P8106_midterm

yimin chen yc4195

2023-03-26

```
library(corrplot)
library(tidyverse)
library(caret)
library(mgcv)
library(patchwork)
library(rpart)
library(rpart.plot)
library(party)
library(randomForest)
library(ranger)
library(e1071)
library(pdp)
library(earth)
library(splines)
library(pdp)
library(ggplot2)
library(gridExtra)
library(glmnet)
library(MASS)
library(pROC)
library(vip)
library(AppliedPredictiveModeling)
library(klaR)
```

import and Data

```
set.seed(4195)
load("recovery.Rdata")
dat <- dat[sample(1:10000, 2000),]
dat <- dat[,-1]%>% #ignore the id variable
    mutate(study = case_when( # from character variable to a numeric variable
    study == "A" ~ 1,
    study == "B" ~ 2,
    study == "C" ~ 3)) %>%
    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)
  )
skimr::skim_without_charts(dat)
```

Table 1: Data summary

Name	dat
Number of rows	2000
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: 1009, 1: 991
race	0	1	FALSE	4	1: 1246, 3: 439, 4: 221, 2: 94
smoking	0	1	FALSE	3	0: 1209, 1: 583, 2: 208
hypertension	0	1	FALSE	2	0: 1030, 1: 970
diabetes	0	1	FALSE	2	0: 1722, 1: 278
vaccine	0	1	FALSE	2	1: 1189, 0: 811
severity	0	1	FALSE	2	0: 1804, 1: 196
study	0	1	FALSE	3	2: 1152, 3: 432, 1: 416

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
age	0	1	60.08	4.58	45.0	57.00	60.0	63.0	77.0
height	0	1	170.10	5.89	151.4	166.20	170.3	174.1	189.3
weight	0	1	79.92	6.99	57.5	75.00	79.9	84.7	104.2
bmi	0	1	27.67	2.65	19.7	25.80	27.6	29.4	37.1
SBP	0	1	130.43	8.06	104.0	125.00	130.0	136.0	156.0
LDL	0	1	110.30	20.22	47.0	96.75	110.0	124.0	172.0
recovery_time	0	1	41.83	27.05	2.0	28.00	38.5	49.0	365.0

Create x and y matrixs for modeling

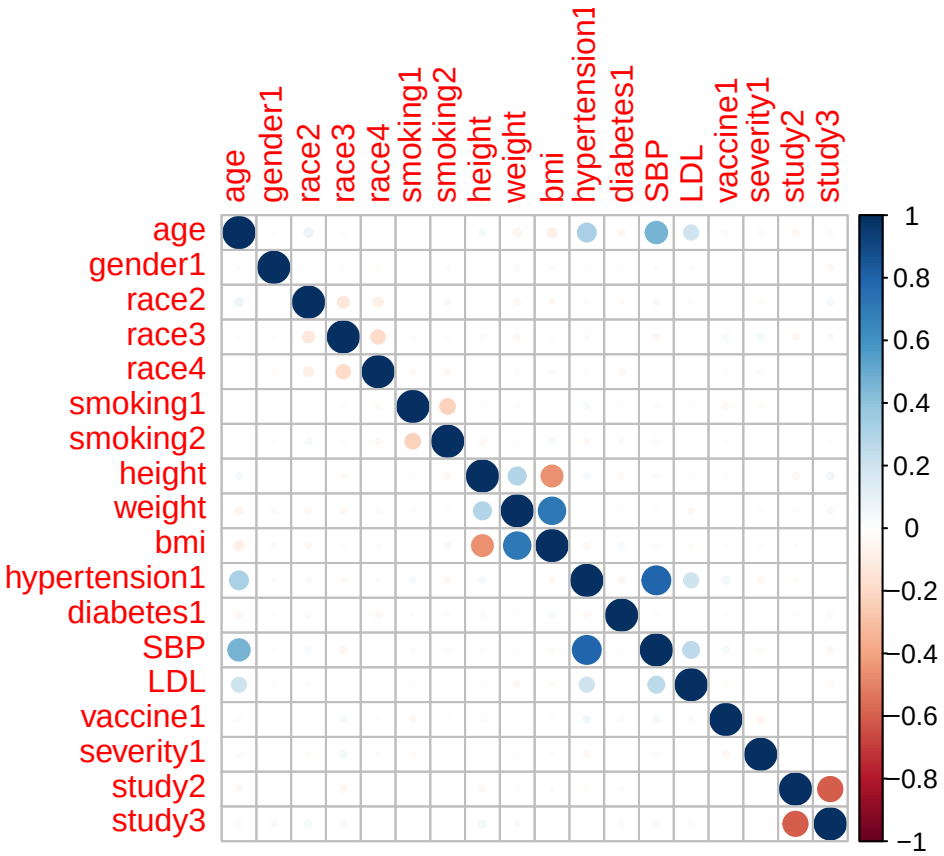
```
ctrl <- trainControl(method = "repeatedcv",
                     repeats = 5,
                     number=10)
set.seed(4195)
```

```
traindata <- createDataPartition(dat$recovery_time, p = 0.7, list = FALSE)
traindataset <- dat[traindata,]
testdataset <- dat[-traindata,]
#train
x = model.matrix(recovery_time ~ ., dat)[traindata,-1]
y = traindataset$recovery_time

#test
x_test = model.matrix(recovery_time ~ ., dat)[-traindata,-1]
y_test = testdataset$recovery_time
```

Exploratory analysis and data visualization:

```
#corr plot
corrplot(cor(x), method = "circle", type = "full")
```

