final_project

wz2631 rz2614 jn2855

2023-04-30

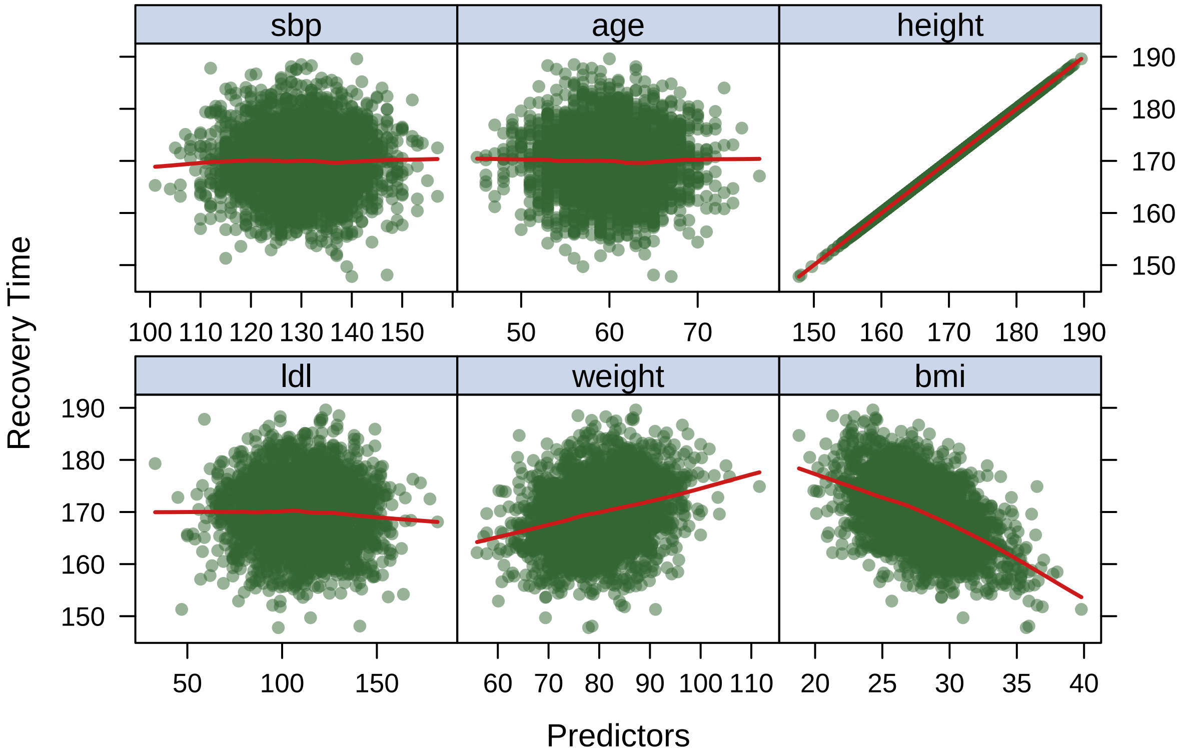
Data preparation

```r
# draw 2 random samples of 2000 participants
load("./recovery.Rdata")
set.seed(2631)
dat1 <- dat[sample(1:10000, 2000),] %>%
  janitor::clean_names() %>%
  na.omit()
set.seed(2855)
dat2 <- dat[sample(1:10000, 2000),] %>%
  janitor::clean_names() %>%
  na.omit()
dat <- rbind.fill(dat1, dat2) %>%
   dplyr::select(-id) %>%
  unique() %>% mutate( gender=fct_recode(factor(gender),male='1',female='0'),
    race=fct_recode(factor(race),white='1',asian='2',black='3',hispanic='4'),
    smoking=fct_recode(factor(smoking),never='0',former='1',current='2'),
    hypertension=factor(hypertension),
    diabetes=factor(diabetes),
    vaccine=factor(vaccine),
    severity=factor(severity),
    study=factor(study),
    recovery_t = if_else(recovery_time <= 30, 't1','t2'),
    recovery_t = factor(recovery_t)
  )
```

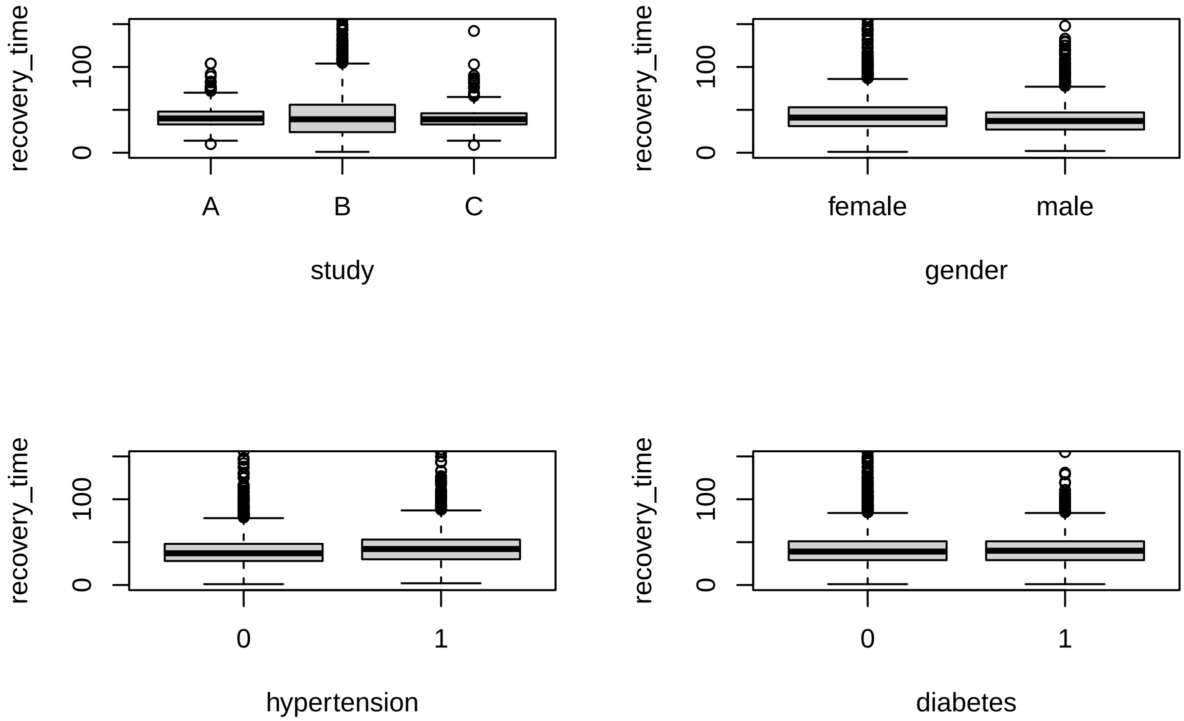
```r
#data partition
set.seed(2023)
train_index=createDataPartition(y = dat$recovery_time,
                                p = 0.8,
                                list = FALSE)
train_dat=dat[train_index,]
test_dat=dat[-train_index,]
#training data
train_data=dat[train_index,]
x1 = model.matrix(recovery_time~., data=dat)[train_index,-1]
y1=dat$recovery_time[train_index]
#testing data
test_data=dat[-train_index,]
x2=model.matrix(recovery_time~., data=dat)[-train_index,-1]
y2=dat$recovery_time[-train_index]
```

```
#exploratory analysis and data visualization
visualization = train_dat %>%
  mutate(study=case_when(
    study == "A" ~ 1,
    study == "B" ~ 2,
    study == "C" ~ 3
  )) %>%
  dplyr::select(ldl,weight,bmi,sbp,age,height)
non_numeric= sapply(visualization, function(x) !is.numeric(x))
visualization[, non_numeric] = lapply(visualization[, non_numeric], as.numeric)
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)

featurePlot(x = visualization[ ,1:6],
            y = visualization[ ,6],
            plot = "scatter",
            span = .5,
            labels = c("Predictors", "Recovery Time"),
            main = "Figure 1. the relationship between predictors and recovery time",
            type = c("p", "smooth"))
```

