homework 4

Jeffrey Zhuohui Liang

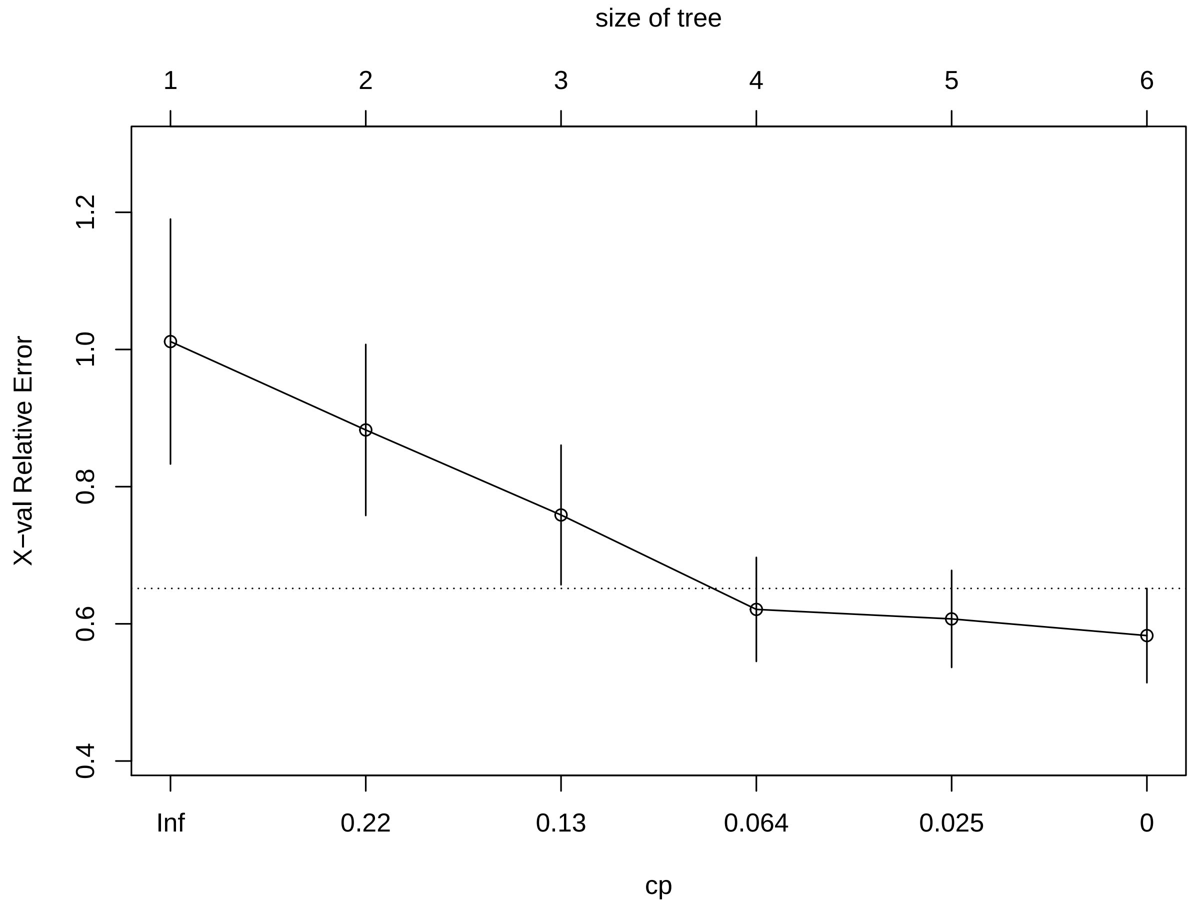
3/29/2021

1

1

```
data(Prostate)

Prostate = Prostate %>%
  janitor::clean_names()

trainindex = createDataPartition(Prostate$lpsa,p=0.8,list = F)

X_tr = model.matrix(lpsa~.,Prostate[trainindex,])[,-1]

Y_tr = Prostate[trainindex,"lpsa"]


X_ts = model.matrix(lpsa~.,Prostate[-trainindex,])[,-1]

Y_ts = Prostate[-trainindex,"lpsa"]

ctrl = trainControl(method = "repeatedcv",number = 5, repeats = 5)

pre = c("center","scale")
```

```
sng_tree = rpart(lpsa ~ .,
                     data = Prostate,