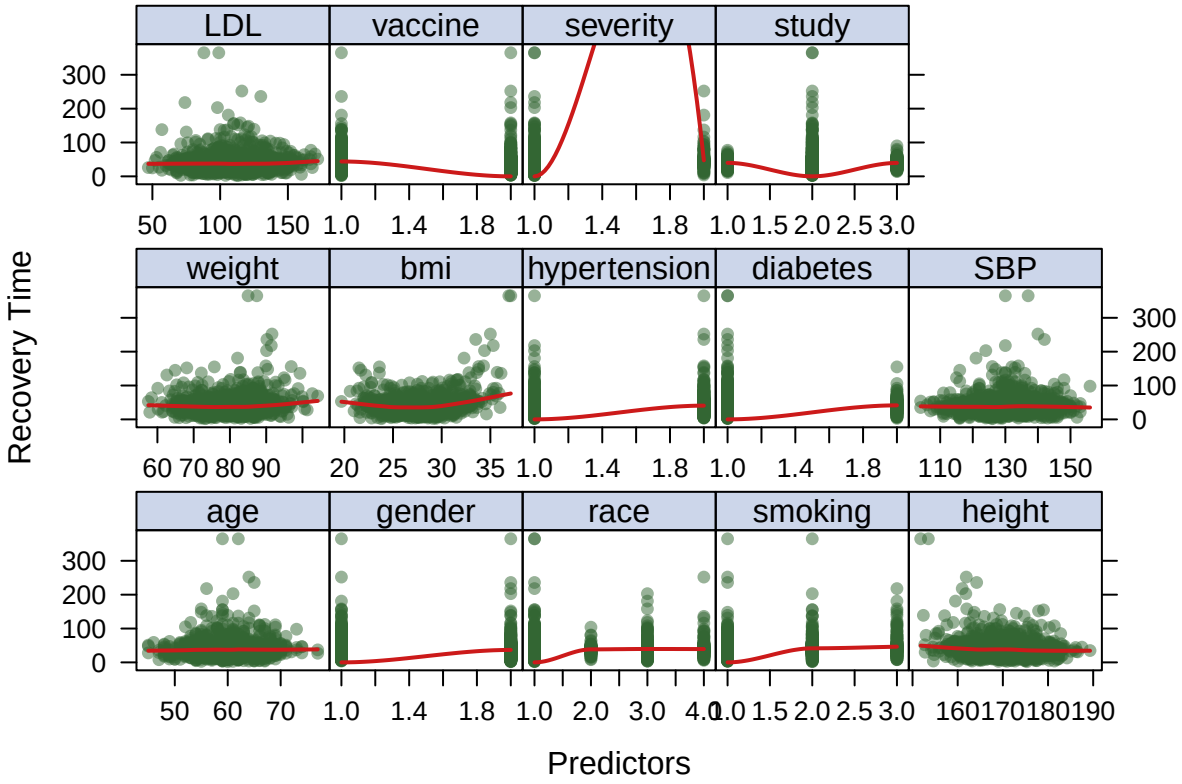



```
# create dataset for exploratory analysis and data visualization
traindataset1 <- traindataset%>%
    mutate(
    gender = as.numeric(gender),
    race = as.numeric(race),
    smoking = as.numeric(smoking),
    hypertension = as.numeric(hypertension),
```

```
    diabetes = as.numeric(diabetes),
    vaccine = as.numeric(vaccine),
    severity = as.numeric(severity),
    study = as.numeric(study))
```

```
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)

## Exploratory analysis and visualization
featurePlot(x = traindataset1[ ,1:14],
            y = traindataset1[ ,15],
            plot = "scatter",
            span = .5,
            #layout=c(4,4),
            labels = c("Predictors", "Recovery Time"),
            type = c("p", "smooth"))
```



Model training:

Fit a linear model:

```
set.seed(4195)
# Fit a linear regression model
lm.fit <- train(recovery_time ~ age + gender + race + smoking + height + weight +
                  bmi + hypertension + diabetes + SBP + LDL + vaccine +
                  severity + study,
                data = traindataset,
                method = "lm",
                trControl = ctrl)
# model summary
summary(lm.fit$finalModel)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -82.006 -12.582  -0.872   9.012 235.079
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.580e+03  1.888e+02 -13.666  < 2e-16 ***
## age            2.584e-01  1.551e-01   1.666  0.09600 .
## gender1       -5.893e+00  1.245e+00  -4.733 2.44e-06 ***
## race2         -4.165e+00  2.869e+00  -1.452  0.14676
## race3          3.060e-01  1.556e+00   0.197  0.84412
## race4         -7.088e-01  2.041e+00  -0.347  0.72838
## smoking1       4.534e+00  1.398e+00   3.242  0.00122 **
## smoking2       1.021e+01  2.083e+00   4.900 1.07e-06 ***
## height         1.509e+01  1.111e+00  13.586  < 2e-16 ***
## weight        -1.639e+01  1.176e+00 -13.934  < 2e-16 ***
## bmi            4.892e+01  3.366e+00  14.534  < 2e-16 ***
## hypertension1  2.698e+00  2.066e+00   1.306  0.19168
## diabetes1     -3.749e-01  1.819e+00  -0.206  0.83674
## SBP            1.498e-02  1.356e-01   0.110  0.91205
## LDL           -4.738e-02  3.187e-02  -1.486  0.13739
## vaccine1      -6.875e+00  1.280e+00  -5.373 9.07e-08 ***
## severity1      8.467e+00  2.130e+00   3.975 7.40e-05 ***
## study2         2.775e+00  1.573e+00   1.764  0.07788 .
## study3        -9.477e-01  1.900e+00  -0.499  0.61805
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.25 on 1383 degrees of freedom
## Multiple R-squared:  0.2404, Adjusted R-squared:  0.2305
## F-statistic: 24.31 on 18 and 1383 DF,  p-value: < 2.2e-16
```

```
# RMSE
test_pred_lm <- predict(lm.fit, newdata = testdataset)
test_rmse_lm <- sqrt(mean((test_pred_lm -y_test)^2))
test_rmse_lm
```

```
## [1] 24.21453
```