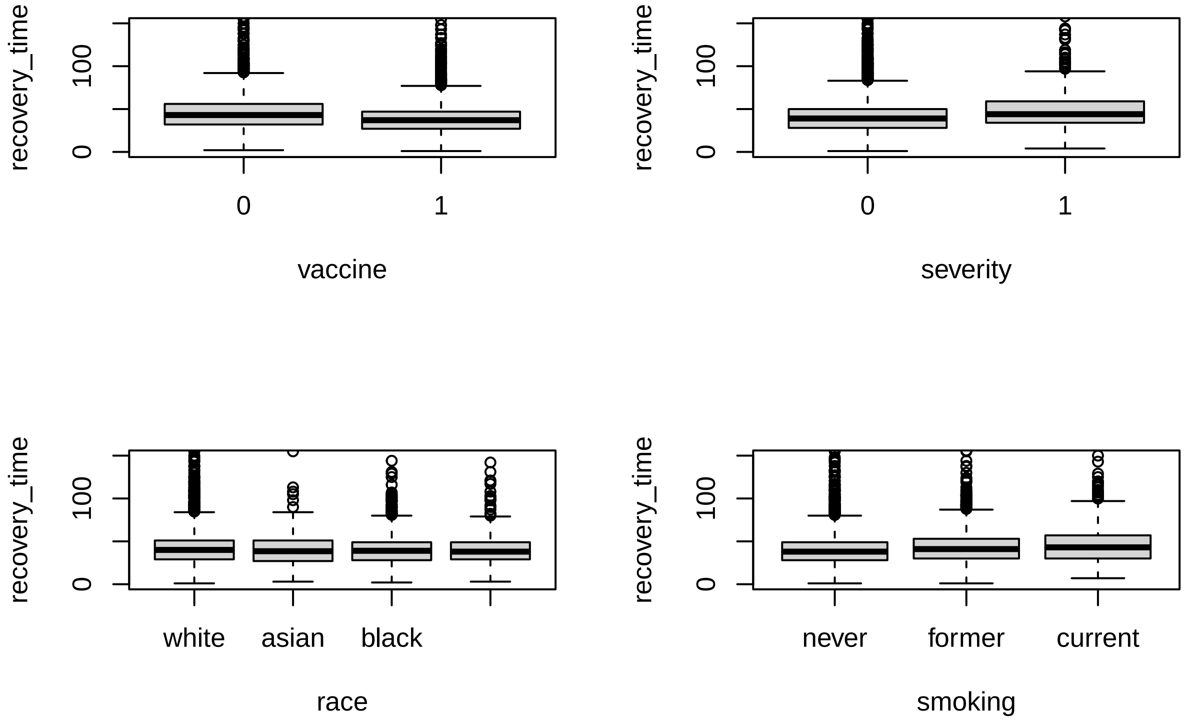
Figure 1. the relationship between predictors and recovery time



```
par(mfrow=c(2,2))
boxplot(recovery_time~study, data=dat, xlab="study", ylim=c(0,150))
boxplot(recovery_time~gender, data=dat, xlab="gender", ylim=c(0,150))
boxplot(recovery_time~hypertension, data=dat, xlab="hypertension", ylim=c(0,150))
boxplot(recovery_time~diabetes, data=dat, xlab="diabetes", ylim=c(0,150))
```