                     subset = trainindex,
                     control = rpart.control(cp = 0))

plotcp(sng_tree)
```



```
cpTable = sng_tree$cptable

minErr = which.min(cpTable[,4])

sng_tree_1se = prune(sng_tree,
                     cp=cpTable[cpTable[,4]<cpTable[minErr,4]+cpTable[minErr,5],1][1])

sng_tree_min = prune(sng_tree,cp = cpTable[minErr,1][1])

last(sng_tree_min$cptable[,2])
```

```
## [1] 5
```

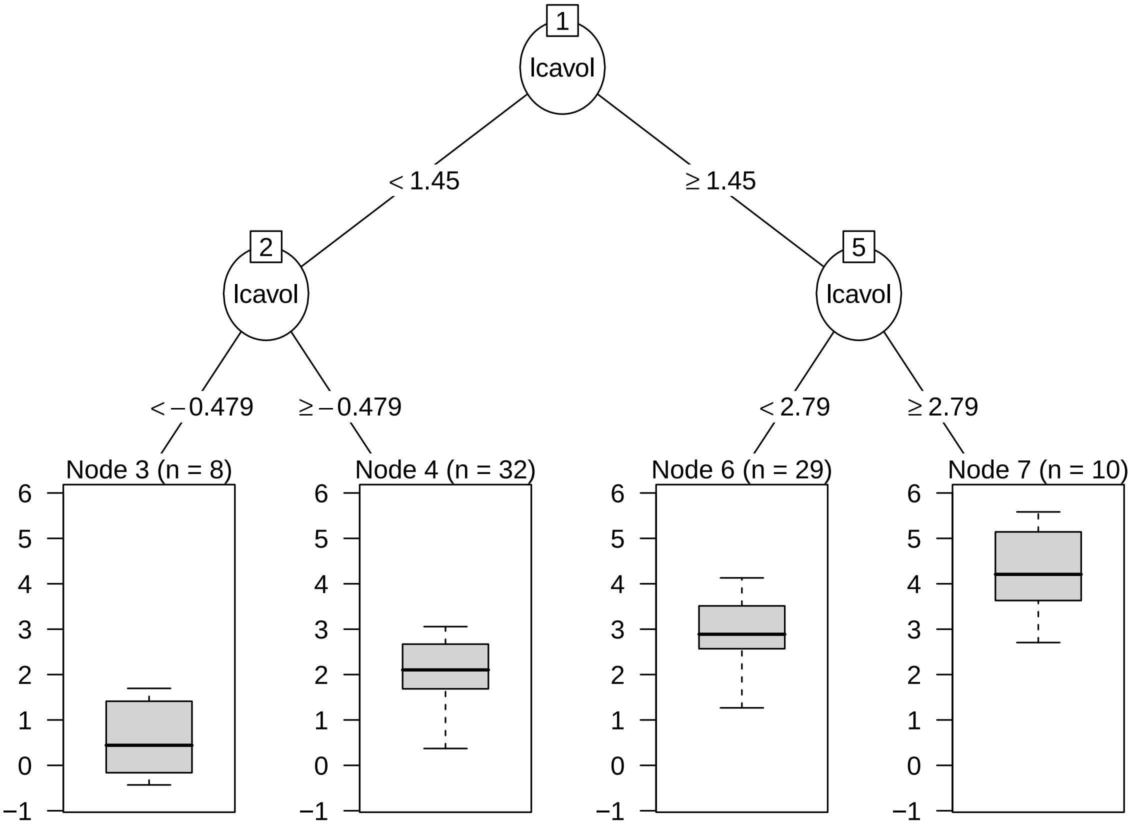
```
last(sng_tree_1se$cptable[,2])
```

```
## [1] 3
```

The number of node of minumum RMSE tree is the same as 1se tree's number of node.

