

final_proj

Me

Today

```
# import and subset data
load("./recovery.RData")
set.seed(5095)
dat.1 = dat[sample(1:10000, 2000),]

set.seed(5296)
dat.2 = dat[sample(1:10000, 2000),]

dat.all = rbind(dat.1, dat.2)%>%
  unique.array()

# transform variables as needed
dat1 = dat.all[2:16] %>%
  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))

# transform into matrix
dat2 = model.matrix(recovery_time ~ ., dat1)[ , -1]

# split data into training set and test set
set.seed(1)
trainRows = createDataPartition(y = dat1$recovery_time, p = 0.8, list = FALSE)

# extract training data
x.train = dat2[trainRows,]
y.train = dat1$recovery_time[trainRows]

# correlation plot
x_cor = dat2[trainRows, c("age", "height", "weight", "bmi", "SBP", "LDL")]
png(height=1800, width=1800, units = "px", file="corrplot.png", res = 200)
corrplot::corrplot(cor(x_cor), method = "circle", type = "full")
dev.off()

## pdf
## 2
```

```

#library("PerformanceAnalytics")
#chart.Correlation(x, histogram=TRUE, pch=19)

x.test = dat2[-trainRows,]
y.test = dat1$recovery_time[-trainRows]

# plot numeric variables
theme1 <- trellis.par.get()
theme1$plot.symbol$col <- rgb(.2, .4, .2, .5)
theme1$plot.symbol$pch <- 16
theme1$plot.line$col <- rgb(.8, .1, .1, 1)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <- rgb(.0, .2, .6, .2)
trellis.par.set(theme1)

png(height=1800, width=1800, units = "px", file="featureplot.png", res = 200)
p1 = featurePlot(x.train[,c(1,8, 9, 10, 13, 14)], y.train, plot = "scatter", labels = c("", "Y"),
                 type = c("p"), layout = c(3, 2))
p1
dev.off()

```

```

## pdf
## 2

```

```

# plot categorical variables
p2 = ggplot(dat1, aes(y = recovery_time, x = gender))+
  geom_boxplot()
p3 = ggplot(dat1, aes(y = recovery_time, x = race))+
  geom_boxplot()
p4 = ggplot(dat1, aes(y = recovery_time, x = smoking))+
  geom_boxplot()
p5 = ggplot(dat1, aes(y = recovery_time, x = hypertension))+
  geom_boxplot()
p6 = ggplot(dat1, aes(y = recovery_time, x = diabetes))+
  geom_boxplot()
p7 = ggplot(dat1, aes(y = recovery_time, x = vaccine))+
  geom_boxplot()
p8 = ggplot(dat1, aes(y = recovery_time, x = severity))+
  geom_boxplot()
p9 = ggplot(dat1, aes(y = recovery_time, x = study))+
  geom_boxplot()

arrange = ggarrange(p2, p3, p4, p5, p6, p7, p8, p9, ncol = 4, nrow = 2)
ggsave("arrangedplot.png", arrange)

```

```

## Saving 6.5 x 4.5 in image

```

```

ctrl1 = trainControl(method = "cv")

```

Linear regression

```
set.seed(1)
lm.fit <- train(x.train, y.train,
                method = "lm",
                trControl = ctrl1)
summary(lm.fit)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -91.266 -14.386  -1.309   10.491  249.295
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.765e+03  1.328e+02 -20.816  < 2e-16 ***
## age          5.418e-02  1.163e-01   0.466   0.6413
## gender1     -4.167e+00  9.375e-01  -4.445  9.13e-06 ***
## race2        8.711e-01  2.085e+00   0.418   0.6761
## race3     -2.596e+00  1.195e+00  -2.173   0.0299 *
## race4     -2.298e+00  1.582e+00  -1.453   0.1464
## smoking1     5.844e+00  1.056e+00   5.536  3.38e-08 ***
## smoking2     7.779e+00  1.579e+00   4.926  8.87e-07 ***
## height       1.613e+01  7.826e-01  20.605  < 2e-16 ***
## weight     -1.751e+01  8.287e-01 -21.125  < 2e-16 ***
## bmi          5.288e+01  2.364e+00  22.366  < 2e-16 ***
## hypertension1 3.605e+00  1.589e+00   2.268   0.0234 *
## diabetes1    -8.238e-01  1.292e+00  -0.638   0.5238
## SBP          1.935e-02  1.046e-01   0.185   0.8533
## LDL         -5.858e-02  2.480e-02  -2.362   0.0183 *
## vaccine1     -7.888e+00  9.610e-01  -8.208  3.37e-16 ***
## severity1     8.905e+00  1.522e+00   5.852  5.40e-09 ***
## studyB        5.322e+00  1.207e+00   4.411  1.07e-05 ***
## studyC       -1.153e+00  1.453e+00  -0.794   0.4274
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.08 on 2863 degrees of freedom
## Multiple R-squared:  0.2776, Adjusted R-squared:  0.2731
## F-statistic: 61.12 on 18 and 2863 DF, p-value: < 2.2e-16
```

Ridge

```
set.seed(1)
ridge.fit <- train(x.train, y.train,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = 0,
```

```

                                lambda = exp(seq(10, -5, length=200))),
preProc = c("center", "scale"),
trControl = ctrl1)

```

```

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.

```

```

png(file="ridge.png", height=1800, width=1800, units = "px", res=200)
plot(ridge.fit, xTrans = log)
dev.off()

```

```

## pdf
## 2

```

```

ridge.fit$bestTune

```

```

##      alpha      lambda
## 65      0 0.8387191

```

```

# coefficients in the final model
coef(ridge.fit$finalModel, s = ridge.fit$bestTune$lambda)

```

```

## 19 x 1 sparse Matrix of class "dgCMatrix"
##              s1
## (Intercept) 43.1384455
## age         0.2316867
## gender1     -2.1774835
## race2       0.4152253
## race3      -1.0194303
## race4      -0.9515189
## smoking1    2.5563613
## smoking2    2.2580929
## height     2.3521736
## weight     -5.6570081
## bmi        14.1549514
## hypertension1 1.3789345
## diabetes1   -0.1003691
## SBP         0.5960578
## LDL        -0.9994673
## vaccine1    -3.8355786
## severity1   2.7137367
## studyB      2.8016358
## studyC     -0.2768422

```

Lasso

```

set.seed(1)
lasso.fit <- train(x.train, y.train,
                    method = "glmnet",

```

```
tuneGrid = expand.grid(alpha = 1,
                        lambda = exp(seq(10, -5, length = 200))),
trControl = ctrl1)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
png(file="lasso.png", height=1800, width=1800, units = "px", res=200)
plot(lasso.fit, xTrans = log)
dev.off()
```

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

```
lasso.fit$bestTune
```

```
## alpha      lambda
## 1         1 0.006737947
```

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

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##              s1
## (Intercept) -2.659526e+03
## age          5.288381e-02
## gender1      -4.166055e+00
## race2         8.874830e-01
## race3        -2.577262e+00
## race4        -2.309793e+00
## smoking1      5.823620e+00
## smoking2      7.746311e+00
## height        1.550239e+01
## weight       -1.684532e+01
## bmi           5.100002e+01
## hypertension1 3.568911e+00
## diabetes1     -7.834496e-01
## SBP           2.097807e-02
## LDL          -5.789145e-02
## vaccine1      -7.882531e+00
## severity1     8.888889e+00
## studyB        5.339616e+00
## studyC       -1.117748e+00
```

Elastic net