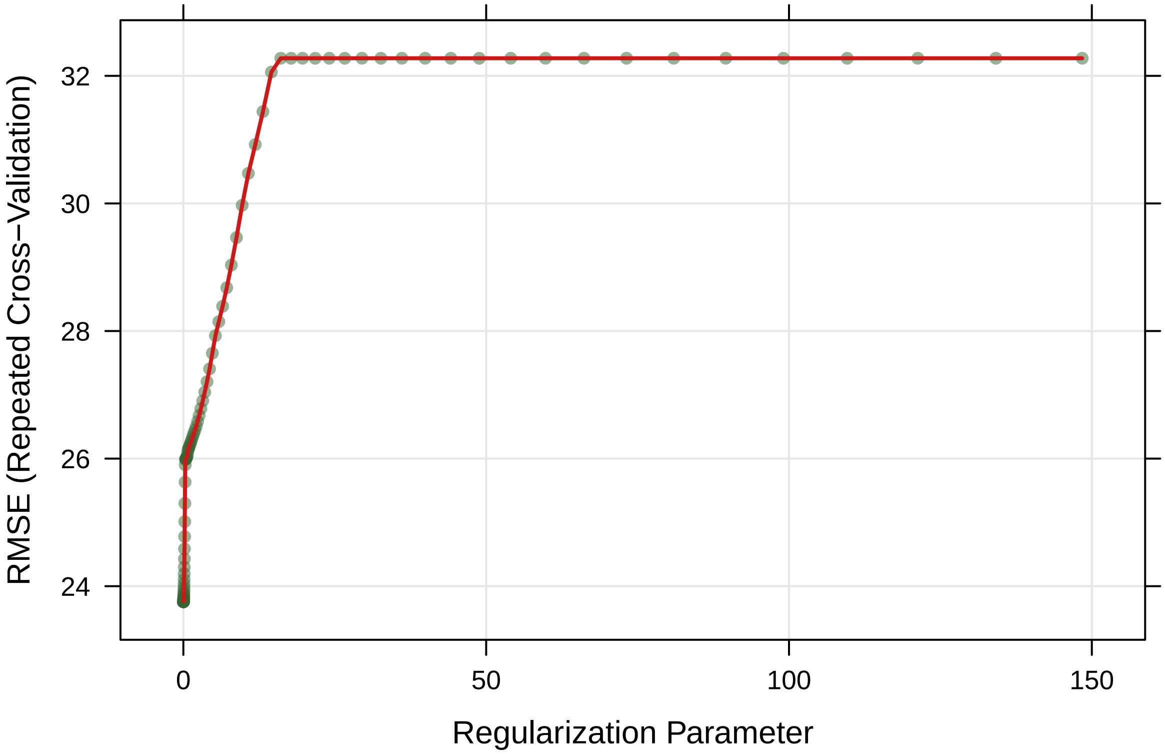


```
boxplot(recovery_time~vaccine, data=dat, xlab="vaccine", ylim=c(0,150))
boxplot(recovery_time~severity, data=dat, xlab="severity", ylim=c(0,150))
boxplot(recovery_time~race, data=dat, xlab="race", ylim=c(0,150))
boxplot(recovery_time~smoking, data=dat, xlab="smoking", ylim=c(0,150))
```



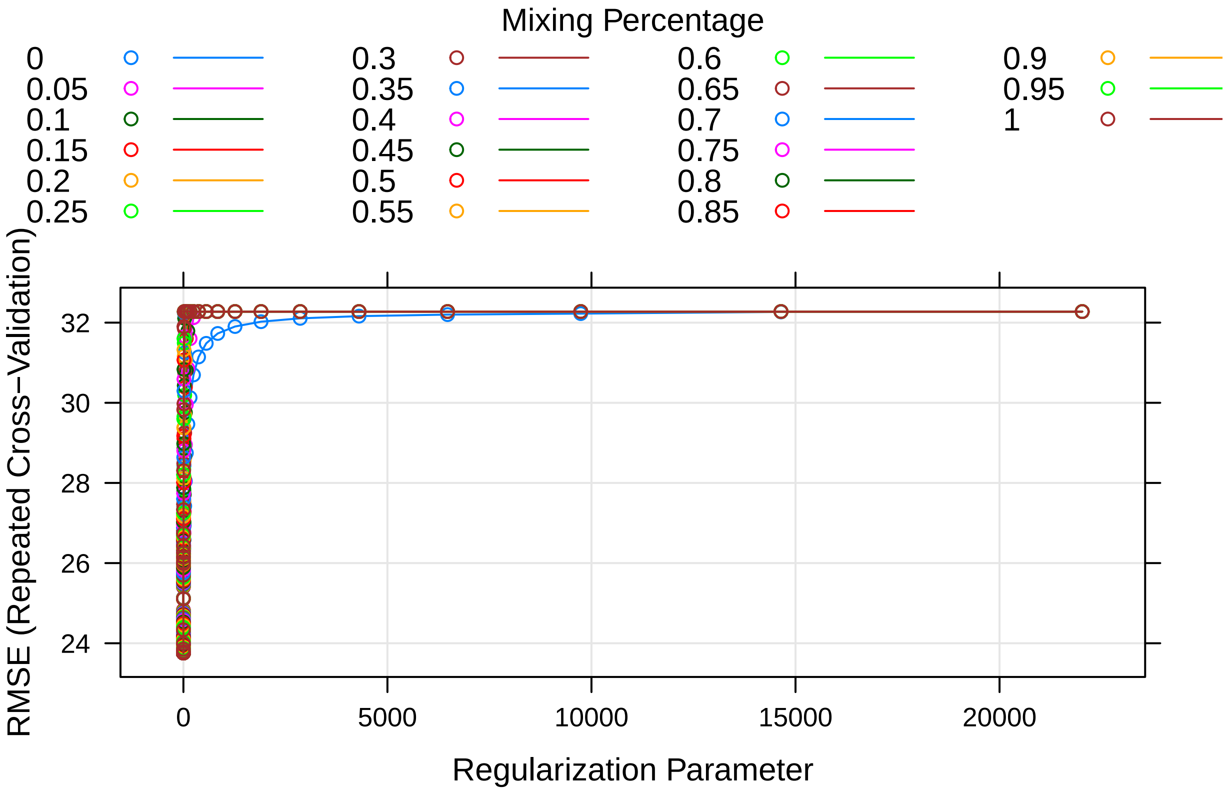
```
#linear model
set.seed(2023)
ctrl=trainControl(method = "repeatedcv", number =10, repeats = 5)
linear = train(recovery_time ~ age + gender + race + smoking + height +
                        weight + bmi + hypertension + diabetes + sbp + ldl +
                        vaccine + severity + study,
               data = train_dat,
               method = "lm",
               trControl = ctrl)
summary(linear$finalModel)
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -85.895 -14.897  -1.583  11.054 250.397
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.398e+03  1.372e+02 -24.774  < 2e-16 ***
## age             1.790e-01  1.272e-01   1.406   0.1597
## gendermale     -5.601e+00  1.008e+00  -5.556 3.02e-08 ***
## raceasian      -2.674e-01  2.354e+00  -0.114   0.9096
## raceblack      -2.943e+00  1.272e+00  -2.314   0.0207 *
## racehispanic   -1.030e+00  1.734e+00  -0.594   0.5525
## smokingformer   5.233e+00  1.135e+00   4.611 4.19e-06 ***
## smokingcurrent  7.340e+00  1.696e+00   4.328 1.56e-05 ***
## height          1.973e+01  8.058e-01  24.479  < 2e-16 ***
## weight         -2.134e+01  8.486e-01 -25.140  < 2e-16 ***
## bmi             6.424e+01  2.427e+00  26.468  < 2e-16 ***
## hypertension1   2.655e+00  1.682e+00   1.578   0.1146
## diabetes1       9.470e-01  1.373e+00   0.690   0.4904
## sbp             5.394e-02  1.096e-01   0.492   0.6226
## ldl            -4.210e-02  2.686e-02  -1.567   0.1171
## vaccine1       -8.254e+00  1.028e+00  -8.025 1.46e-15 ***
## severity1       8.467e+00  1.630e+00   5.195 2.19e-07 ***
## studyB          6.723e+00  1.308e+00   5.139 2.95e-07 ***
## studyC          3.407e-01  1.595e+00   0.214   0.8309
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.97 on 2859 degrees of freedom
## Multiple R-squared:  0.3235, Adjusted R-squared:  0.3192
## F-statistic: 75.94 on 18 and 2859 DF,  p-value: < 2.2e-16
RMSE
## function (pred, obs, na.rm = FALSE)
## sqrt(mean((pred - obs)^2, na.rm = na.rm))
## <bytecode: 0x7f8690bceff0>
## <environment: namespace:caret>
test_pred1=predict(linear,newdata = test_dat)
rmse1=sqrt(mean((test_pred1-test_dat$recovery_time)**2))
rmse1
## [1] 23.74624
```

```
#lasso
set.seed(2023)
ctrl=trainControl(method = "repeatedcv", number =10, repeats = 5)
lasso=train(x1,y1,
            method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(-5, 5, length = 100))),
                   trControl = ctrl)
coef(lasso$finalModel, lasso$bestTune$lambda)
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)    -2.946231e+03
## age             1.216632e-01
## gendermale     -3.415100e+00
## raceasian       1.035131e+00
## raceblack      -2.068045e+00
## racehispanic   -9.367240e-01
## smokingformer   3.432322e+00
## smokingcurrent  6.523183e+00
## height          1.698483e+01
## weight         -1.833806e+01
## bmi             5.533357e+01
## hypertension1   1.243674e+00
## diabetes1       4.114282e-01
## sbp              4.419412e-02
## ldl             -3.690833e-02
## vaccine1        -5.145263e+00
## severity1        5.402671e+00
## studyB           1.220249e+01
## studyC           .
## recovery_tt2     2.949668e+01
lasso$bestTunetest_pred2=predict(lasso,newdata=x2)
pred_lasso=predict(lasso, newx = x2, s = lasso$lambda.min)
rmse_lasso= sqrt(mean((pred_lasso-y2)**2))
rmse_lasso
## [1] 34.30053
coef=coef(lasso, s = lasso$lambda.min)
n.pred=sum(coef[-1] != 0)
n.pred
## [1] 0
plot(lasso)
```

