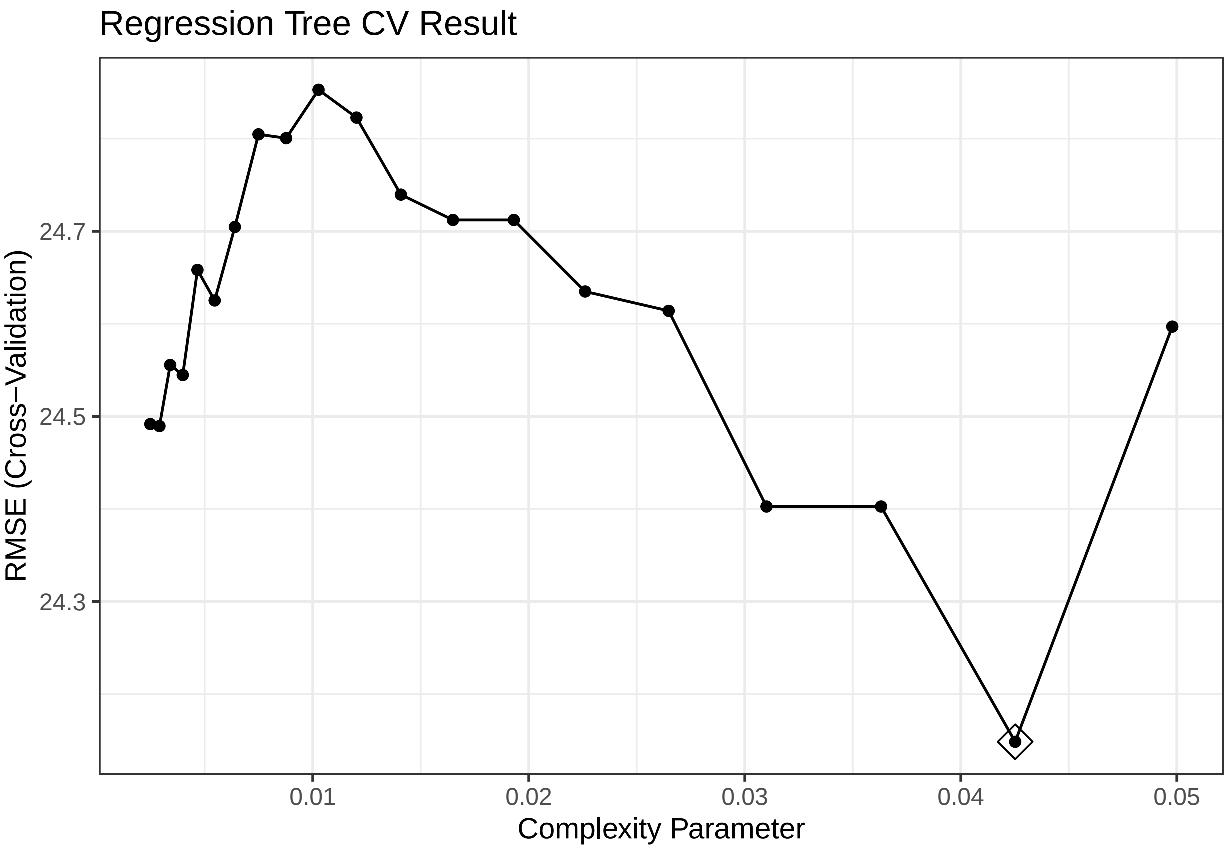
```r
ggsave("./figure/boosting_cv.jpeg", dpi = 500)

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