```
set.seed(1)
enet.fit <- train(x.train, y.train,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
```

```

                                lambda = exp(seq(2, -2, length = 50))),
                                trControl = ctrl1)
enet.fit$bestTune

##      alpha      lambda
## 1001      1 0.1353353

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

png(file="enet.png", height=1800, width=1800, units = "px", res=200)
plot(enet.fit, par.settings = myPar)
dev.off()

## pdf
## 2

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

```

```

## 19 x 1 sparse Matrix of class "dgCMatrix"
##              s1
## (Intercept) -1.350889e+03
## age         2.809390e-02
## gender1     -4.052707e+00
## race2       9.488674e-01
## race3      -2.193139e+00
## race4      -2.278903e+00
## smoking1    5.441868e+00
## smoking2    7.128083e+00
## height     7.753069e+00
## weight     -8.628636e+00
## bmi        2.754474e+01
## hypertension1 3.074113e+00
## diabetes1   -1.468496e-01
## SBP        3.786311e-02
## LDL       -4.603493e-02
## vaccine1   -7.719121e+00
## severity1   8.543192e+00
## studyB     5.482654e+00
## studyC    -6.157684e-01

```

PCR

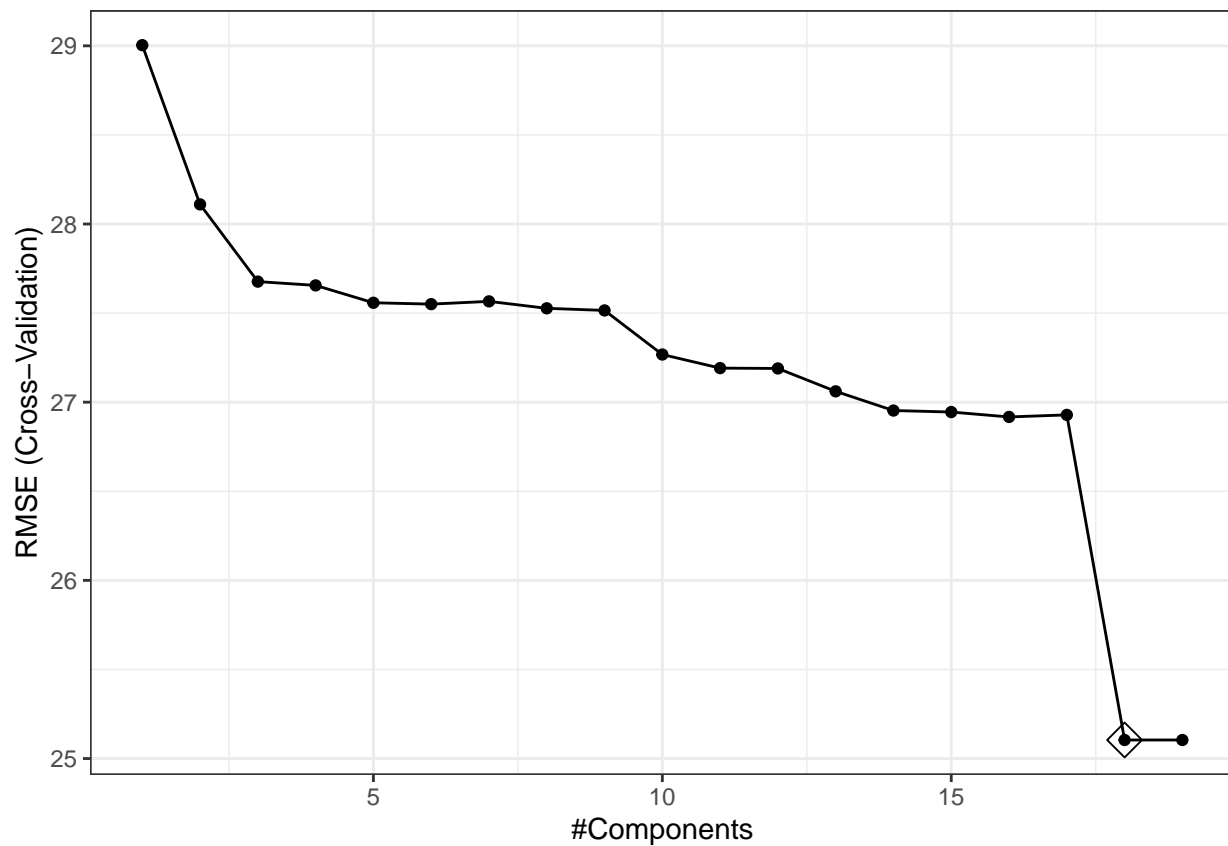
```

set.seed(1)
pcr.fit <- train(x.train, y.train,
                 method = "pcr",
                 tuneGrid = data.frame(ncomp = 1:19),
                 trControl = ctrl1,
                 preProcess = c("center", "scale"))
summary(pcr.fit)

```

```
## Data:      X dimension: 2882 18
## Y dimension: 2882 1
## Fit method: svdpc
## Number of components considered: 18
## TRAINING: % variance explained
##           1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps
## X          12.2664 22.254  31.12  38.15  44.79  51.42  57.61
## .outcome    0.5074  7.019  10.27  10.47  10.53  11.47  11.53
##           8 comps 9 comps 10 comps 11 comps 12 comps 13 comps 14 comps
## X           63.38  68.91  74.34  79.65  84.44  88.73  92.86
## .outcome    11.82  11.83  13.69  13.91  13.92  15.01  15.73
##          15 comps 16 comps 17 comps 18 comps
## X           96.84  98.95  99.99 100.00
## .outcome    15.82  16.01  16.01  27.76
```