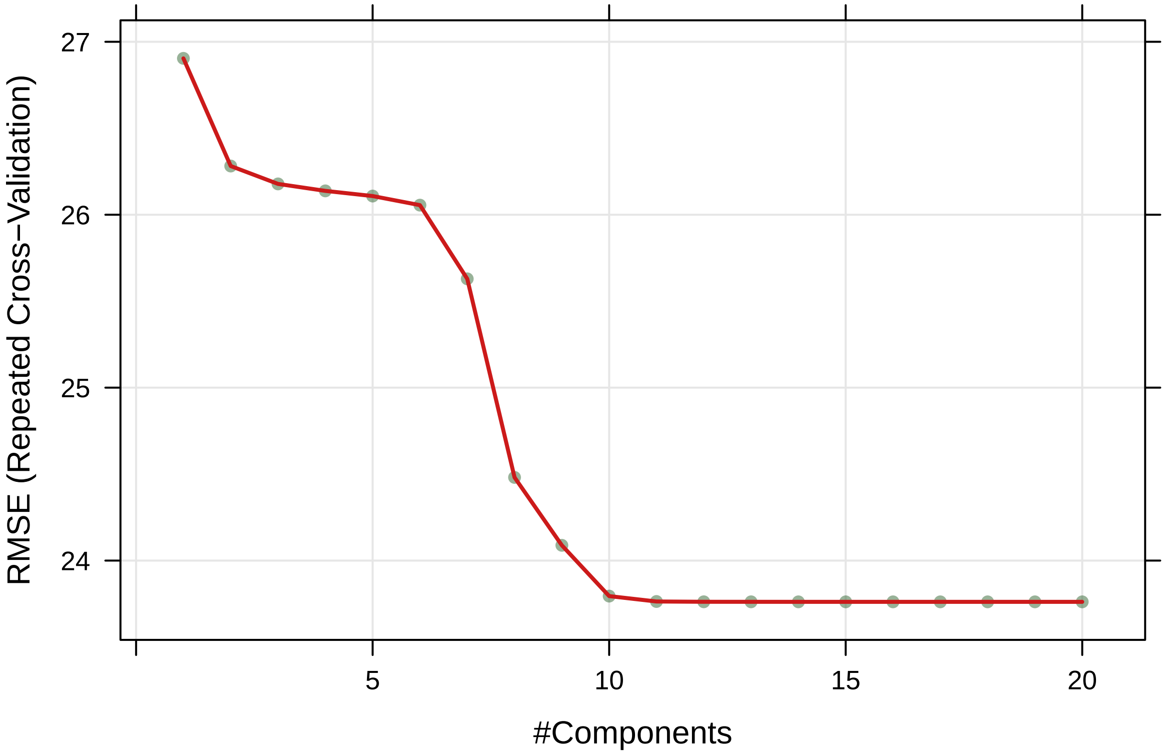


```
#elastic net
set.seed(2023)
elastic_net=train(x1, y1,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                         lambda = exp(seq(10, -10, length = 50))),
                  trControl = ctrl)
```

```
elastic_net$bestTune
##     alpha      lambda
## 411   0.4 0.002689588
test_pred_elastic=predict(elastic_net, newdata = x2)
rmse_elastic=sqrt(mean((test_pred_elastic - test_dat$recovery_time)**2))
rmse_elastic
## [1] 19.68248
plot(elastic_net)
```