```
##                                    var      rel.inf
## bmi                     bmi 58.96414769
## studyB               studyB 15.85619711
## LDL                     LDL  6.50952120
## height               height  4.61582502
## vaccine1           vaccine1  3.78444689
## weight               weight  3.70589114
## age                     age  1.36637699
## SBP                     SBP  1.35342972
## gender1             gender1  1.16795535
## severity1         severity1  0.90341683
## smoking2           smoking2  0.62280498
## hypertension1 hypertension1  0.31496695
## smoking1           smoking1  0.22066905
## race2                 race2  0.21684477
## race3                 race3  0.12927099
## diabetes1         diabetes1  0.11547616
## race4                 race4  0.10962368
## studyC               studyC  0.04313547
```

1.1.13 Regression Trees

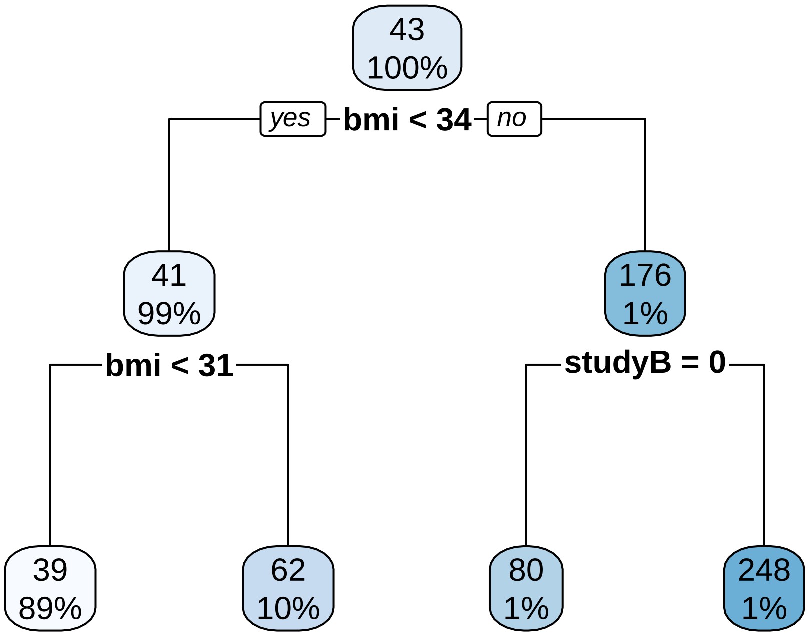
```
rpart.grid <- expand.grid(cp = exp(seq(-6,-3, length = 20)))
set.seed(2023)
rpart.fit1 <- train(train.x,
                   train.y,
                   method = "rpart",
                   tuneGrid = rpart.grid,
                   trControl = ctrl1)

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



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

rpart.fit1$bestTune
```

```
##            cp
## 19 0.04251515
```

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

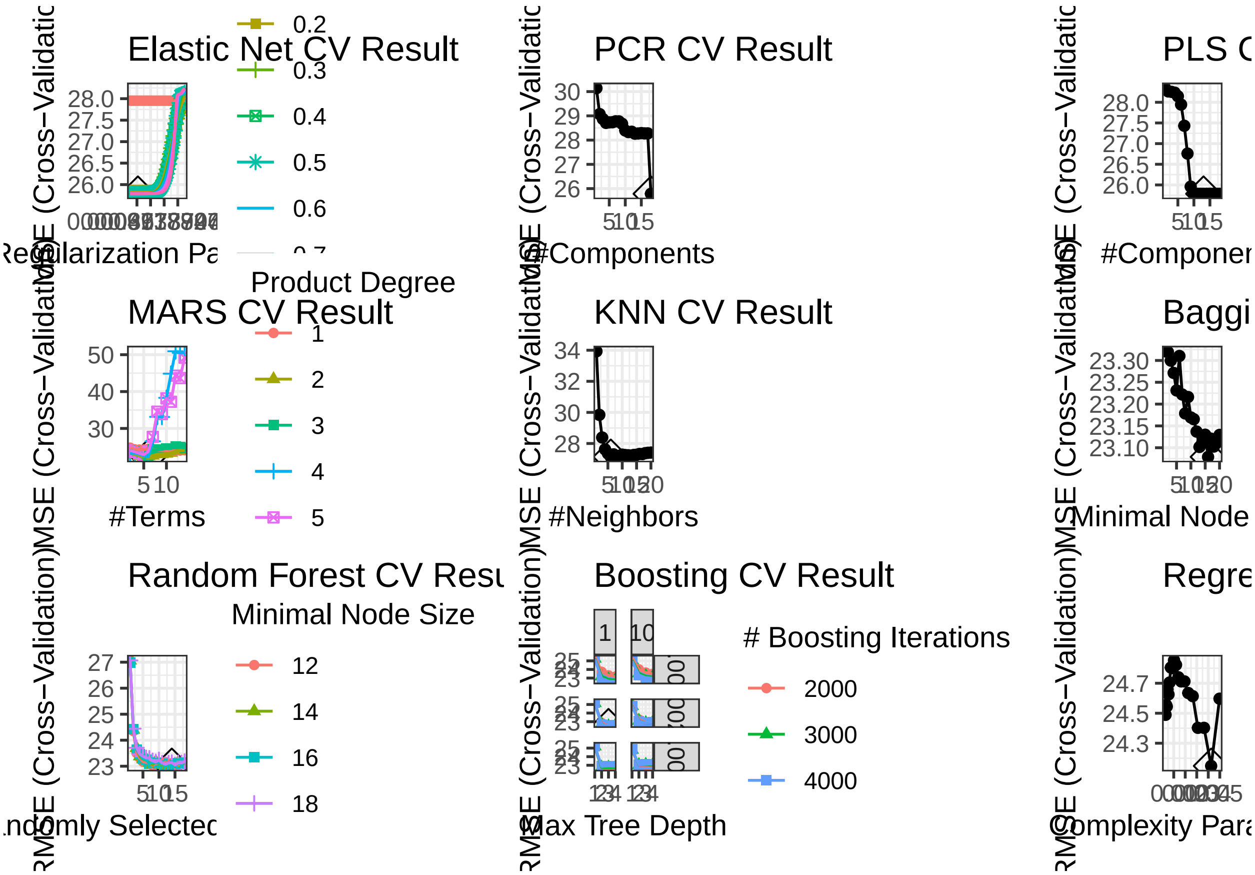