```
ggplot(pcr.fit, highlight = TRUE) + theme_bw()
```



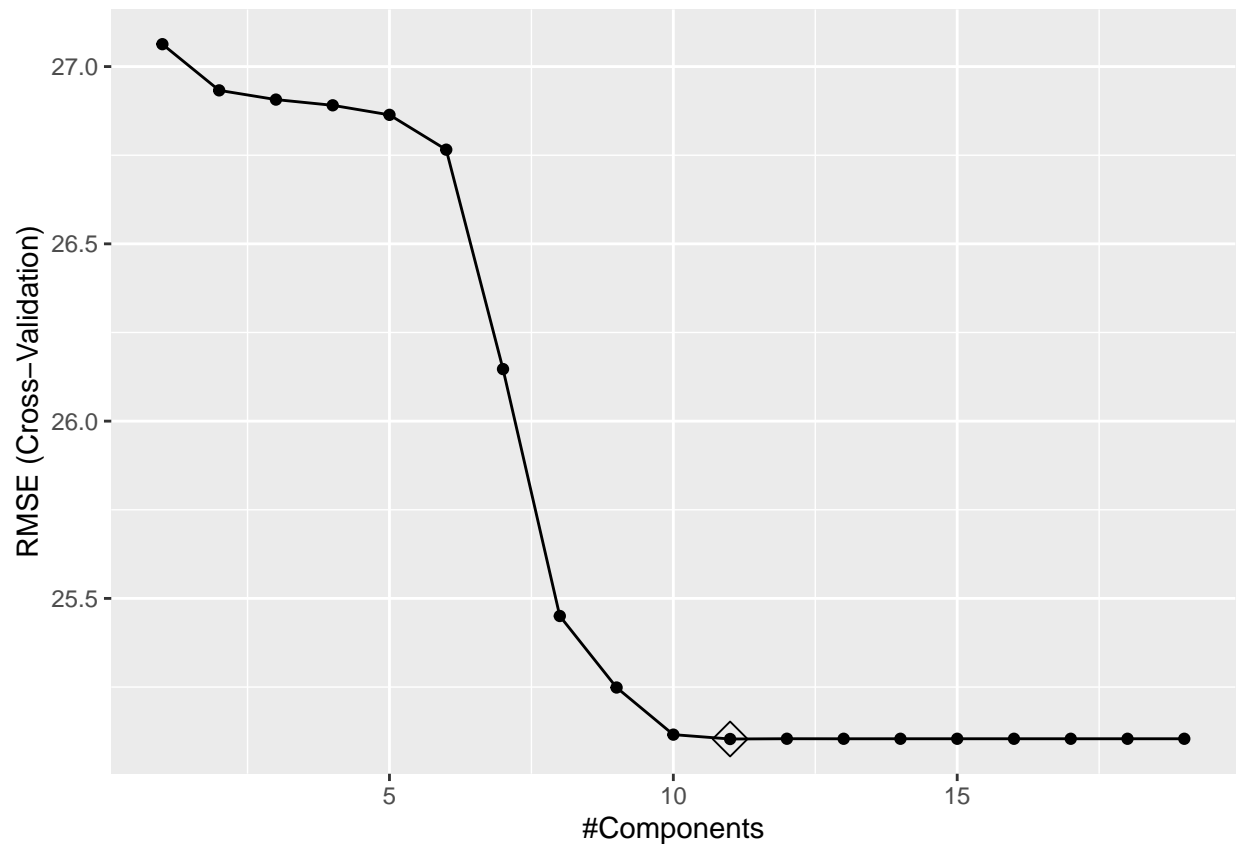
```
ggsave("pcr.tiff", dpi="print")
```

```
## Saving 6.5 x 4.5 in image
```

PLS

```
set.seed(1)
pls.fit <- train(x.train, y.train,
  method = "pls",
  tuneGrid = data.frame(ncomp = 1:19),
  trControl = ctrl1,
  preProcess = c("center", "scale"))

ggplot(pls.fit, highlight = TRUE)
```



```
ggsave("pls.png", dpi="print")
```

```
## Saving 6.5 x 4.5 in image
```

GAM model

```
set.seed(1)
gam.fit <- train(x.train, y.train,
  method = "gam",
  tuneGrid = data.frame(method = "GCV.Cp", select = c(TRUE,FALSE)),
  trControl = ctrl1)
```

```
## 载入需要的程辑包：mgcv
```



```

## 载入需要的程辑包：nlme

##
## 载入程辑包：'nlme'

## The following object is masked from 'package:dplyr':
##
## collapse

## This is mgcv 1.8-42. For overview type 'help("mgcv-package")'.

summary(gam.fit)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender1 + race2 + race3 + race4 + smoking1 + smoking2 +
## hypertension1 + diabetes1 + vaccine1 + severity1 + studyB +
## studyC + s(age) + s(SBP) + s(LDL) + s(bmi) + s(height) +
## s(weight)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  43.2570    1.2848   33.668 < 2e-16 ***
## gender1      -4.7965    0.8214   -5.839 5.84e-09 ***
## race2         0.6993    1.8250    0.383  0.702
## race3        -1.0536    1.0464   -1.007  0.314
## race4        -2.0949    1.3842   -1.513  0.130
## smoking1      4.9659    0.9242    5.373 8.37e-08 ***
## smoking2      7.8849    1.3829    5.702 1.31e-08 ***
## hypertension1 3.4663    0.8248    4.203 2.72e-05 ***
## diabetes1     -0.9107    1.1309   -0.805  0.421
## vaccine1     -8.0399    0.8415   -9.555 < 2e-16 ***
## severity1      9.4416    1.3296    7.101 1.56e-12 ***
## studyB         4.8228    1.0562    4.566 5.17e-06 ***
## studyC        -0.9022    1.2719   -0.709  0.478
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(age)        5.732e-09     9  0.000  0.960
## s(SBP)        7.785e-09     9  0.000  0.757
## s(LDL)        2.455e-01     9  0.036  0.253
## s(bmi)        8.737e+00     9 222.822 <2e-16 ***
## s(height)    2.090e-08     9  0.000  0.408
## s(weight)    1.895e-08     9  0.000  0.446
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.444   Deviance explained = 44.8%
## GCV = 484.97   Scale est. = 481.27      n = 2882

```

```
gam.fit$bestTune
```

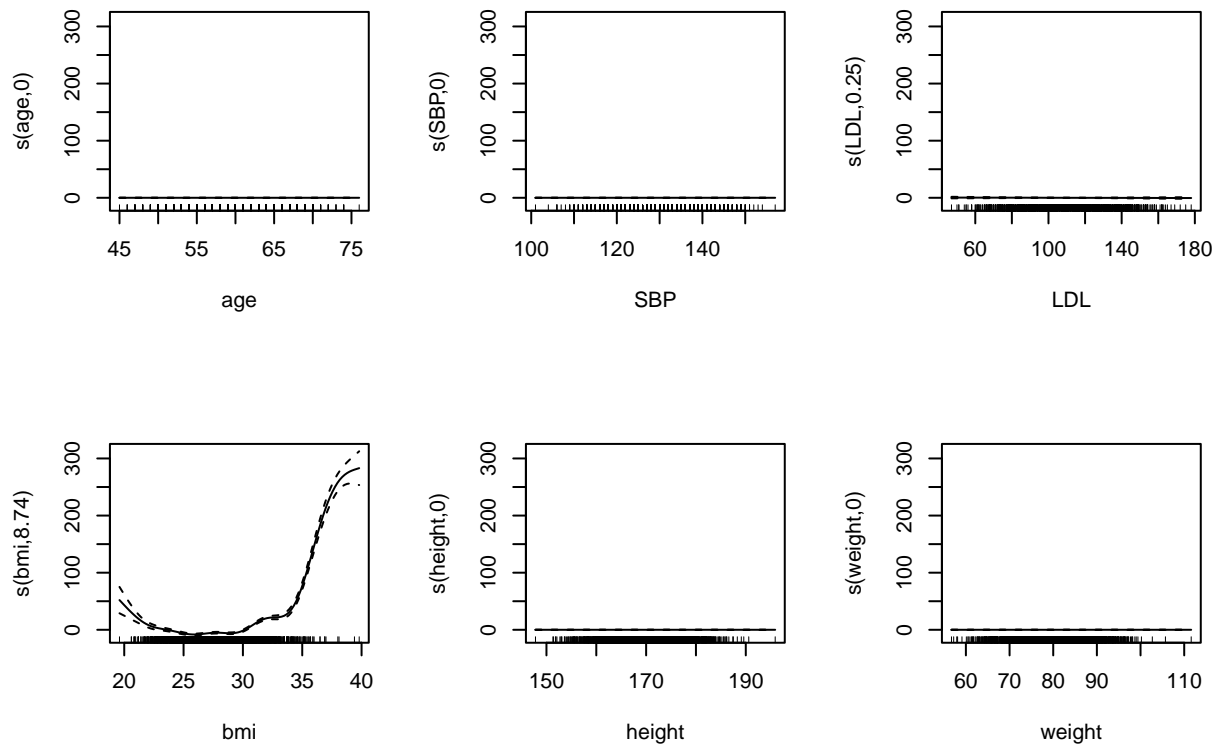
```
## select method  
## 2 TRUE GCV.Cp
```