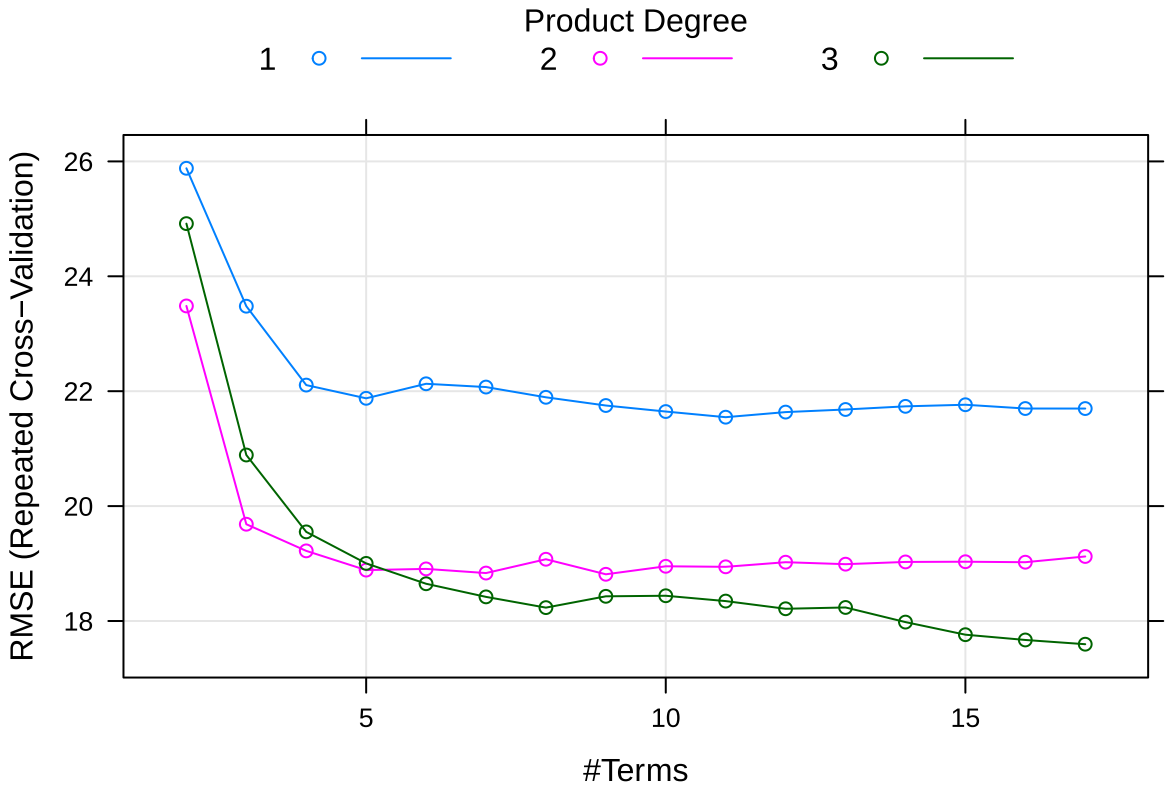
```
#pls
set.seed(2023)
pls=train(x1, y1,
          method = "pls",
          tuneGrid = data.frame(ncomp = 1:20),
          trControl = ctrl,
          preProcess = c("center", "scale"))

summary(pls$finalModel)
## Data:    X dimension: 2878 19
##  Y dimension: 2878 1
## Fit method: oscorespls
## Number of components considered: 18
## TRAINING: % variance explained
##            1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X            8.425    15.83    24.05    30.50    34.82    41.85    46.44
## .outcome   31.437    34.49    34.93    35.15    35.36    35.67    38.49
##           8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X           48.98    52.73     54.76     59.54     65.03     70.23     75.72
## .outcome    44.60    46.18     47.24     47.37     47.37     47.37     47.37
##           15 comps  16 comps  17 comps  18 comps
## X            80.18     85.20     89.54     94.70
## .outcome     47.37     47.37     47.37     47.37
test_pred_pls=predict(pls, newdata = x2)
rmse_pls=sqrt(mean((test_pred_pls - test_dat$recovery_time)**2))
rmse_pls
## [1] 19.75362
plot(pls)
```



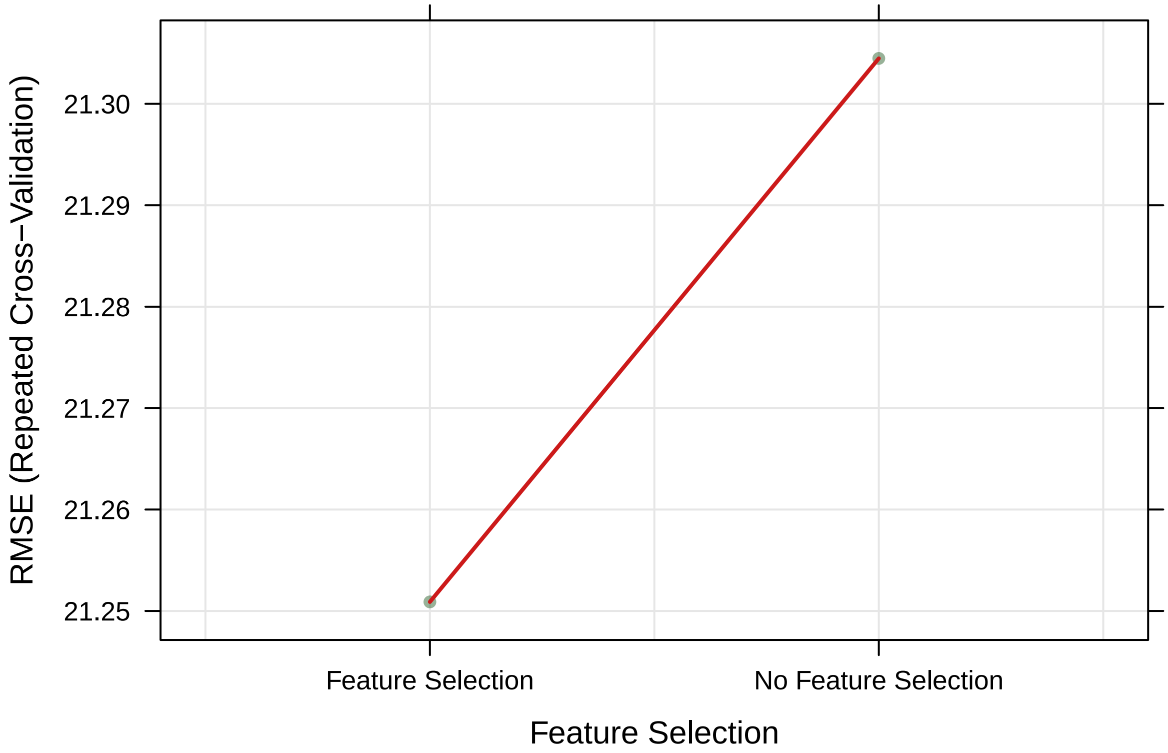
```
#mars
set.seed(2023)
mars_grid = expand.grid(degree = 1:3,
                        nprune = 2:17)
mars = train(x1, y1,
             method = "earth",
             tuneGrid = mars_grid,
             trControl = ctrl)
kable(mars$bestTune,"simple")
```