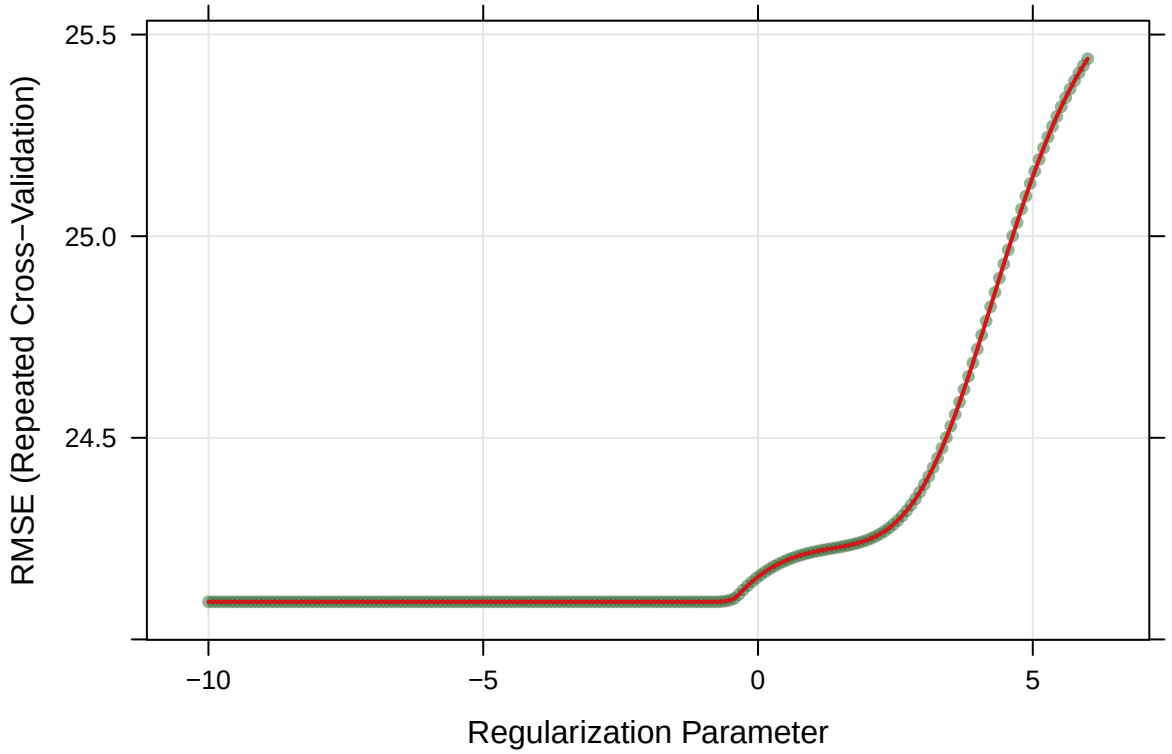
RMSE = 24.214526

Fit Ridge Regression

```
set.seed(4195)
ridge.fit = train(x, y,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = 0,
                                         lambda = exp(seq(-10, 6, length = 200))),
                  trControl = ctrl)
ridge_pred = predict(ridge.fit$finalModel, newx = x_test, s = ridge.fit$bestTune$lambda, type = "respons
# model summary
summary(ridge.fit$finalModel)
```

```
##             Length Class      Mode
## a0           100   -none-     numeric
## beta        1800   dgCMatrix  S4
## df           100   -none-     numeric
## dim            2   -none-     numeric
## lambda       100   -none-     numeric
## dev.ratio    100   -none-     numeric
## nulldev        1   -none-     numeric
## npasses        1   -none-     numeric
## jerr           1   -none-     numeric
## offset         1   -none-     logical
## call           5   -none-     call
## nobs           1   -none-     numeric
## lambdaOpt      1   -none-     numeric
## xNames        18   -none-     character
## problemType    1   -none-     character
## tuneValue      2   data.frame list
## obsLevels      1   -none-     logical
## param          0   -none-     list
```

```
plot(ridge.fit,xTrans =log)
```




```
ridge.fit$bestTune
```

```
##     alpha    lambda
## 116     0 0.4705896
```

```
# RMSE
test_pred_ridge <- predict(ridge.fit, newdata = data.frame(x_test))
test_rmse_ridge <- sqrt(mean((test_pred_ridge -y_test)^2))
test_rmse_ridge
```

```
## [1] 26.43512
```

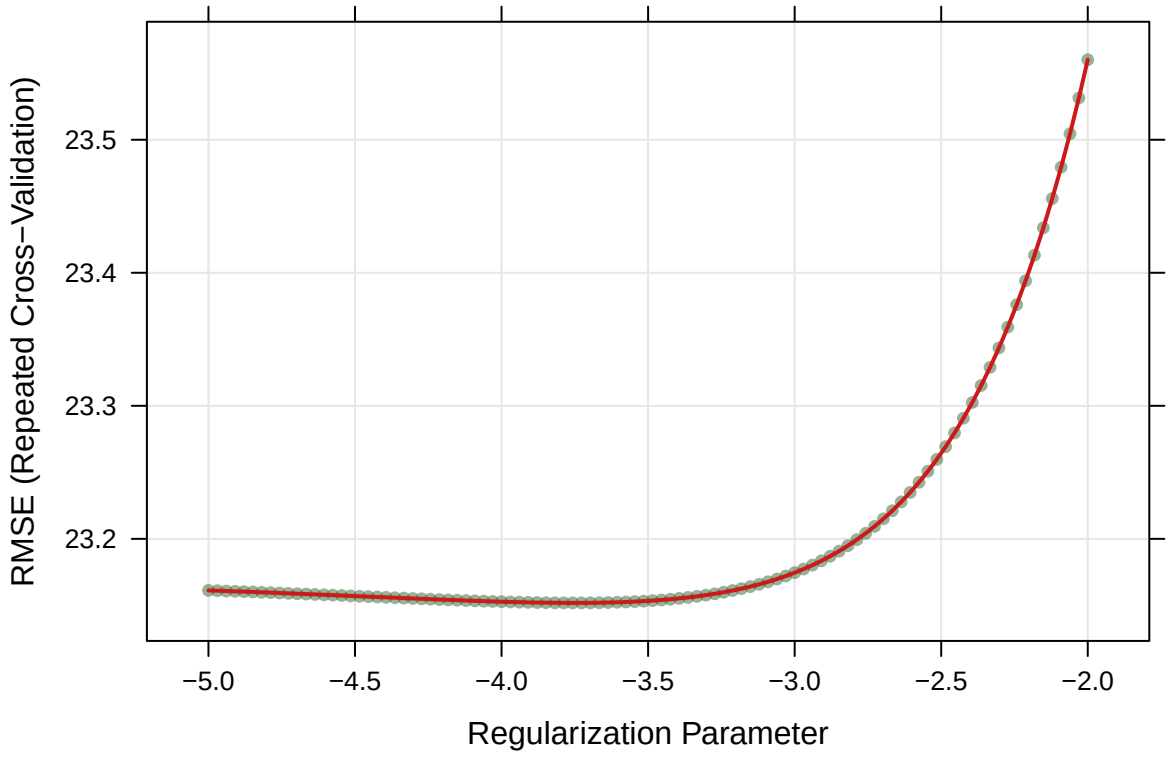
RMSE = 26.4351178

Fit Lasso model:

```
set.seed(4195)
lasso.fit <- train(x, y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(-5, -2, length = 100))),
                   trControl = ctrl)
# view the model summary
summary(lasso.fit$finalModel)
```

```
##             Length Class      Mode
## a0             93   -none-     numeric
## beta         1674   dgCMatrix  S4
## df             93   -none-     numeric
## dim             2   -none-     numeric
## lambda         93   -none-     numeric
## dev.ratio      93   -none-     numeric
## nulldev         1   -none-     numeric
## npasses         1   -none-     numeric
## jerr            1   -none-     numeric
## offset          1   -none-     logical
## call            5   -none-     call
## nobs            1   -none-     numeric
## lambdaOpt       1   -none-     numeric
## xNames         18   -none-     character
## problemType     1   -none-     character
## tuneValue       2   data.frame list
## obsLevels       1   -none-     logical
## param           0   -none-     list
```

```
plot(lasso.fit,xTrans =log)
```




```
lasso.fit$bestTune
```

```
##    alpha     lambda
## 42     1 0.02334025
```

```r
# view performance on the test set (RMSE)
lasso_pred <- predict(lasso.fit, newdata = data.frame(x_test))
test_rmse_lasso<- sqrt(mean((lasso_pred - y_test)^2))
test_rmse_lasso
```

```
## [1] 24.34447
```