```
gam.fit$finalModel
```

```
##  
## Family: gaussian  
## Link function: identity  
##  
## Formula:  
## .outcome ~ gender1 + race2 + 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 0.000 0.245 8.737 0.000 0.000 total = 21.98  
##  
## GCV score: 484.9695
```

```
par(mfrow = c(2, 3))
```

```
plot(gam.fit$finalModel)
```

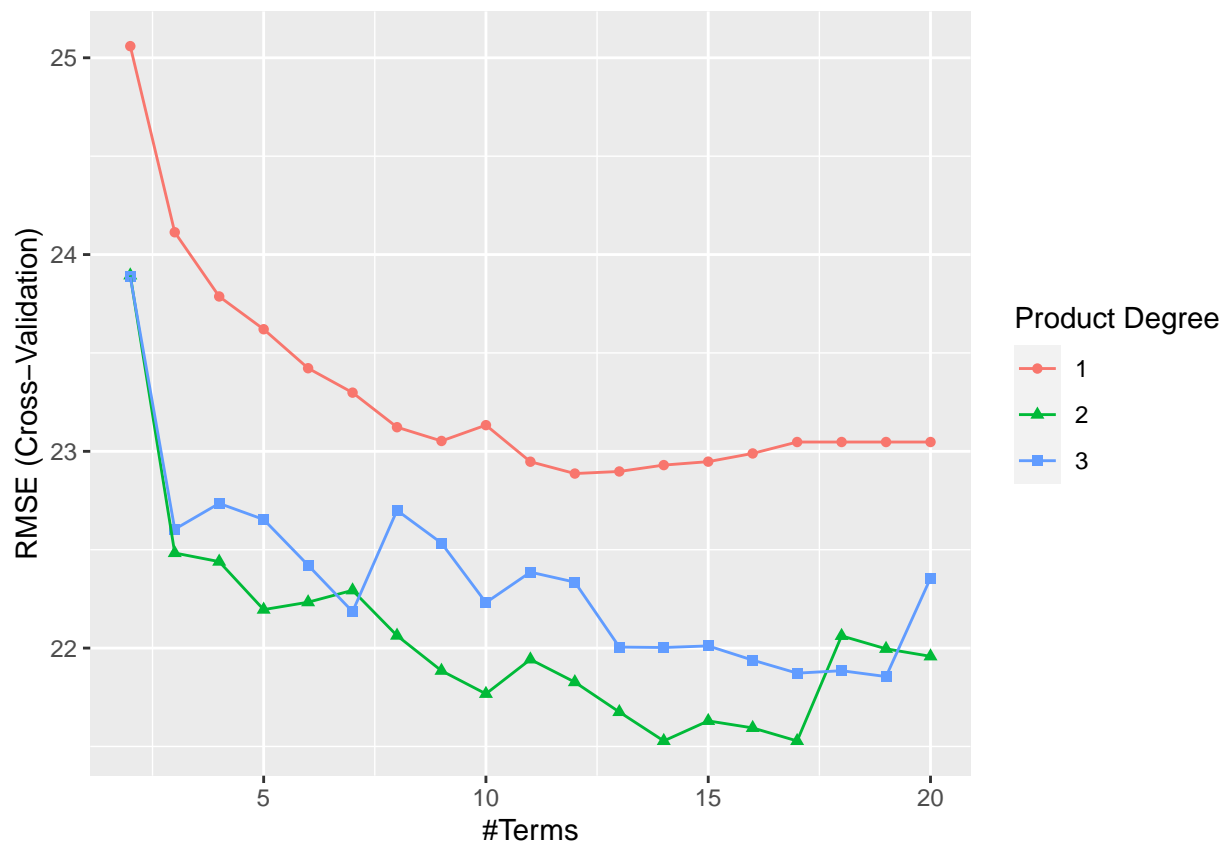


MARS model

```
mars_grid <- expand.grid(degree = 1:3,  
                        nprune = 2:20)  
  
set.seed(1)  
mars.fit <- train(x.train, y.train,  
                 method = "earth",  
                 tuneGrid = mars_grid,  
                 trControl = ctrl1)
```

```
## 载入需要的程辑包：earth
```

```
ggplot(mars.fit)
```



```
mars.fit$bestTune
```

```
##      nprune degree  
## 35      17      2
```

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