```
jpeg("./figure/rpart1.jpeg", width = 8, height = 6, units="in", res=500)
rpart.plot(rpart.fit1$finalModel)
dev.off()
```

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

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

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

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

pcr <- ggplot(pcr.fit, highlight = TRUE) +
  labs(title  ="PCR CV Result") +
  theme_bw()

pls <- ggplot(pls.fit, highlight = TRUE) +
  labs(title  ="PLS CV Result") +
  theme_bw()

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

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

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

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

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

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

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

p <- wrap_plots(enet, pcr,
           pls,
           mars, knn,
           bagging, rf, boosting, tree,
           ncol = 3)
p
```



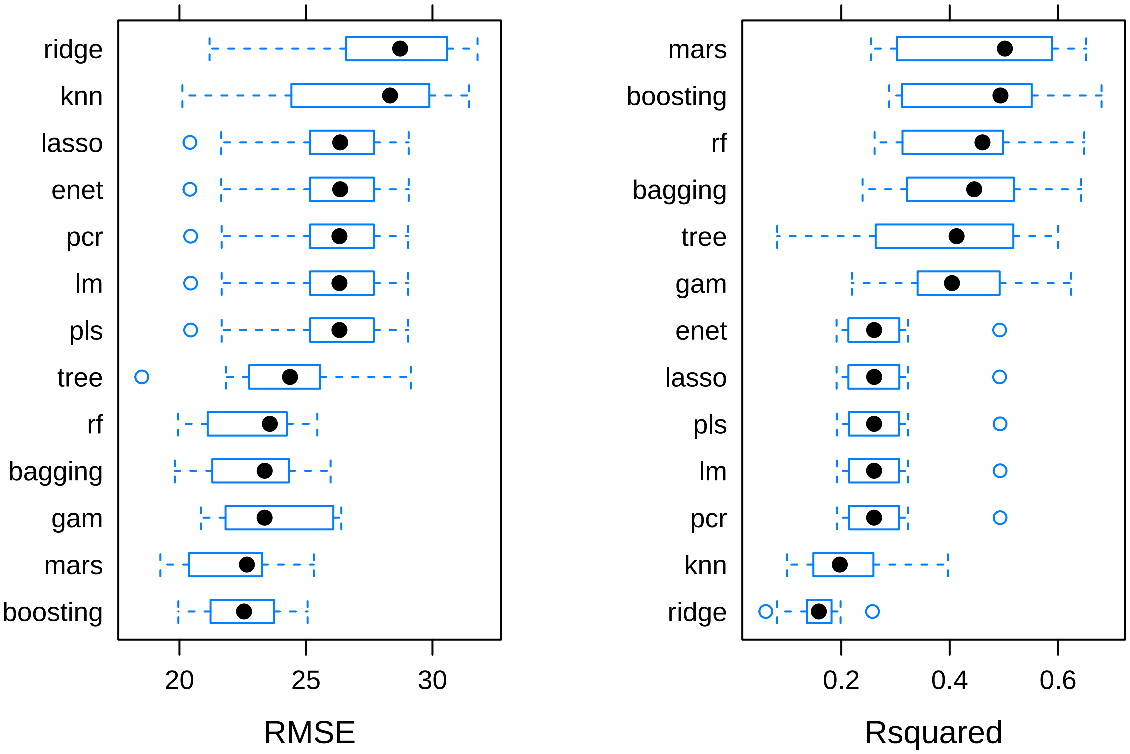
```r
ggsave(plot = p, filename = "./primary_cv.jpeg", dpi = 500)
```