2

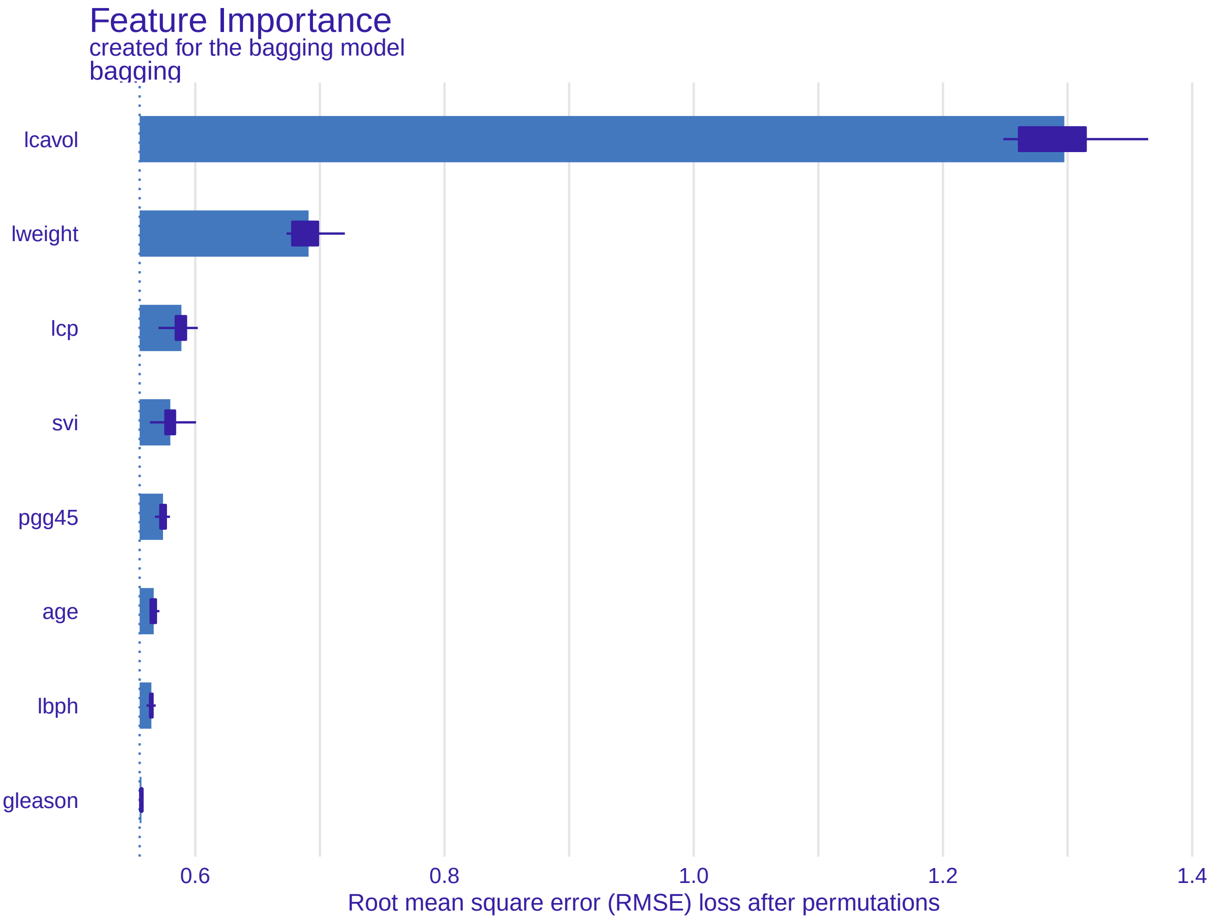
```
plot(partykit::as.party(sng_tree_1se))
```