	nprune	degree
48	17	3

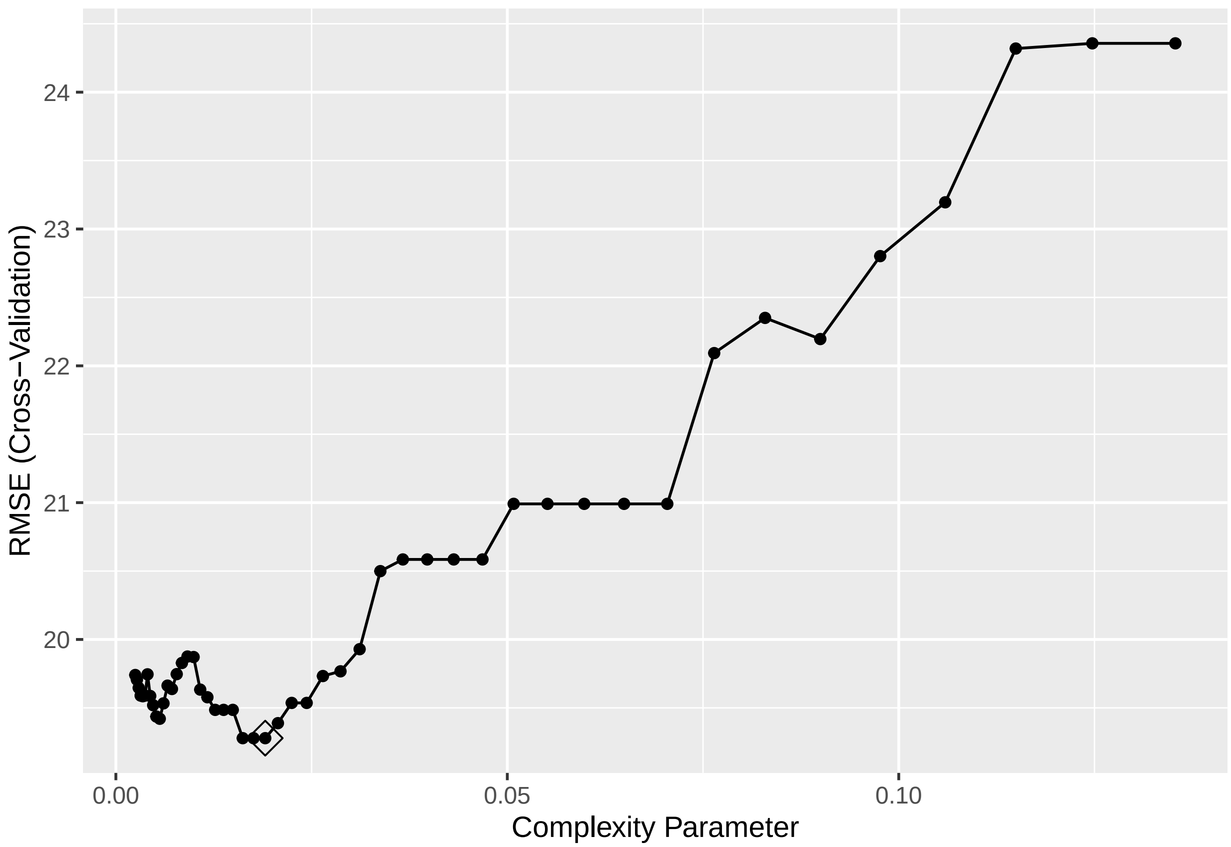
```
coef(mars$finalModel)
##                           (Intercept)                           h(bmi-31.4)
##                             21.054679                              9.041933
##                          recovery_tt2                 studyB * recovery_tt2
##                             20.720795                              7.516211
##      h(bmi-31.4) * severity1 * studyB      h(age-63) * h(bmi-31.4) * studyB
##                             20.672796                             49.718869
##     h(bmi-26) * studyB * recovery_tt2     h(26-bmi) * studyB * recovery_tt2
##                              3.331891                              7.323239
##   h(bmi-32.2) * studyB * recovery_tt2     h(bmi-31.4) * h(sbp-136) * studyB
##                             50.562930                             -3.451793
##     h(bmi-31.4) * h(136-sbp) * studyB     h(bmi-31.4) * h(ldl-119) * studyB
##                             -1.480191                              1.013023
##      vaccine1 * studyB * recovery_tt2 smokingcurrent * h(bmi-31.4) * studyB
##                             -8.417471                             18.549723
##      h(age-64) * h(bmi-31.4) * studyB      h(age-61) * h(bmi-31.4) * studyB
##                            -28.483749                            -15.197017
##     gendermale * h(bmi-31.4) * studyB
##                             -9.865810
test_pred_mars=predict(mars, newdata = x2)
rmse_mars=sqrt(mean((test_pred_mars - test_dat$recovery_time)**2))
rmse_mars
## [1] 15.78523
summary(mars)
## Call: earth(x=matrix[2878,19], y=c(15,56,42,62,4...), keepxy=TRUE, degree=3,
##             nprune=17)
##
##                                       coefficients
## (Intercept)                              21.054679
## recovery_tt2                             20.720795
## h(bmi-31.4)                               9.041933
## studyB * recovery_tt2                     7.516211
## vaccine1 * studyB * recovery_tt2         -8.417471
## gendermale * h(bmi-31.4) * studyB        -9.865810
## smokingcurrent * h(bmi-31.4) * studyB    18.549723
## h(bmi-31.4) * severity1 * studyB         20.672796
## h(26-bmi) * studyB * recovery_tt2         7.323239
## h(bmi-26) * studyB * recovery_tt2         3.331891
## h(bmi-32.2) * studyB * recovery_tt2      50.562930
## h(age-61) * h(bmi-31.4) * studyB        -15.197017
## h(age-63) * h(bmi-31.4) * studyB         49.718869
## h(age-64) * h(bmi-31.4) * studyB        -28.483749
## h(bmi-31.4) * h(sbp-136) * studyB        -3.451793
## h(bmi-31.4) * h(136-sbp) * studyB        -1.480191
## h(bmi-31.4) * h(ldl-119) * studyB         1.013023
##
## Selected 17 of 26 terms, and 10 of 19 predictors (nprune=17)
## Termination condition: Reached nk 39
## Importance: bmi, studyB, recovery_tt2, age, severity1, sbp, smokingcurrent, ...
## Number of terms at each degree of interaction: 1 2 1 13
## GCV 244.2741    RSS 683133    GRSq 0.7714824    RSq 0.7777926
plot(mars)
```



```
#gam
gam = train(x1, y1,
                 method = "gam",
                 trControl = ctrl,
                 control = gam.control(maxit = 200))
summary(gam$finalModel)
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gendermale + raceblack + racehispanic + smokingformer +
##     smokingcurrent + hypertension1 + diabetes1 + vaccine1 + severity1 +
##     studyB + studyC + recovery_tt2 + s(age) + s(sbp) + s(ldl) +
##     s(bmi) + s(height) + s(weight)
##
## Parametric coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       18.3728     1.4255  12.889  < 2e-16 ***
## gendermale        -2.9840     0.7735  -3.858 0.000117 ***
## raceblack         -1.2701     0.9647  -1.317 0.188071
## racehispanic    -0.6079     1.3147  -0.462 0.643825
## smokingformer    3.4597     0.8670   3.991 6.76e-05 ***
## smokingcurrent   7.1847     1.2936   5.554 3.05e-08 ***
## hypertension1    2.2592     0.7697   2.935 0.003361 **
## diabetes1        0.7835     1.0468   0.748 0.454233
## vaccine1        -5.2527     0.7901  -6.648 3.54e-11 ***
## severity1        5.3175     1.2455   4.269 2.02e-05 ***
## studyB          12.3393     1.0120  12.193  < 2e-16 ***
## studyC           0.4666     1.2162   0.384 0.701252
## recovery_tt2    28.1495     0.8977  31.358  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df       F  p-value
## s(age)    7.046e-08      9   0.000    0.692
## s(sbp)    4.882e-08      9   0.000    0.936
## s(ldl)    6.690e-08      9   0.000    0.393
## s(bmi)    6.939e+00      9 158.198  < 2e-16 ***
## s(height) 6.288e+00      9   4.214 1.04e-06 ***
## s(weight) 7.616e+00      9   6.804  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.606   Deviance explained =   61%
## GCV = 426.21  Scale est. = 421.2     n = 2878
gam$df.residual
## NULL
test_pred_gam=predict(gam, newdata = x2)
rmse_gam=sqrt(mean((test_pred_gam-test_dat$recovery_time)**2))
rmse_gam
## [1] 16.24292
plot(gam)
```