1.2 Model Selection

```r
set.seed(2023)
resamp1 <- 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(resamp1)
```

```
##
## Call:
## summary.resamples(object = resamp1)
##
## Models: lm, lasso, ridge, enet, pcr, pls, gam, mars, knn, bagging, rf, boosting, tree
## Number of resamples: 10
## 
## MAE 
##              Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lm       15.54483 15.80758 16.63529 16.59842 17.13204 18.12333    0
## lasso    15.51069 15.78658 16.61245 16.57052 17.09219 18.09015    0
## ridge    15.34004 16.62387 16.79935 16.84047 17.23997 18.17959    0
## enet     15.51026 15.78694 16.61217 16.57069 17.09223 18.09088    0
## pcr      15.54483 15.80758 16.63529 16.59842 17.13204 18.12333    0
## pls      15.54482 15.80753 16.63528 16.59840 17.13208 18.12332    0
## gam      14.60392 14.76502 15.40409 15.42678 15.78762 17.02963    0
## mars     13.99273 14.46938 14.98372 14.92537 15.32753 15.89972    0
## knn      14.43602 16.28400 16.79135 16.77166 17.45629 18.38966    0
## bagging  14.22336 14.45336 15.13841 15.11634 15.80201 15.93486    0
## rf       14.25716 14.38885 15.08343 15.06824 15.64831 16.12144    0
## boosting 13.80473 14.23345 14.66386 14.63306 15.06668 15.39472    0
## tree     13.85891 15.27840 15.72822 15.63398 16.07374 16.57189    0
## 
## RMSE 
##              Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lm       20.44180 25.16612 26.32308 25.78528 27.58385 29.03646    0
## lasso    20.41446 25.16779 26.35443 25.78553 27.58286 29.05994    0
## ridge    21.18921 26.76934 28.72409 27.95459 30.39855 31.78080    0
## enet     20.41395 25.16792 26.35390 25.78540 27.58280 29.06018    0
## pcr      20.44180 25.16612 26.32308 25.78528 27.58385 29.03646    0
## pls      20.44179 25.16611 26.32305 25.78526 27.58386 29.03644    0
## gam      20.84135 22.00149 23.36475 23.68977 25.89070 26.39798    0
## mars     19.24575 20.68200 22.66604 22.28254 23.25541 25.30793    0
## knn      20.11678 25.01933 28.32298 27.13762 29.65682 31.44427    0
## bagging  19.81748 21.50728 23.36814 23.07896 24.24524 25.97220    0
## rf       19.95074 21.25601 23.57112 23.00997 24.19683 25.45015    0
## boosting 19.95593 21.25085 22.55222 22.45723 23.72734 25.06383    0
## tree     18.51152 23.03638 24.36887 24.14847 25.41831 29.14098    0
## 
## Rsquared 
##                Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm       0.19215021 0.2201628 0.2605519 0.2764092 0.3001496 0.4930552    0
## lasso    0.19133277 0.2196625 0.2606385 0.2763222 0.3004686 0.4921785    0
## ridge    0.06069200 0.1374835 0.1585905 0.1552980 0.1812584 0.2575556    0
## enet     0.19132111 0.2196526 0.2606417 0.2763214 0.3004556 0.4921930    0
## pcr      0.19215021 0.2201628 0.2605519 0.2764092 0.3001496 0.4930552    0
## pls      0.19215072 0.2201634 0.2605543 0.2764103 0.3001491 0.4930584    0
## gam      0.21948254 0.3453667 0.4042745 0.4084093 0.4864782 0.6243131    0
## mars     0.25523324 0.3327875 0.5019111 0.4666544 0.5812142 0.6518673    0
## knn      0.09988269 0.1495740 0.1971881 0.2119237 0.2570144 0.3966191    0
## bagging  0.23926439 0.3254906 0.4452885 0.4315122 0.5070804 0.6426996    0
## rf       0.26142227 0.3159714 0.4605716 0.4347653 0.4923031 0.6483873    0
## boosting 0.28848789 0.3516448 0.4937148 0.4718739 0.5503996 0.6802951    0
## tree     0.08170188 0.2889918 0.4127453 0.3758431 0.4958400 0.6000578    0
```