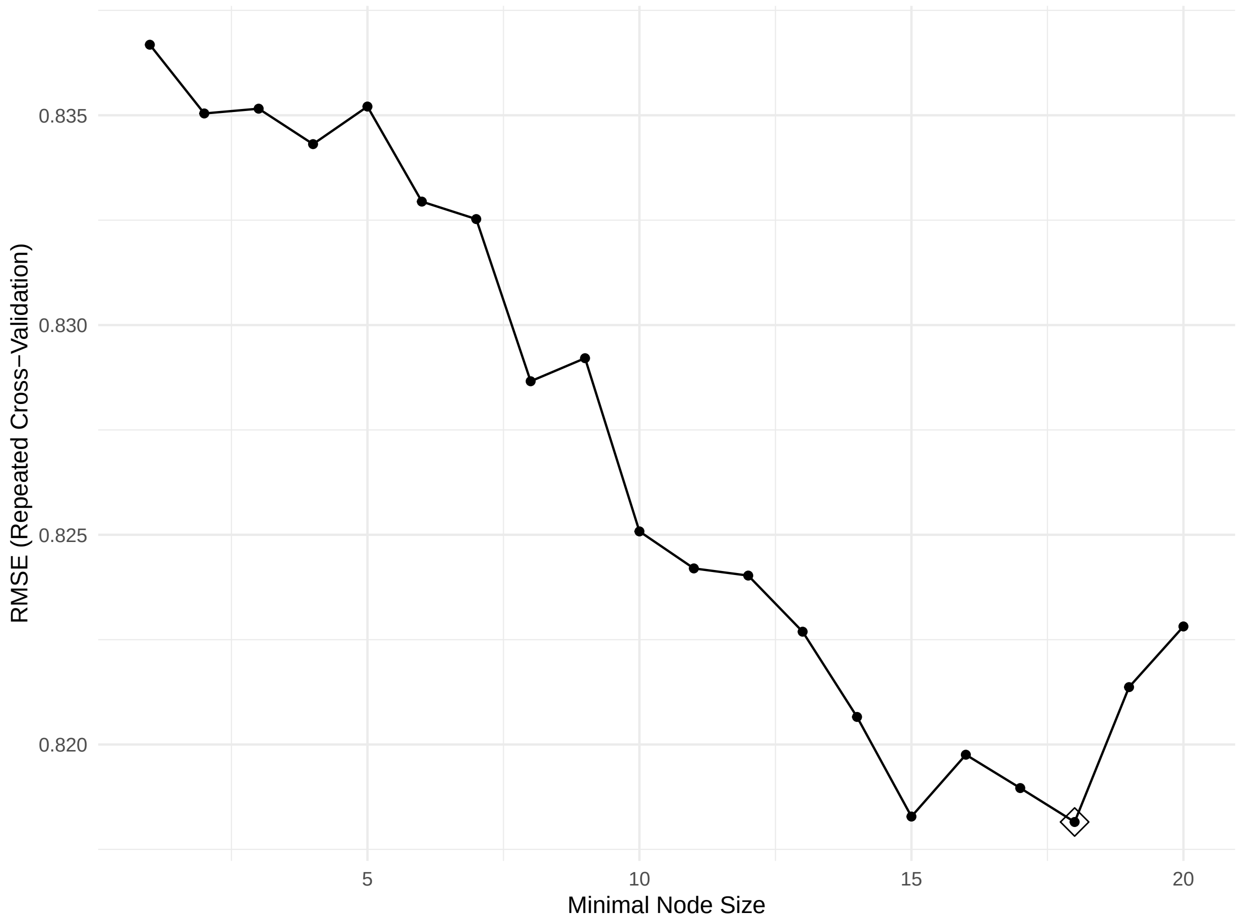
For those observation whom lcavol<-0.479 will go into terminal node 3, which has a mean response near 0.5 with 8 observations.