RMSE = 24.3444707

Fit Elastic net model:

```r
set.seed(4195)
enet.fit <- train(x, y,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                         lambda = exp(seq(2, -2, length = 50))),
                  trControl = ctrl)
```

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

```r
# view the model summary
summary(enet.fit$finalModel)
```

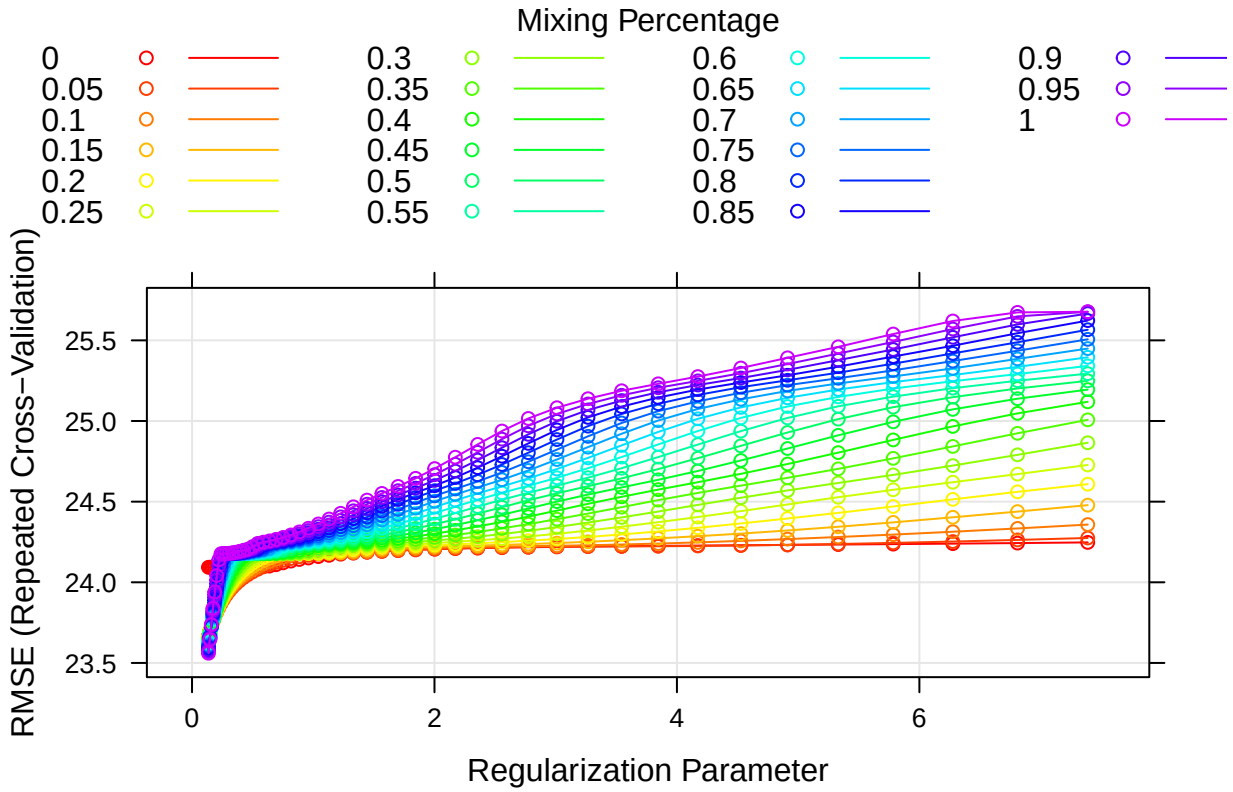
```
##             Length Class      Mode
## a0            93   -none-     numeric
## beta        1674   dgCMatrix  S4
## df            93   -none-     numeric
## dim            2   -none-     numeric
## lambda        93   -none-     numeric
## dev.ratio     93   -none-     numeric
## nulldev        1   -none-     numeric
## npasses        1   -none-     numeric
## jerr           1   -none-     numeric
## offset         1   -none-     logical
## call           5   -none-     call
## nobs           1   -none-     numeric
## lambdaOpt      1   -none-     numeric
## xNames        18   -none-     character
## problemType    1   -none-     character
## tuneValue      2   data.frame list
## obsLevels      1   -none-     logical
## param          0   -none-     list
```

```r
enet.fit$bestTune
```

```
##      alpha    lambda
## 1001     1 0.1353353
```

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

plot(enet.fit, par.settings = myPar)
```




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

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                             s1
## (Intercept)   -971.60882559
## age              0.24586872
## gender1         -5.74078448
## race2           -3.14016554
## race3            .
## race4           -0.25586131
## smoking1         4.04765122
## smoking2        10.25806216
## height           5.59355972
## weight          -6.32138317
## bmi             20.09733437
## hypertension1    2.47119605
## diabetes1       -0.12472863
## SBP              0.01248712
## LDL             -0.03516016
## vaccine1        -6.80532972
## severity1        8.35373727
```

```
## study2           2.81593202
## study3          -0.40909368
```

```
# view performance on the test set (RMSE)
enet_pred <- predict(enet.fit, data.frame(x_test))
test_rmse_enet <- sqrt(mean((enet_pred - y_test)^2))
test_rmse_enet
```

```
## [1] 25.35936
```

```
# calculate RMSE
sqrt(mean((enet_pred - y_test)^2))
```

```
## [1] 25.35936
```