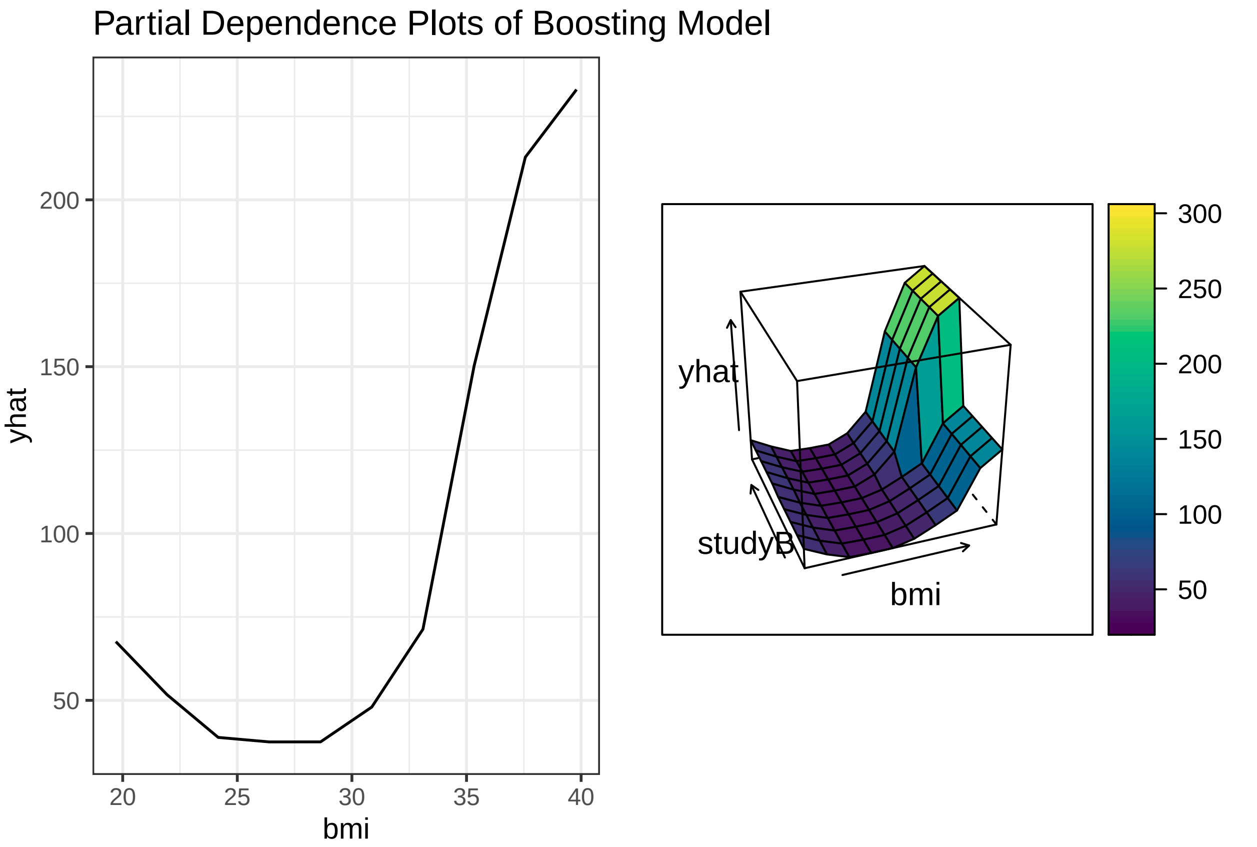
```r
p1=bwplot(resamp1, metric = "RMSE")
p2=bwplot(resamp1, metric = "Rsquared")
grid.arrange(p1, p2 ,ncol=2)
```



```
jpeg("./figure/resample1.jpeg", width = 8, height=6, units="in", res=500)
p1=bwplot(resamp1, metric = "RMSE")
p2=bwplot(resamp1, metric = "Rsquared")
grid.arrange(p1, p2, ncol=2)
dev.off()
```