3

```
cl = parallel::makePSOCKcluster(5)
doParallel::registerDoParallel(cl)

sng_tree = train(X_tr,
                 Y_tr,
                 method = "rpart",
                 tuneGrid = expand.grid(cp = exp(seq(-3,-5,len = 20))),
                 preProcess = pre,
                 trControl = ctrl
                 )

bagging =
  train(X_tr,
        Y_tr,
        method = "ranger",
        tuneGrid = expand.grid(mtry = 8,
                               splitrule = "variance",
                               min.node.size = 1:20),
        metric = "RMSE",
        trControl = ctrl,
        preProcess = c("center","scale"))

bag_explain = DALEX::explain(bagging,
                             label = "bagging",
                             data=X_tr,
                             y= Y_tr,
                             verbose = F)

bag_imp = DALEX::model_parts(bag_explain)

plot(bag_imp)
```



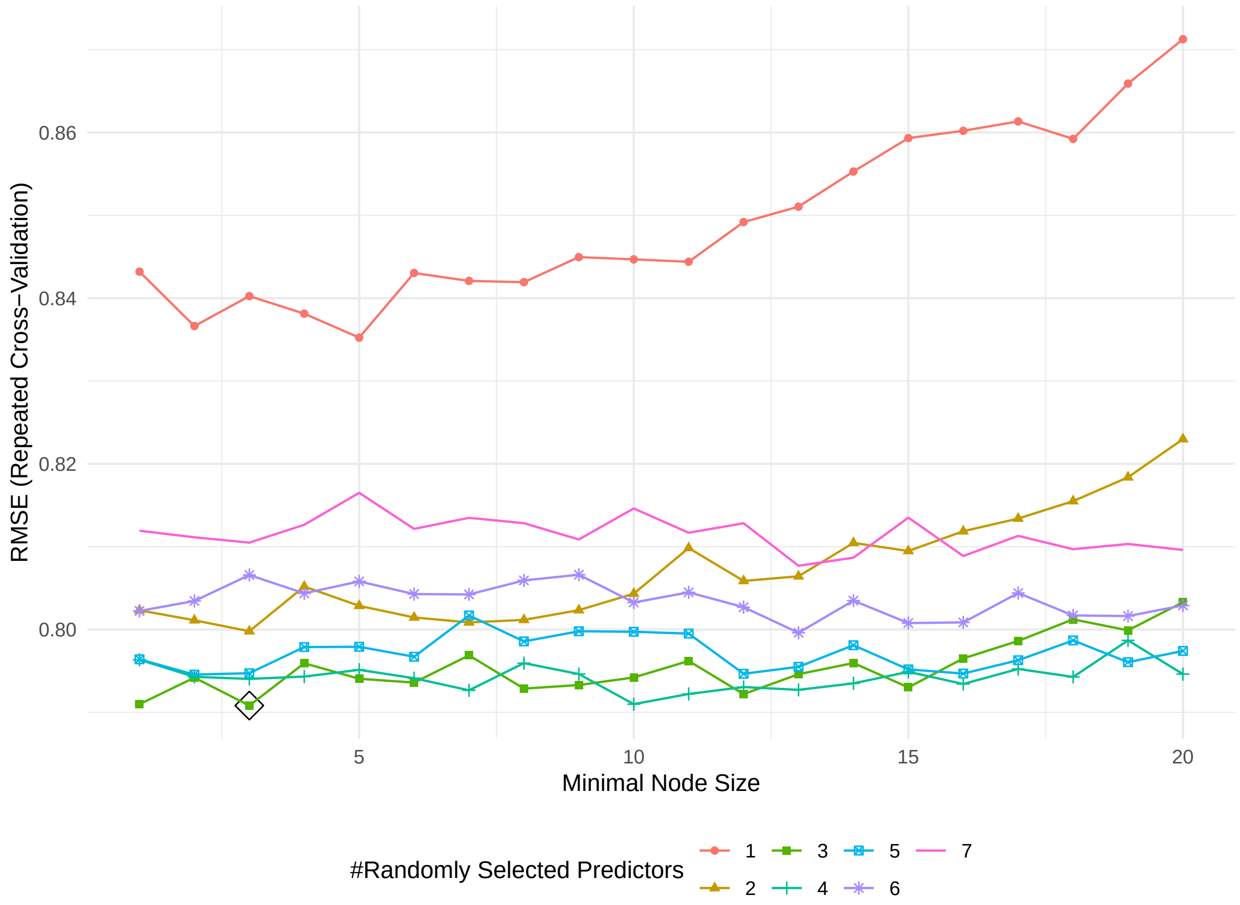
```
parallel::stopCluster(cl)

ggplot(bagging,highlight = T)
```