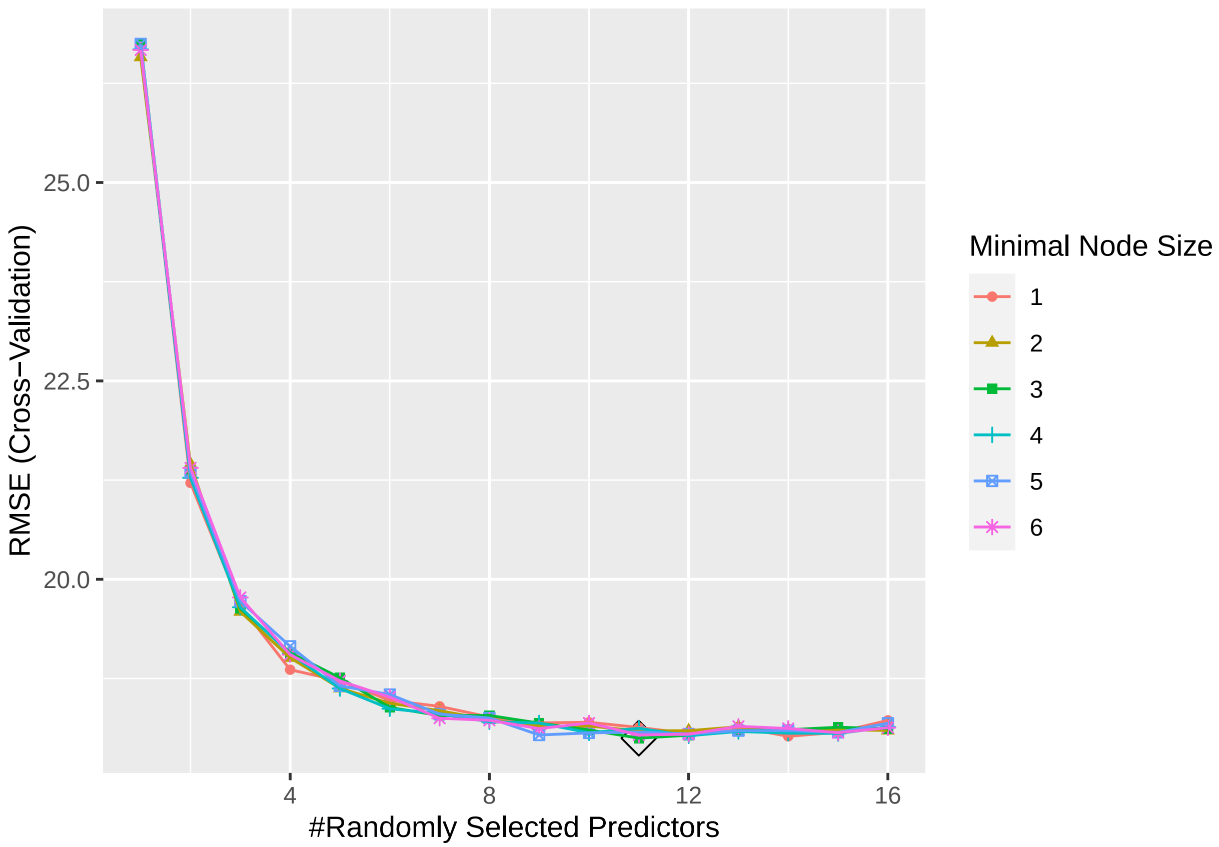
```
#tree
ctrl1=trainControl(method = "cv")
set.seed(2023)
rpart_fit <- train(recovery_time ~., data = dat[train_index,],
                   method = "rpart",
                   tuneGrid = data.frame(cp = exp(seq(-6, -2, length = 50))),
                   trControl = ctrl1)
rpart_fit$bestTune
##            cp
## 26 0.01907868
ggplot(rpart_fit,highlight = TRUE)
```