```
##           (Intercept)                h(33.3-bmi)      h(bmi-33.3) * studyB
##          -26.8275772                7.5229372          35.5992024
##           vaccine1      h(bmi-28.5) * studyB      race4 * h(bmi-33.3)
##          -7.5760327                6.7123484          -60.2055112
## h(bmi-28.5) * severity1      smoking1 * h(bmi-33.3)      gender1
##           6.4214446                5.5093431          -4.9104162
##           h(bmi-23.9) h(bmi-28.5) * h(SBP-138) h(bmi-28.5) * h(SBP-128)
##           6.9086839                -1.3269459          0.5530760
##           smoking1      smoking2 * h(bmi-33.3) h(bmi-33.3) * h(LDL-115)
##           4.7734050                -54.9024634          0.8394659
## h(bmi-33.3) * h(115-LDL)      smoking2 * h(bmi-28.5)
##           0.5197540                8.2746042
```

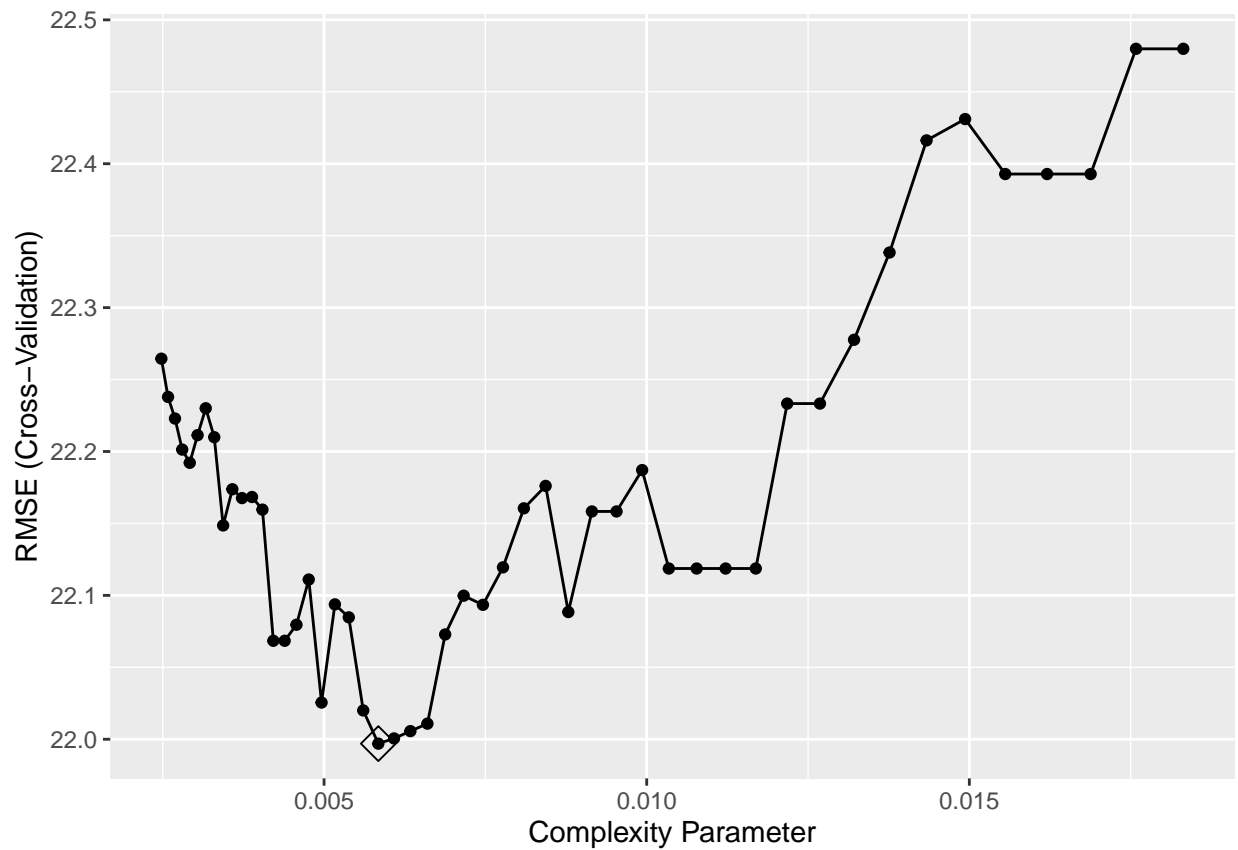
```
summary(mars.fit)
```

```
## Call: earth(x=matrix[2882,18], y=c(56,44,53,51,3...), keepxy=TRUE, degree=2,
##           nprune=17)
##
##               coefficients
## (Intercept)      -26.827577
## gender1          -4.910416
## smoking1         4.773405
## vaccine1        -7.576033
## h(bmi-23.9)       6.908684
## h(33.3-bmi)       7.522937
## race4 * h(bmi-33.3) -60.205511
## smoking1 * h(bmi-33.3) 5.509343
## smoking2 * h(bmi-33.3) -54.902463
## smoking2 * h(bmi-28.5) 8.274604
## h(bmi-28.5) * severity1 6.421445
## h(bmi-28.5) * studyB    6.712348
## h(bmi-33.3) * studyB   35.599202
## h(bmi-28.5) * h(SBP-138) -1.326946
## h(bmi-28.5) * h(SBP-128) 0.553076
## h(bmi-33.3) * h(LDL-115) 0.839466
## h(bmi-33.3) * h(115-LDL) 0.519754
##
## Selected 17 of 22 terms, and 10 of 18 predictors (nprune=17)
## Termination condition: Reached nk 37
## Importance: bmi, studyB, vaccine1, severity1, race4, SBP, gender1, ...
## Number of terms at each degree of interaction: 1 5 11
## GCV 397.1968   RSS 1112383   GRSq 0.5411432   RSq 0.5537964
```

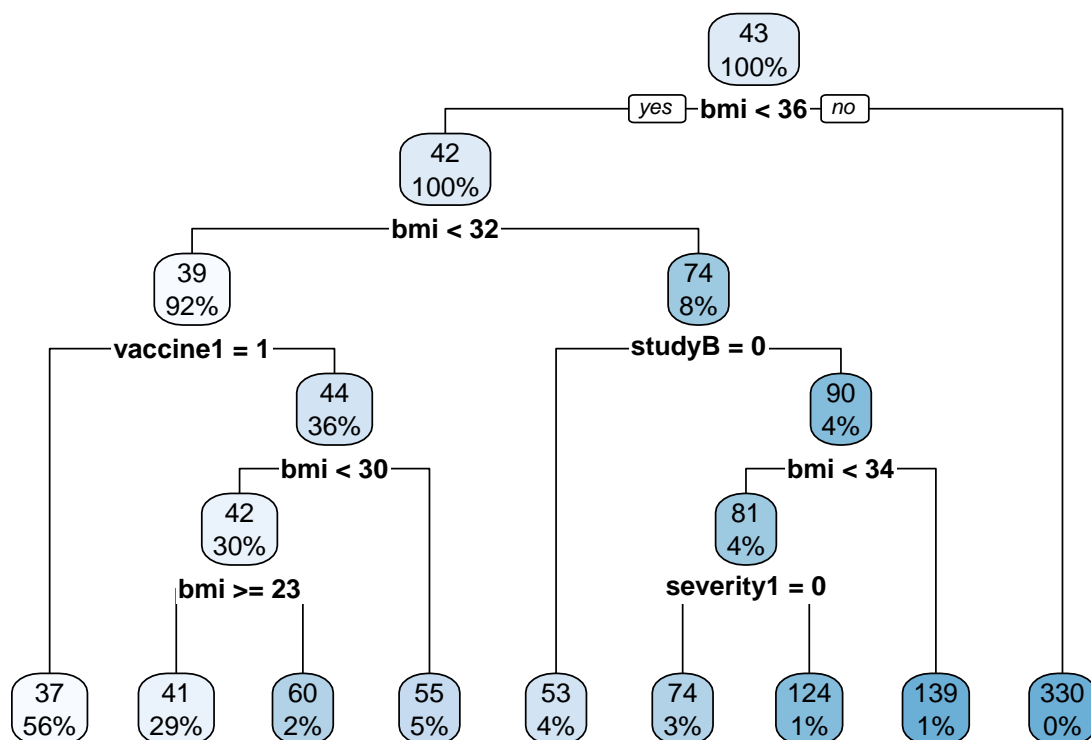
```
png(file="marspdp.png", height=1800, width=1800, units = "px", res=200)
p1 <- pdp::partial(mars.fit, pred.var = c("bmi"), grid.resolution = 10) %>% autoplot()
p2 <- pdp::partial(mars.fit, pred.var = c("SBP"), grid.resolution = 10) %>% autoplot()
p4 <- pdp::partial(mars.fit, pred.var = c("bmi", "SBP"),
  grid.resolution = 10) %>%
  pdp::plotPartial(levelplot = FALSE, zlab = "yhat", drape = TRUE,
    screen = list(z = 20, x = -60))
p1
p2
p4
```

Regression Tree