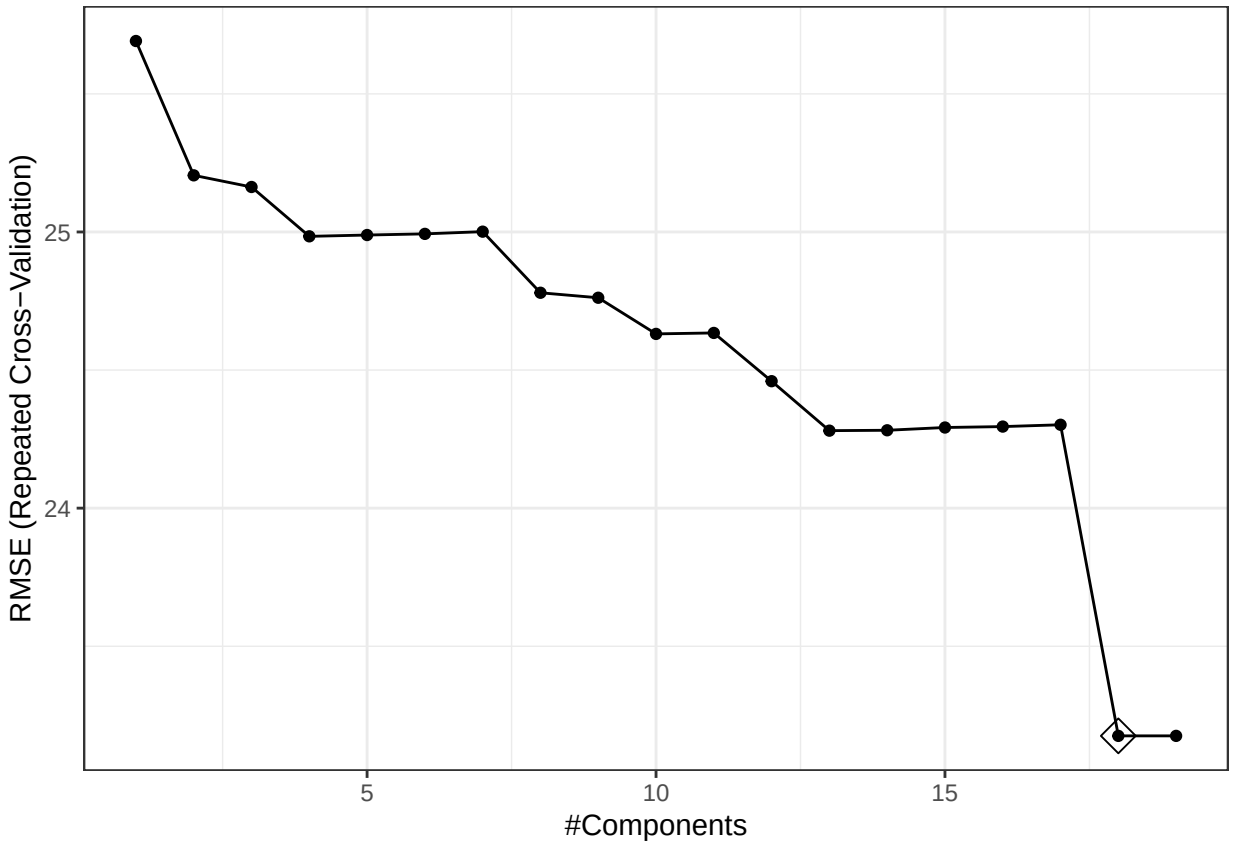
RMSE = 25.3593609

Fit a PCR model

```
set.seed(4195)
pcr.fit <- train(x, y,
                 method = "pcr",
                 tuneGrid  = data.frame(ncomp = 1:19),
                 trControl = ctrl,
                 preProcess = c("center", "scale"))
# model summary
summary(pcr.fit$finalModel)
```

```
## Data:    X dimension: 1402 18
##  Y dimension: 1402 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         12.51500   22.141    31.18   38.300   45.194   51.708   57.806
## .outcome   0.01676    4.682     5.10    6.685    6.797    6.979    7.018
##           8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X          63.619   69.264     74.69     79.98     84.69     88.94     92.93
## .outcome    8.773    8.844     10.23     10.24     11.57     12.90     12.95
##           15 comps  16 comps  17 comps  18 comps
## X            96.76     98.96     99.99    100.00
## .outcome     13.02     13.06     13.08     24.04
```

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




```
# RMSE
test_pred_pcr <- predict(pcr.fit, newdata = data.frame(x_test))
test_rmse_pcr <- sqrt(mean((test_pred_pcr -y_test)^2))
test_rmse_pcr
```

```
## [1] 24.21453
```

RMSE = 24.214526

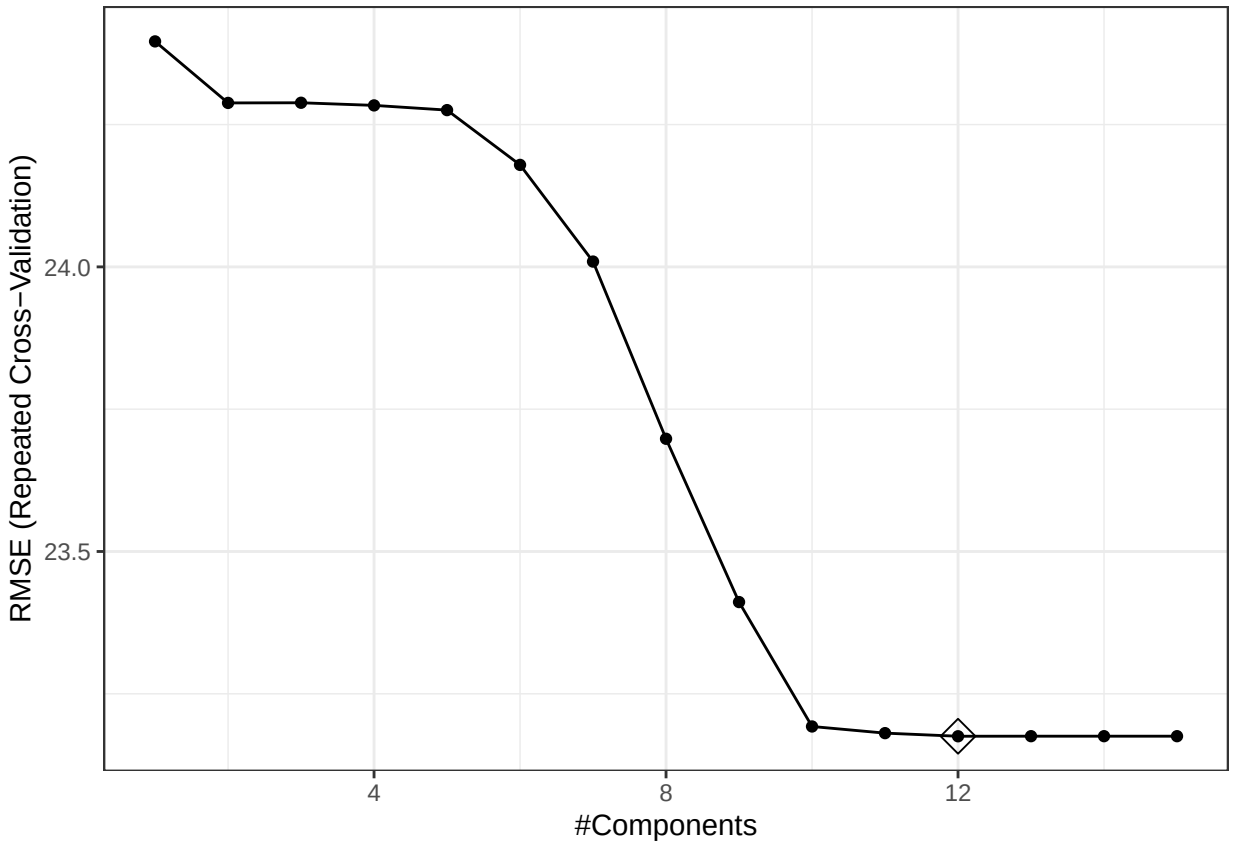
Fit a Partial least squares model(PLS):

```
set.seed(4195)
pls.fit <- train(x, y,
                 method = "pls",
                 tuneGrid = data.frame(ncomp = 1:15),
                 trControl = ctrl,
                 preProcess = c("center", "scale"))
# view the model summary
summary(pls.fit$finalModel)
```

```
## Data:    X dimension: 1402 18
##  Y dimension: 1402 1
## Fit method: oscorespls
```

```
## Number of components considered: 12
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X           8.308    14.97    22.46    31.92    36.72    41.27    47.00
## .outcome   12.019    13.09    13.19    13.28    13.81    14.93    16.75
##           8 comps  9 comps  10 comps  11 comps  12 comps
## X           50.70    54.04     57.10     62.31     67.54
## .outcome    19.99    22.55     23.94     24.03     24.04
```