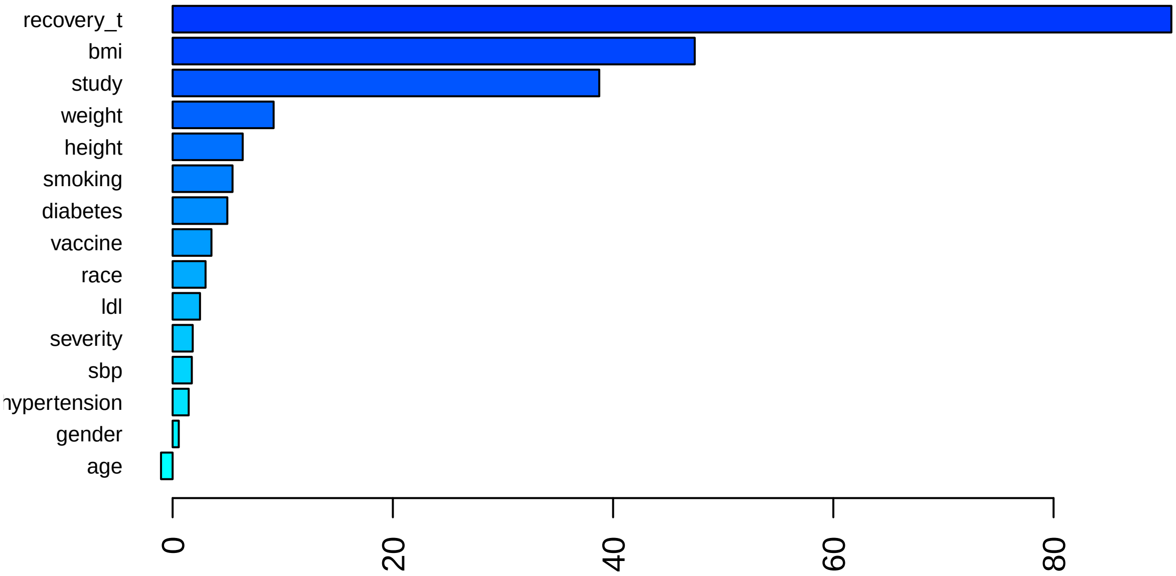
```
test_pred_tree = predict(rpart_fit, newdata = dat[-train_index, ])
rmse_tree = mean((test_pred_tree - dat$recovery_time[-train_index])**2)
rmse_tree
## [1] 218.5574
```

The cp value is 0.0190787.

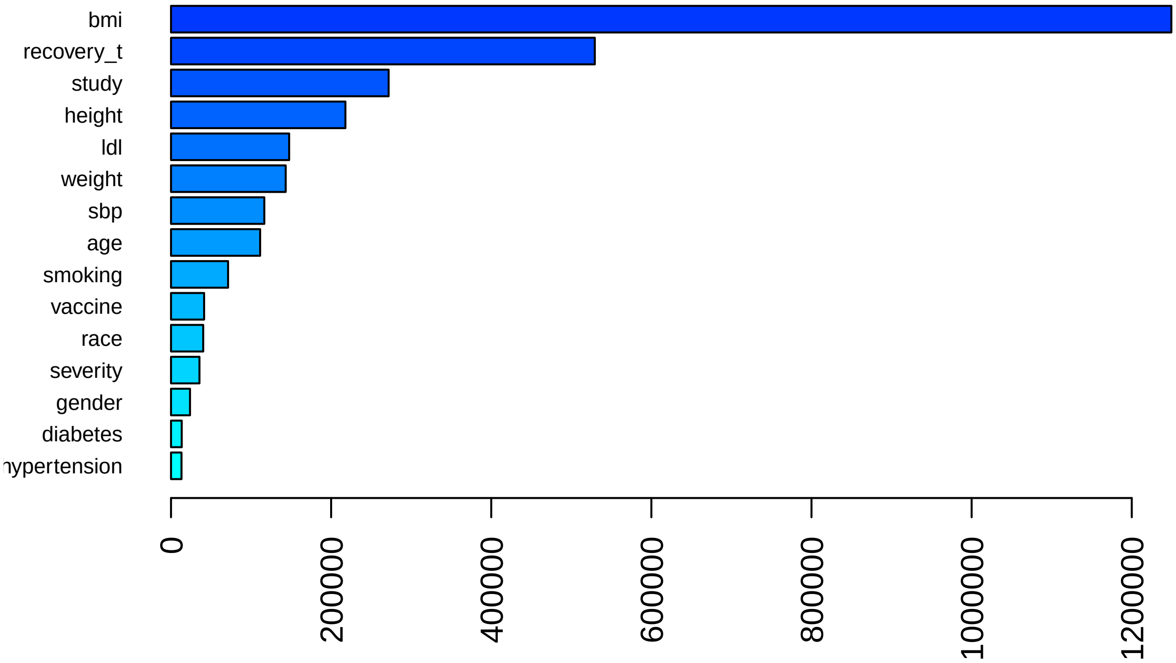
```
#rf
rf_grid=expand.grid(mtry = 1:16,
                       splitrule = "variance",
                       min.node.size = 1:6)
set.seed(2023)
rf_fit=train(recovery_time ~.,
                data = dat[train_index,],
                method = "ranger",
                tuneGrid = rf_grid,
                trControl = ctrl1)
rf_fit$bestTune
##    mtry splitrule min.node.size
## 63   11  variance             3
ggplot(rf_fit,highlight=TRUE)
```



```
set.seed(2023)
rf_perm = ranger(recovery_time ~ . ,
                 train_dat,
                 mtry = rf_fit$bestTune[[1]],
                 splitrule = "variance",
                 min.node.size = rf_fit$bestTune[[3]],
                 importance = "permutation",
                 scale.permutation.importance = TRUE)
barplot(sort(importance(rf_perm), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(19))
```



```
rf_imp = ranger(recovery_time ~ . ,
                train_dat,
                mtry = rf_fit$bestTune[[1]],
                splitrule = "variance",
                min.node.size = rf_fit$bestTune[[3]],
                importance = "impurity")
barplot(sort(importance(rf_imp), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(19))
```


bmi
recovery_t
study
height
ldl
weight
sbp
age
smoking
vaccine
race
severity
gender
diabetes
ıypertension
0
200000
400000
600000
800000
1000000
1200000