```
set.seed(1)
rpart.fit <- train(x=x.train,
                  y = y.train,
                  method = "rpart",
                  tuneGrid = data.frame(cp = exp(seq(-6,-4, length = 50))),
                  trControl = ctrl1)
ggplot(rpart.fit, highlight = TRUE)
```



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



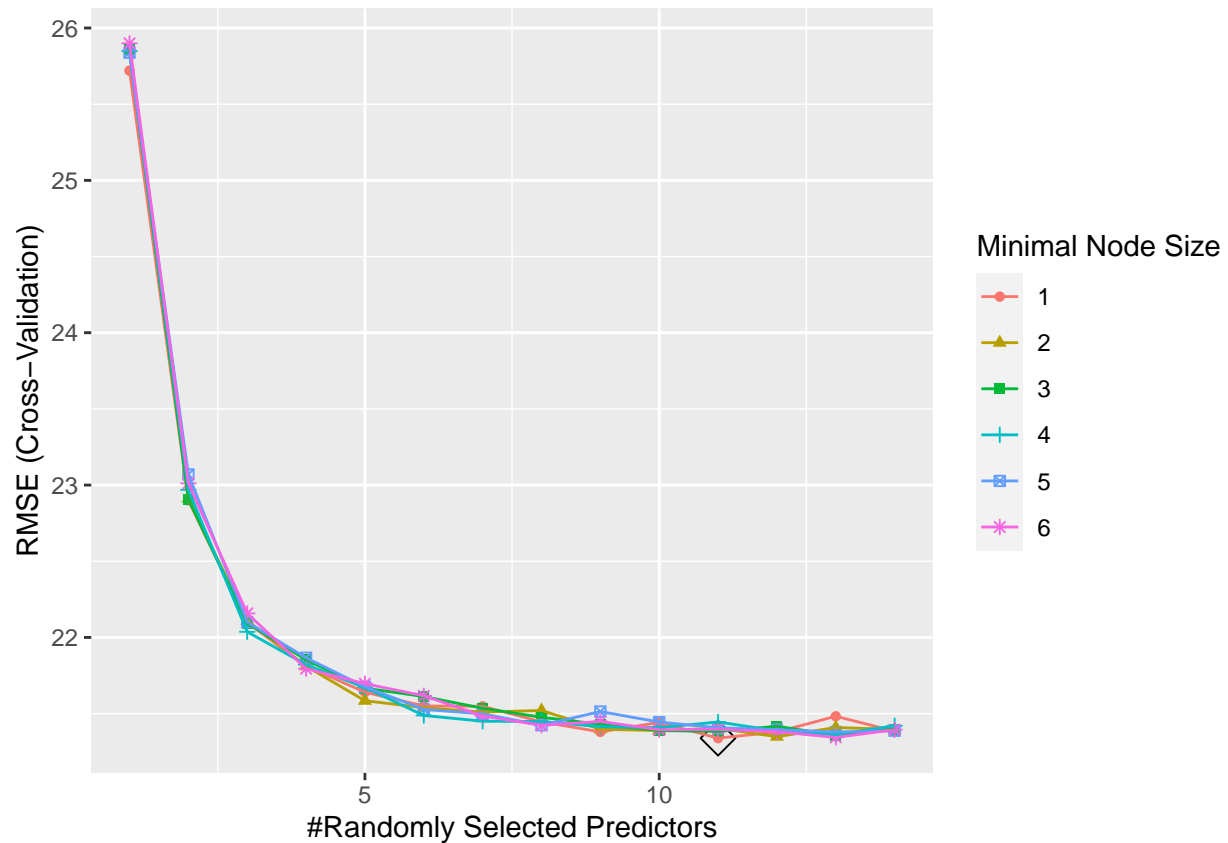
random forest

```
# Try more if possible
rf.grid <- expand.grid(mtry = 1:14,
  splitrule = "variance",
  min.node.size = 1:6)
set.seed(1)
rf.fit <- train(x.train,
  y.train,
  method = "ranger",
  tuneGrid = rf.grid,
  trControl = ctrl1)
```

```
rf.fit$bestTune
```

```
##      mtry splitrule min.node.size
## 61    11  variance                1
```

```
ggplot(rf.fit, highlight = TRUE)
```

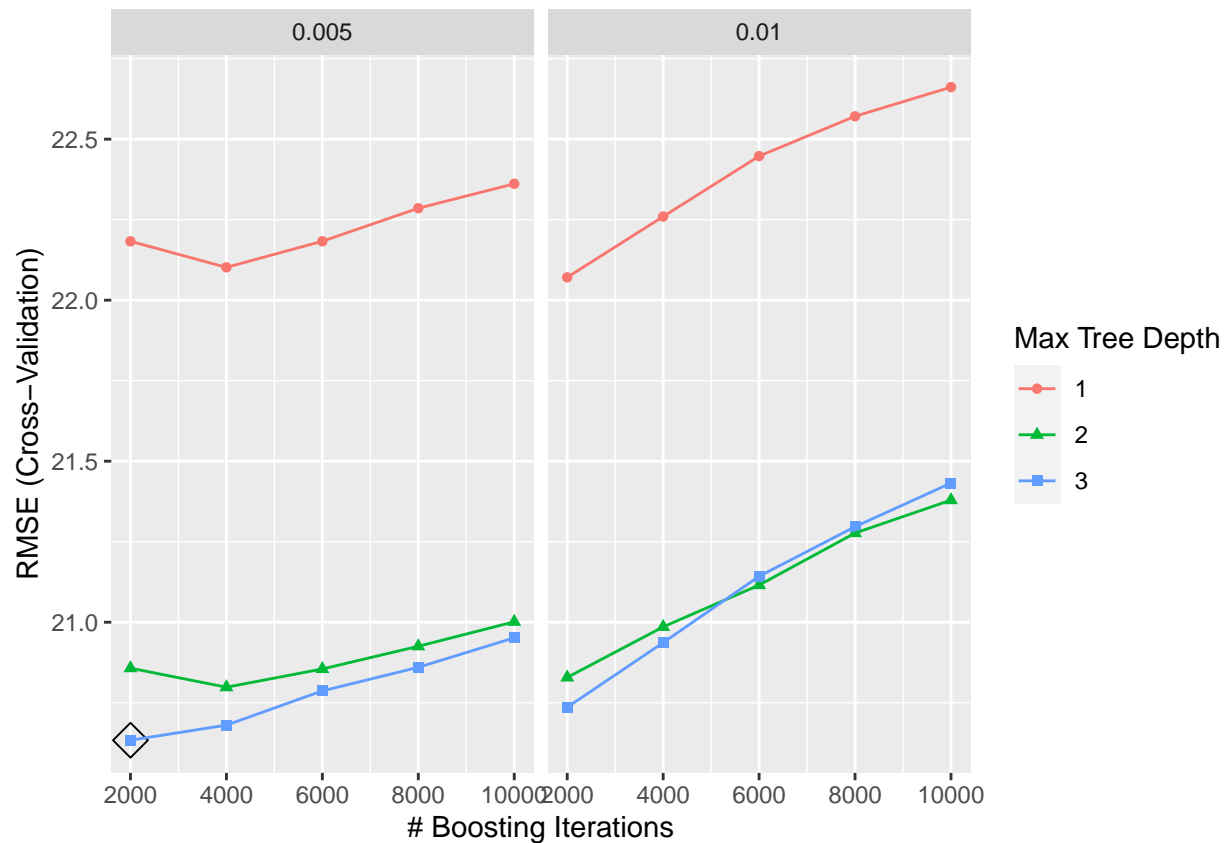