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




```
# view performance on the test set (RMSE)
pls_pred <- predict(pls.fit, data.frame(x_test))
test_rmse_pls <- sqrt(mean((pls_pred - y_test))^2)
test_rmse_pls
```

```
## [1] 1.030057
```

RMSE = 1.0300573

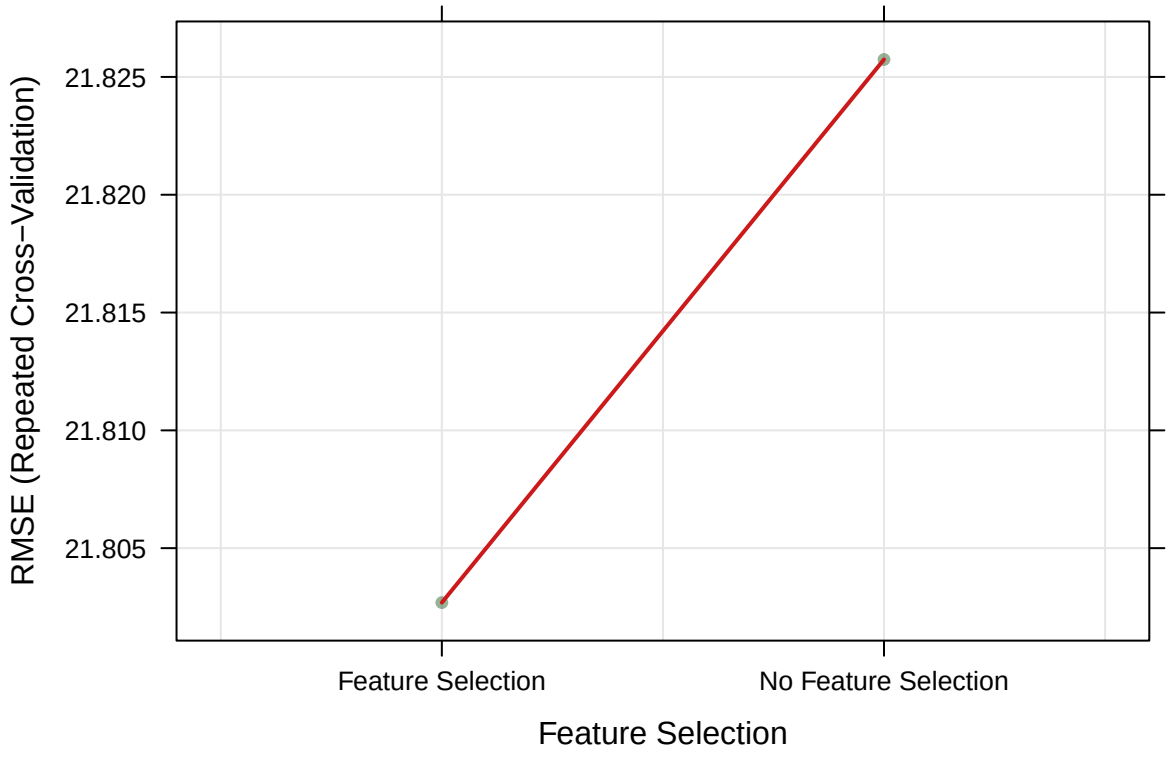
Fit a Generalized additive model (GAM):

```r
set.seed(4195)
# fit GAM using default setting
gam.fit <- train(x, y, # test dataset
                 method = "gam",
                 trControl = ctrl)

# view the model summary
summary(gam.fit$finalModel)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender1 + race2 + race3 + race4 + smoking1 + smoking2 +
##     hypertension1 + diabetes1 + vaccine1 + severity1 + study2 +
##     study3 + s(age) + s(SBP) + s(LDL) + s(bmi) + s(height) +
##     s(weight)
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   42.69008    1.72989  24.678  < 2e-16 ***
## gender1       -6.03913    1.11580  -5.412 7.33e-08 ***
## race2         -2.36813    2.58265  -0.917  0.35934
## race3          0.93986    1.39612   0.673  0.50093
## race4         -1.38293    1.83233  -0.755  0.45053
## smoking1       5.06298    1.25663   4.029 5.91e-05 ***
## smoking2      10.97420    1.86930   5.871 5.43e-09 ***
## hypertension1  3.51457    1.12341   3.128  0.00179 **
## diabetes1     -0.03949    1.63164  -0.024  0.98069
## vaccine1      -6.89201    1.15163  -5.985 2.76e-09 ***
## severity1      9.97838    1.90893   5.227 1.99e-07 ***
## study2         2.29649    1.41166   1.627  0.10401
## study3        -1.06735    1.70593  -0.626  0.53164
## ---
## 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)    4.614e-08      9  0.00  0.3983
## s(SBP)    2.185e-08      9  0.00  0.5511
## s(LDL)    3.601e-08      9  0.00  0.4010
## s(bmi)    8.925e+00      9 72.01  <2e-16 ***
## s(height) 6.577e-09      9  0.00  0.9094
## s(weight) 5.300e+00      9  1.24  0.0435 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.383   Deviance explained = 39.5%
## GCV = 441.96  Scale est. = 433.38    n = 1402
```

```
plot(gam.fit)
```