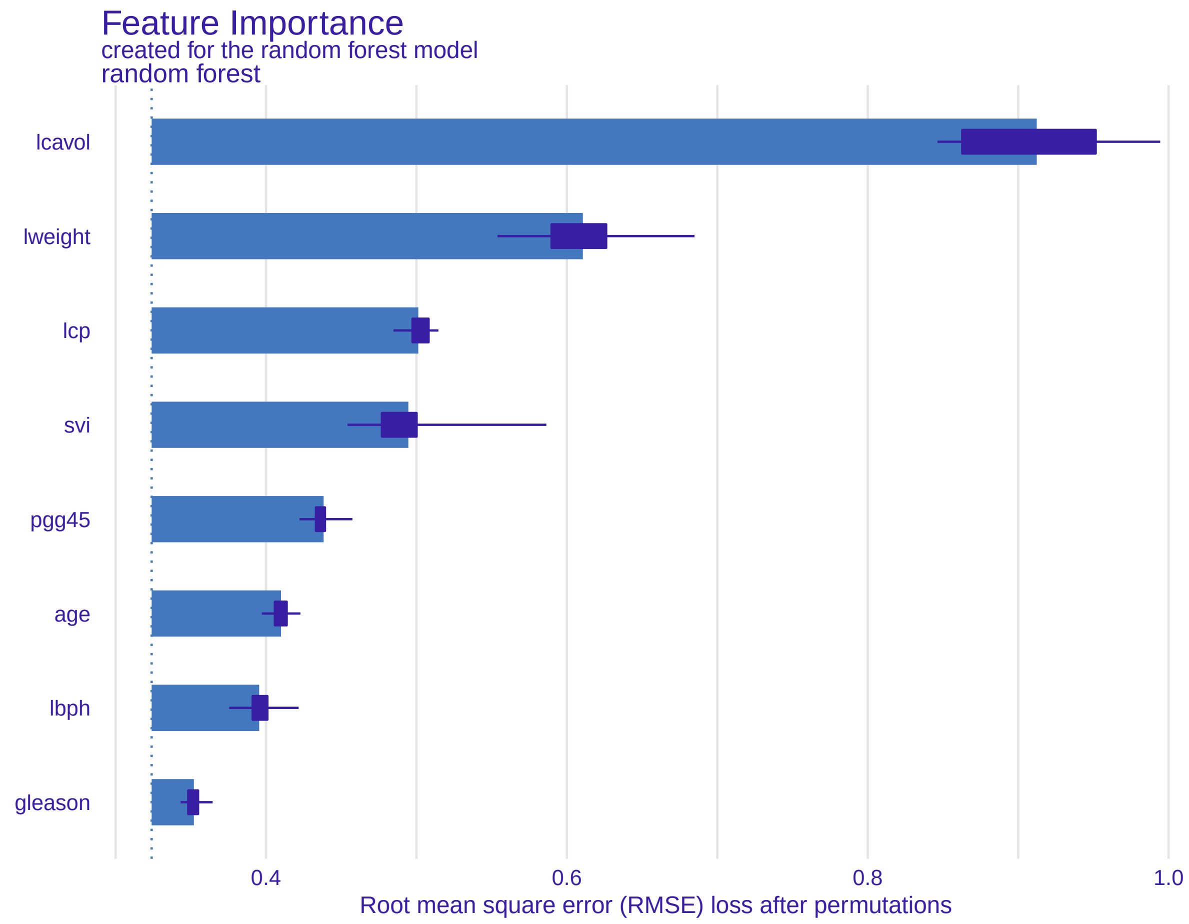
The order of variable importance are lcavol, lweight, age, lbph, svi, lcp, gleason, pgg45.