```
rf.pred <- predict(rf.fit, newdata = x.test)
mean((y.test-rf.pred)^2)
```

```
## [1] 512.5156
```

```
gbm.grid <- expand.grid(n.trees = c(2000,4000,6000,8000,10000),
  interaction.depth = 1:3,
  shrinkage = c(0.005,0.01),
  n.minobsinnode = c(1))
set.seed(1)
gbm.fit <- train(x.train,y.train,
  method = "gbm",
  tuneGrid = gbm.grid,
  trControl = ctrl1,
  verbose = FALSE)
```

```
ggplot(gbm.fit, highlight = TRUE)
```



```
gbm.fit$bestTune
```

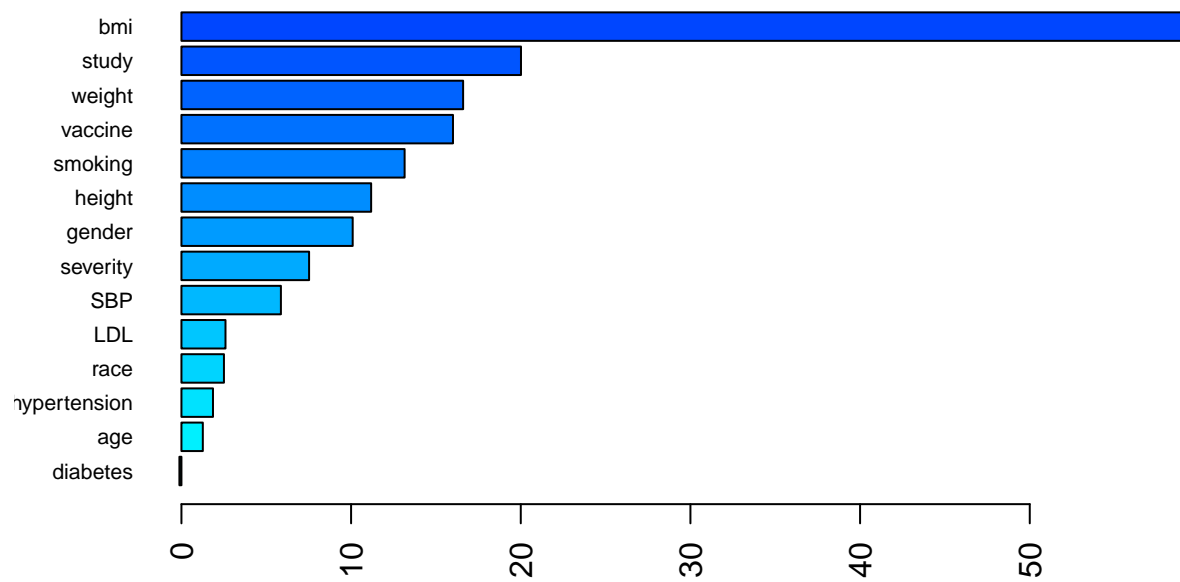
```
##      n.trees interaction.depth shrinkage n.minobsinnode
## 11      2000                3      0.005              1
```

```
gbm.pred <- predict(gbm.fit, newdata = x.test)
mean((y.test-gbm.pred)^2)
```

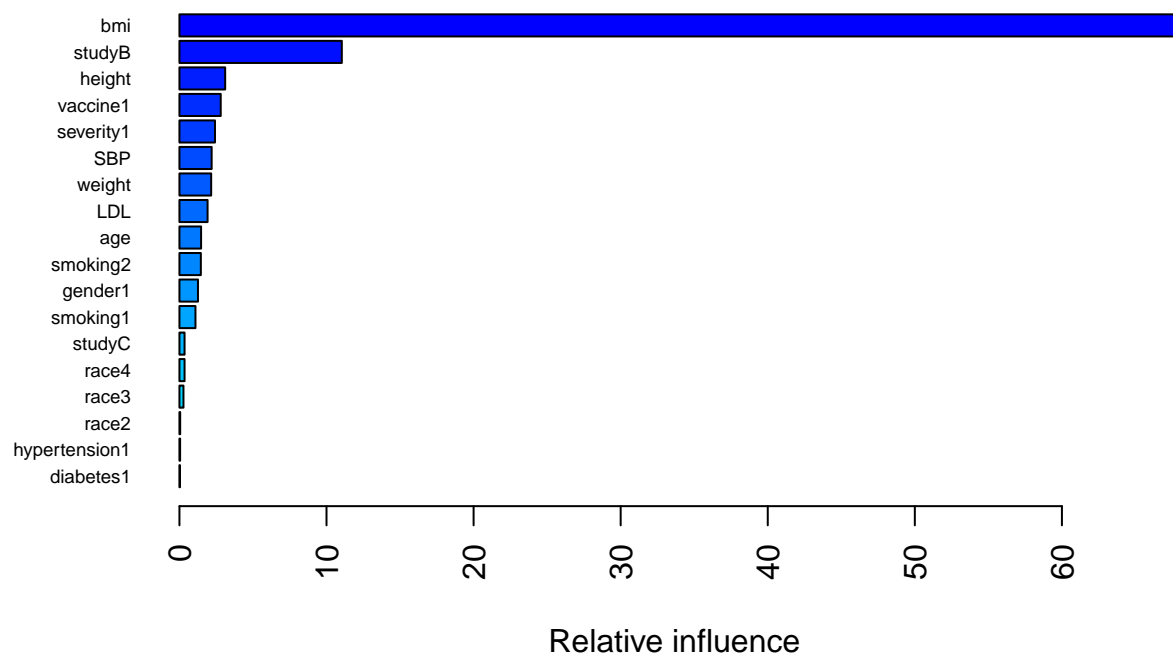
```
## [1] 487.4652
```

```
set.seed(1)
rf2.final.per <- ranger(recovery_time~., dat1[trainRows,],
  mtry = rf.fit$bestTune[[1]],
  splitrule = "variance",
  min.node.size = rf.fit$bestTune[[3]],
  importance = "permutation",
  scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(rf2.final.per), decreasing = FALSE),
  las = 2, horiz = TRUE, cex.names = 0.7,
  col = colorRampPalette(colors = c("cyan", "blue"))(19))
```