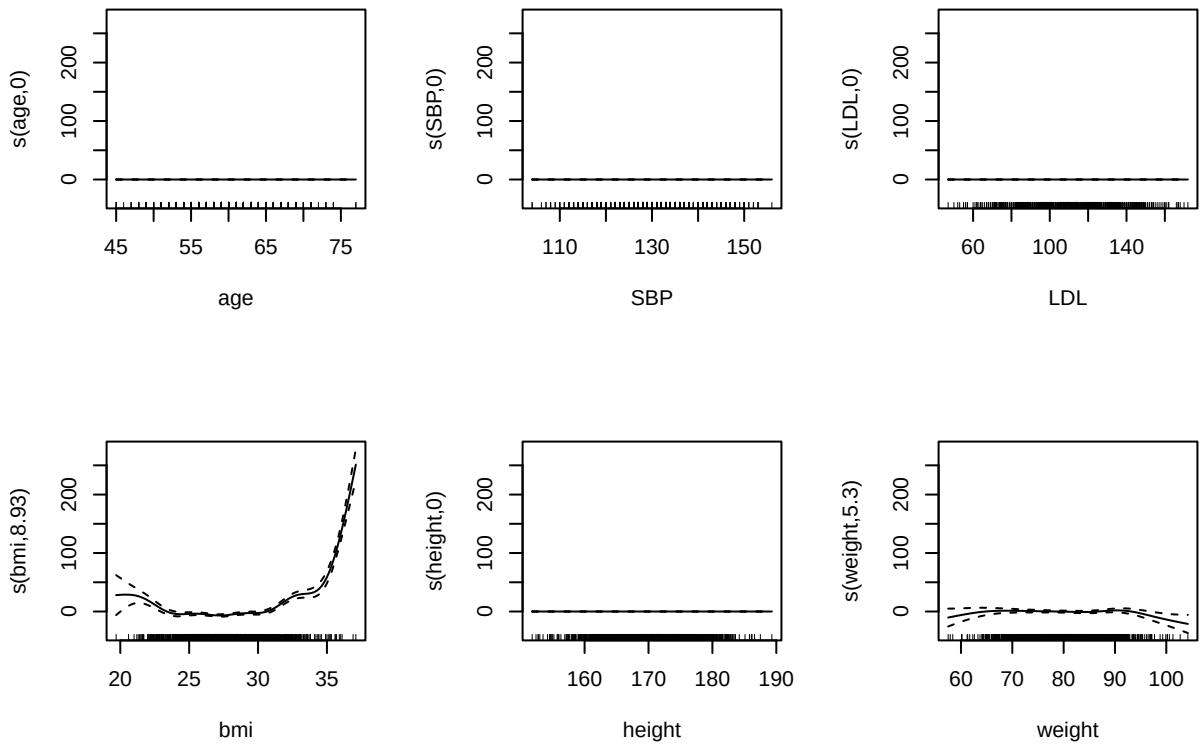

```
gam.fit$bestTune
```

```
##   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 + study2 +
##     study3 + s(age) + s(SBP) + s(LDL) + s(bmi) + s(height) +
##     s(weight)
##
## Estimated degrees of freedom:
## 0.00 0.00 0.00 8.93 0.00 5.30  total = 27.23
##
## GCV score: 441.9595
```

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




```
par(mfrow=c(1, 1))
# view performance on the test set (RMSE)
gam_pred <- predict(gam.fit, data.frame(x_test))
test_rmse_gam <- sqrt(mean((gam_pred - y_test))^2)
test_rmse_gam
```

```
## [1] 0.3650951
```

RMSE for gaml= 0.3650951

Fit a Multivariate adaptive regression spline (MARS) model:

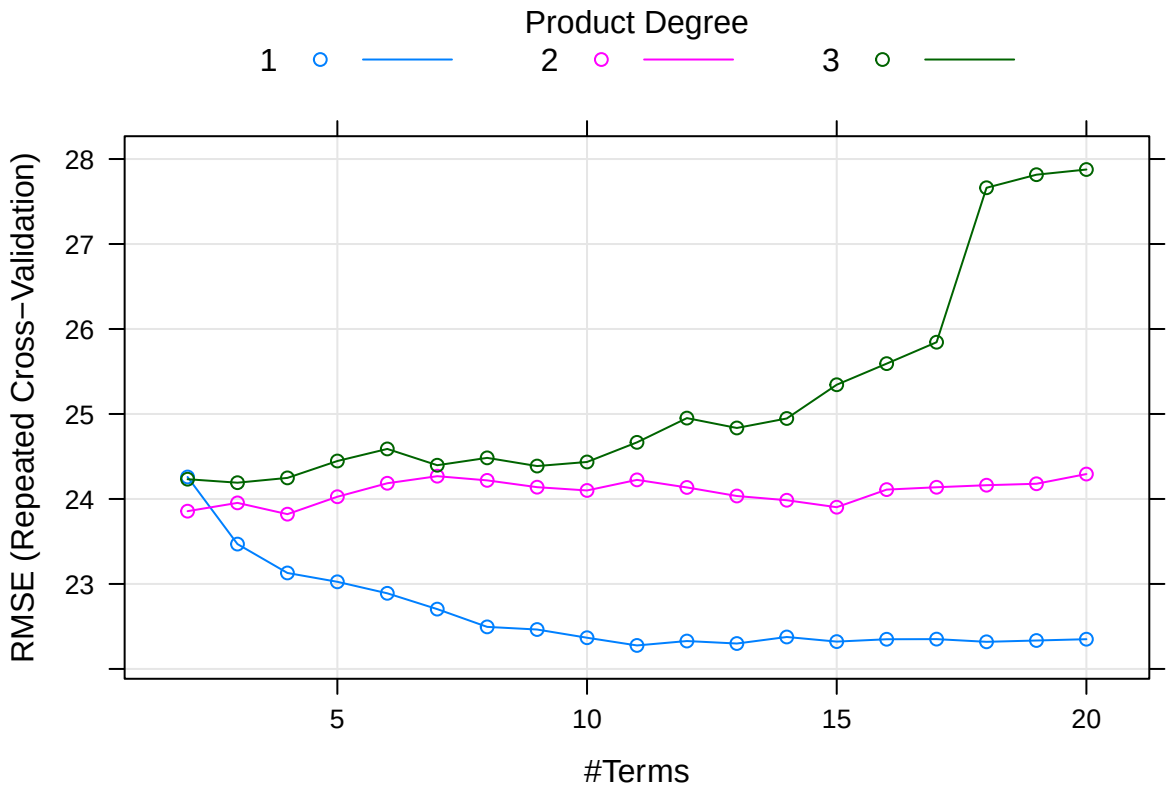
```
set.seed(4195)
# create grid of all possible pairs that can take degree and nprune values
mars_grid <- expand.grid(degree = 1:3, # number of possible product hinge functions in 1 term
                         nprune = 2:20) # Upper bound of number of terms in model
mars.fit <- train(x,
                  y,# training dataset
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl)
```