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

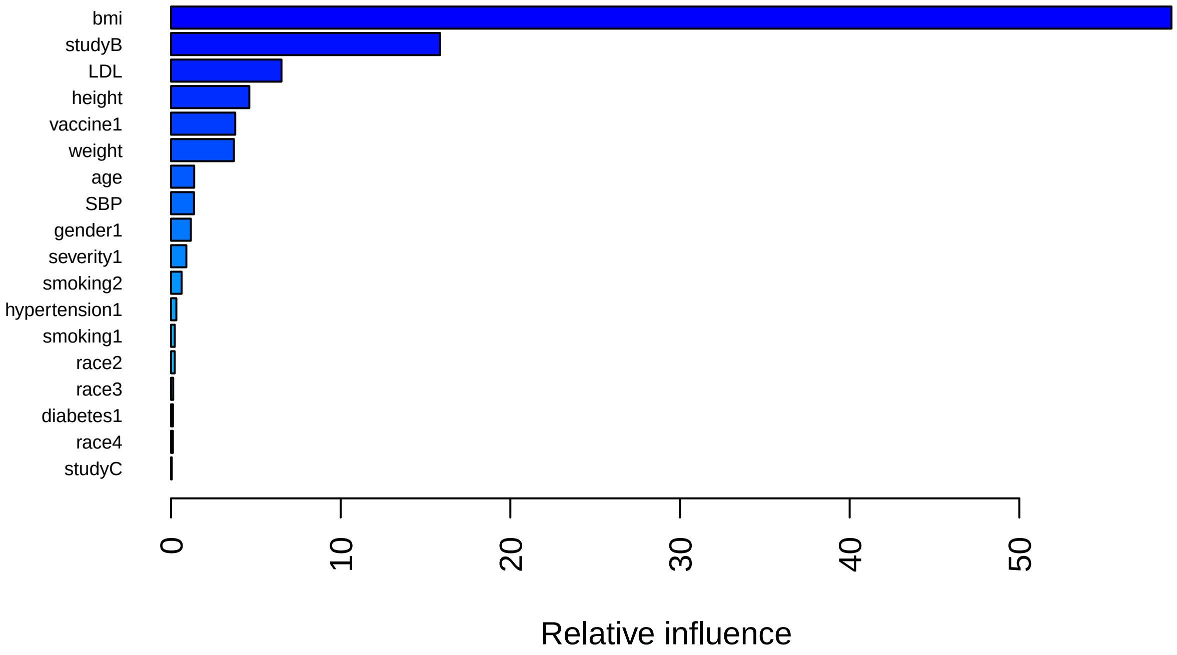
```
p1<- pdp::partial(bst.fit, pred.var = c("bmi"), grid.resolution = 10) %>% autoplot() +
  theme_bw()+
  labs(title = "Partial Dependence Plots of Boosting Model")

p2 <-pdp::partial(bst.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()

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

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

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



```
##                         var     rel.inf
## bmi                     bmi 58.96414769
## studyB               studyB 15.85619711
## LDL                     LDL  6.50952120
## height               height  4.61582502
## vaccine1           vaccine1  3.78444689
## weight               weight  3.70589114
## age                     age  1.36637699
## SBP                     SBP  1.35342972
## gender1             gender1  1.16795535
## severity1         severity1  0.90341683
## smoking2           smoking2  0.62280498
## hypertension1 hypertension1  0.31496695
## smoking1           smoking1  0.22066905
## race2                 race2  0.21684477
## race3                 race3  0.12927099
## diabetes1         diabetes1  0.11547616
## race4                 race4  0.10962368
## studyC               studyC  0.04313547
```

```
jpeg("./figure/bst.importance.jpeg", width = 8, height=6, units="in", res=500)
summary(bst.fit$finalModel, las = 2, cBars = ncol(train.x), cex.names = 0.6)
```

```
##                         var     rel.inf
## bmi                     bmi 58.96414769
## studyB               studyB 15.85619711
## LDL                     LDL  6.50952120
## height               height  4.61582502
## vaccine1           vaccine1  3.78444689
## weight               weight  3.70589114
## age                     age  1.36637699
## SBP                     SBP  1.35342972
## gender1             gender1  1.16795535
## severity1         severity1  0.90341683
## smoking2           smoking2  0.62280498
## hypertension1 hypertension1  0.31496695
## smoking1           smoking1  0.22066905
## race2                 race2  0.21684477
## race3                 race3  0.12927099
## diabetes1         diabetes1  0.11547616
## race4                 race4  0.10962368
## studyC               studyC  0.04313547
```

```
dev.off()
```

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

1.3 Training / Testing Error

```
# boosting
# training error
bst.train.pred <- predict(bst.fit, newdata = train.x)
RMSE(bst.train.pred, train.y)
```

```
## [1] 19.12669
```

```
# test error
bst.test.pred <- predict(bst.fit, newdata = test.x)
RMSE(bst.test.pred, test.y)
```

```
## [1] 22.30385
```

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

```
## [1] 21.67424
```

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

```
## [1] 23.63736
```