```
summary(gbm.fit$finalModel, las = 2, cBars = 19, cex.names = 0.6)
```



```
##           var      rel.inf
## bmi          bmi 67.98507215
## studyB       studyB 11.04019125
## height      height  3.11027713
## vaccine1    vaccine1  2.80598010
## severity1   severity1  2.41754331
## SBP         SBP    2.19109833
## weight      weight  2.15794966
## LDL         LDL    1.91344748
## age         age    1.47161785
## smoking2    smoking2  1.45559538
## gender1     gender1  1.25955715
## smoking1    smoking1  1.09018405
## studyC      studyC  0.34773985
## race4       race4   0.34765420
## race3       race3   0.27311430
## race2       race2   0.05118855
## hypertension1 hypertension1 0.04612109
## diabetes1   diabetes1 0.03566815
```

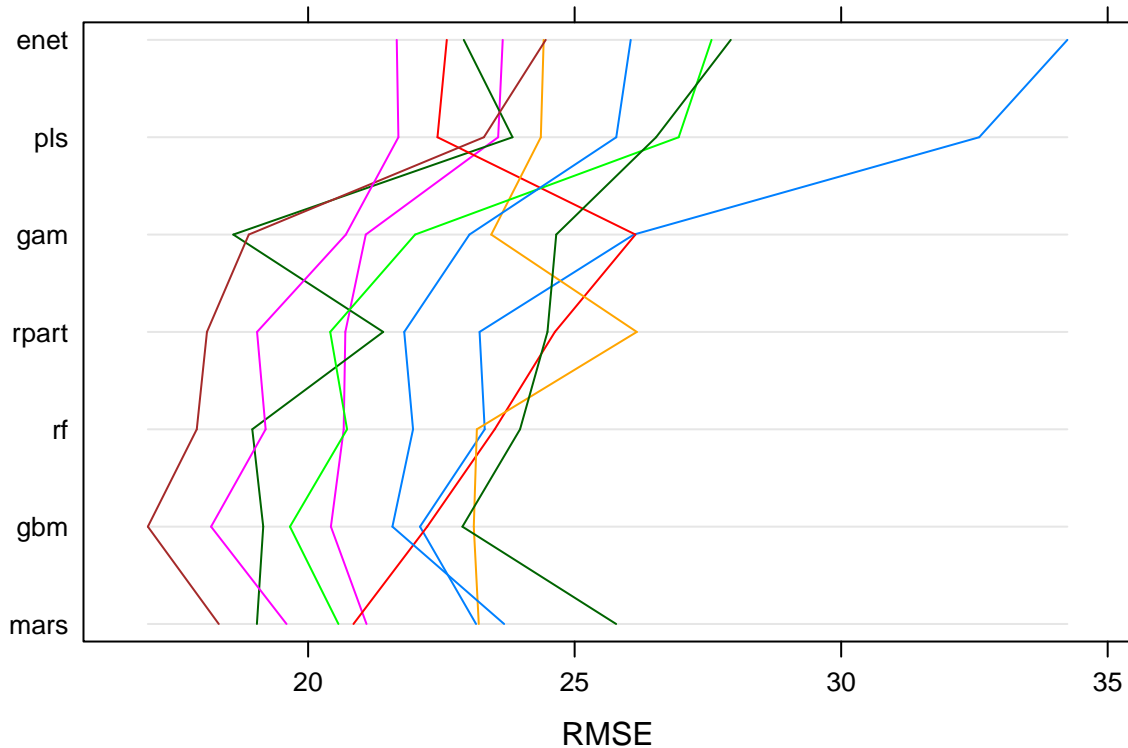
Comparing different models

```
set.seed(1)
resamp = resamples(list(enet =enet.fit, pls = pls.fit, gam = gam.fit, mars = mars.fit, rf = rf.fit, gbm = gbm.fit))
```

```
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: enet, pls, gam, mars, rf, gbm, rpart
## Number of resamples: 10
##
## MAE
##           Min. 1st Qu.  Median    Mean 3rd Qu.    Max. NA's
## enet  14.89563 15.41241 16.31429 16.48704 17.18978 18.68102    0
## pls   14.88596 15.73014 16.78619 16.77285 17.65509 18.70689    0
## gam   13.63349 14.66753 15.56539 15.39251 16.01871 16.82288    0
## mars  13.57177 14.17293 14.70892 14.88652 15.85824 16.12610    0
## rf    13.34452 13.71658 14.81148 14.75197 15.67082 16.18638    0
## gbm   12.80459 13.54969 14.46672 14.36228 15.21148 15.78439    0
## rpart 13.57200 14.43384 15.08326 15.06431 15.64389 16.85209    0
##
## RMSE
##           Min. 1st Qu.  Median    Mean 3rd Qu.    Max. NA's
## enet  21.66236 23.10295 24.43842 25.55056 27.18938 34.24423    0
## pls   21.69353 23.36433 24.10026 25.10344 26.34144 32.59019    0
## gam   18.59663 20.80165 22.51682 22.46502 24.35001 26.14106    0
## mars  18.32398 19.83786 20.97298 21.52795 23.18857 25.77849    0
## rf    17.91126 19.56908 21.34942 21.33878 23.27605 23.97571    0
## gbm   16.99334 19.28323 21.00432 20.63349 22.20037 23.10783    0
## rpart 18.10205 20.48515 21.60586 21.99696 24.17166 26.16617    0
##
## Rsquared
##           Min. 1st Qu.  Median    Mean 3rd Qu.    Max. NA's
## enet  0.08983837 0.1977097 0.2506698 0.2393136 0.2882582 0.3466723    0
## pls   0.11783579 0.2047681 0.2671251 0.2632351 0.3365684 0.3870614    0
## gam   0.14925569 0.3053848 0.4458158 0.4177506 0.5275640 0.6443694    0
## mars  0.14582135 0.3633383 0.4571700 0.4526473 0.5790718 0.6588231    0
## rf    0.14785585 0.3680986 0.4628669 0.4531400 0.5706298 0.6700218    0
## gbm   0.14759097 0.4206024 0.4852933 0.4842539 0.6221652 0.7058248    0
## rpart 0.03222650 0.3573939 0.4622695 0.4296027 0.5761503 0.6673153    0
```

```
parallelplot(resamp, metric = "RMSE")
```



```
png(file="comparison.png", height=1800, width=1800, units = "px", res=200)
bwplot(resamp, metric = "RMSE")
```

```
png(file="VIP.png", height=1800, width=1800, units = "px", res=200)
p1 <- vip(mars.fit, num_features = 40, bar = FALSE, value = "gcv") + ggtitle("GCV")
p2 <- vip(mars.fit, num_features = 40, bar = FALSE, value = "rss") + ggtitle("RSS")

gridExtra::grid.arrange(p1, p2, ncol = 2)
dev.off()
```

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

Prediction

```
predy2.pls = predict(lm.fit, newdata = x.test)
mean((y.test-predy2.pls)^2)
```

```
## [1] 652.6144
```

```
predy2.pls = predict(ridge.fit, newdata = x.test)
mean((y.test-predy2.pls)^2)
```

```
## [1] 745.3525
```

```
predy2.pls = predict(enet.fit, newdata = x.test)
mean((y.test-predy2.pls)^2)
```

```
## [1] 677.4994
```

```
predy2.pls = predict(pls.fit, newdata = x.test)
mean((y.test-predy2.pls)^2)
```

```
## [1] 652.476
```

```
predy2.pcr = predict(pcr.fit, newdata = x.test)
mean((y.test-predy2.pcr)^2)
```

```
## [1] 652.6144
```

```
predy2.mars = predict(mars.fit, newdata = x.test)
mean((y.test-predy2.mars)^2)
```

```
## [1] 439.1718
```

```
predy2.lasso = predict(lasso.fit, newdata = x.test)
mean((y.test-predy2.lasso)^2)
```

```
## [1] 652.7542
```

```
predy2.gam = predict(gam.fit, newdata = x.test)
mean((y.test-predy2.gam)^2)
```

```
## [1] 486.8146
```