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

```
# view the model summary
summary(mars.fit$finalModel)
```

```
## Call: earth(x=matrix[1402,18], y=c(30,118,49,41,...), keepxy=TRUE, degree=1,
##             nprune=11)
##
##             coefficients
## (Intercept)   -34.958407
## gender1        -5.877791
## smoking2        9.425191
## vaccine1       -6.573231
## severity1       9.512094
## h(bmi-23.9)     9.037401
## h(bmi-28.6)     5.225776
## h(32-bmi)       9.704445
## h(bmi-32)       7.002485
## h(bmi-34.3)   -51.894845
## h(bmi-35.2)   156.582406
##
## Selected 11 of 23 terms, and 5 of 18 predictors (nprune=11)
## Termination condition: Reached nk 37
## Importance: bmi, vaccine1, gender1, smoking2, severity1, age-unused, ...
## Number of terms at each degree of interaction: 1 10 (additive model)
## GCV 447.0747    RSS 608162.2    GRSq 0.3642685    RSq 0.3822897
```

```
plot(mars.fit)
```



```
varImp(mars.fit)
```

```
## earth variable importance
##
##           Overall
## bmi       100.000
## vaccine1   17.904
## gender1    11.429
## smoking2    5.528
## severity1   0.000
```

```
# view performance on the test set (RMSE)
mars_pred <- predict(mars.fit, data.frame(x_test))
test_rmse_mars <- sqrt(mean((mars_pred - y_test))^2)
test_rmse_mars
```

```
## [1] 0.498946
```

RMSE = 0.498946

Model comparison:

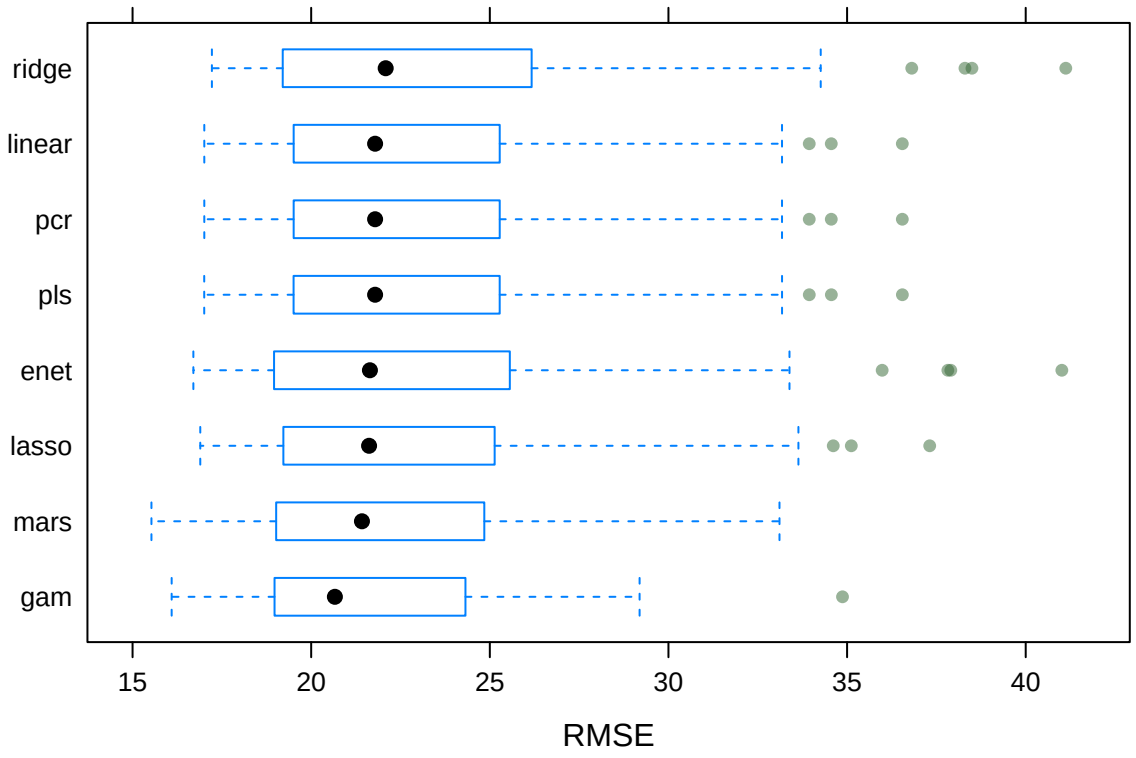
Boxplot

```
set.seed(4195)
resamp = resamples(list(
                        linear=lm.fit,
                        ridge = ridge.fit,
                        lasso = lasso.fit,
                        enet = enet.fit,
                        pcr = pcr.fit,
                        pls = pls.fit,
                        gam = gam.fit,
                        mars = mars.fit))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: linear, ridge, lasso, enet, pcr, pls, gam, mars
## Number of resamples: 50
##
## MAE
##            Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## linear 12.45434 14.36326 15.46439 15.39960 16.07453 18.33422    0
## ridge  12.51644 14.17746 15.06231 15.26539 16.16693 18.84820    0
## lasso  12.33922 14.21415 15.22249 15.24308 15.99849 18.22015    0
## enet   12.24497 13.83944 14.84224 14.99155 15.87681 18.51202    0
```

```
## pcr    12.45434 14.36326 15.46439 15.39960 16.07453 18.33422    0
## pls    12.45441 14.36526 15.46680 15.40030 16.07406 18.34113    0
## gam    11.79423 13.82056 14.66266 14.56811 15.29117 17.21988    0
## mars   10.94788 13.88167 14.64942 14.54511 15.38748 17.06443    0
##
## RMSE
##            Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## linear 17.00778 19.53498 21.78932 23.17550 25.20443 36.54660    0
## ridge  17.22035 19.23008 22.08552 24.09320 26.04269 41.12200    0
## lasso  16.89531 19.27019 21.62040 23.15185 25.12814 37.31288    0
## enet   16.70205 19.04394 21.64395 23.56012 25.43298 41.01151    0
## pcr    17.00778 19.53498 21.78932 23.17550 25.20443 36.54660    0
## pls    17.00605 19.53420 21.78871 23.17539 25.20385 36.54796    0
## gam    16.09422 19.04722 20.66361 21.80269 24.29060 34.87183    0
## mars   15.52845 19.02786 21.42099 22.27648 24.80028 33.10796    0
##
## Rsquared
##               Min.    1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## linear 0.06296364 0.14590446 0.1926016 0.2129520 0.2526136 0.4770537    0
## ridge  0.02164192 0.09911699 0.1239995 0.1346198 0.1703316 0.3108875    0
## lasso  0.06270494 0.14415659 0.1975179 0.2118501 0.2536393 0.4576479    0
## enet   0.04215195 0.13506557 0.1664189 0.1765881 0.2217879 0.3575522    0
## pcr    0.06296364 0.14590446 0.1926016 0.2129520 0.2526136 0.4770537    0
## pls    0.06300174 0.14592884 0.1926212 0.2129513 0.2525843 0.4769383    0
## gam    0.03495173 0.18162408 0.2514815 0.2958699 0.3515494 0.6687155    0
## mars   0.03294532 0.14345855 0.2403353 0.2747823 0.3541213 0.6495820    0
```

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



choose GAM model