3

```
cl = parallel::makePSOCKcluster(5)
doParallel::registerDoParallel(cl)

random_forest =
  train(X_tr,
        Y_tr,
        method = "ranger",
        tuneGrid = expand.grid(mtry = 1:7,
                               splitrule = "variance",
                               min.node.size = 1:20),
        metric = "RMSE",
        trControl = ctrl,
        preProcess = c("center","scale"))

ggplot(random_forest,highlight = T)
```



```
rf_explain = DALEX::explain(random_forest,
                             label = "random forest",
                             data=X_tr,
                             y= Y_tr,
                            verbose = F)

rf_imp = DALEX::model_parts(rf_explain)

plot(rf_imp)
```



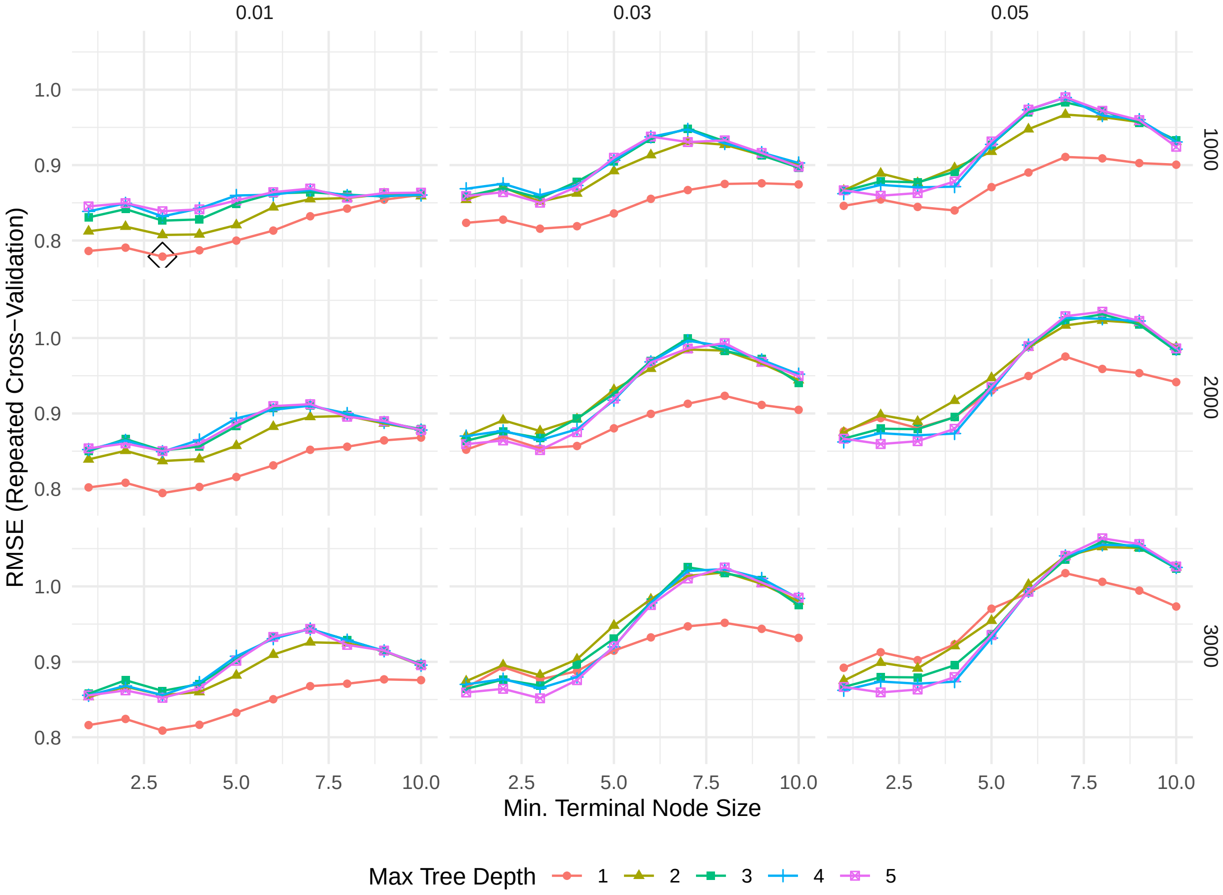
```
parallel::stopCluster(cl)
```

The order of variable importance are lcavol, lweight, age, lbph, svi, lcp, gleason, pgg45.

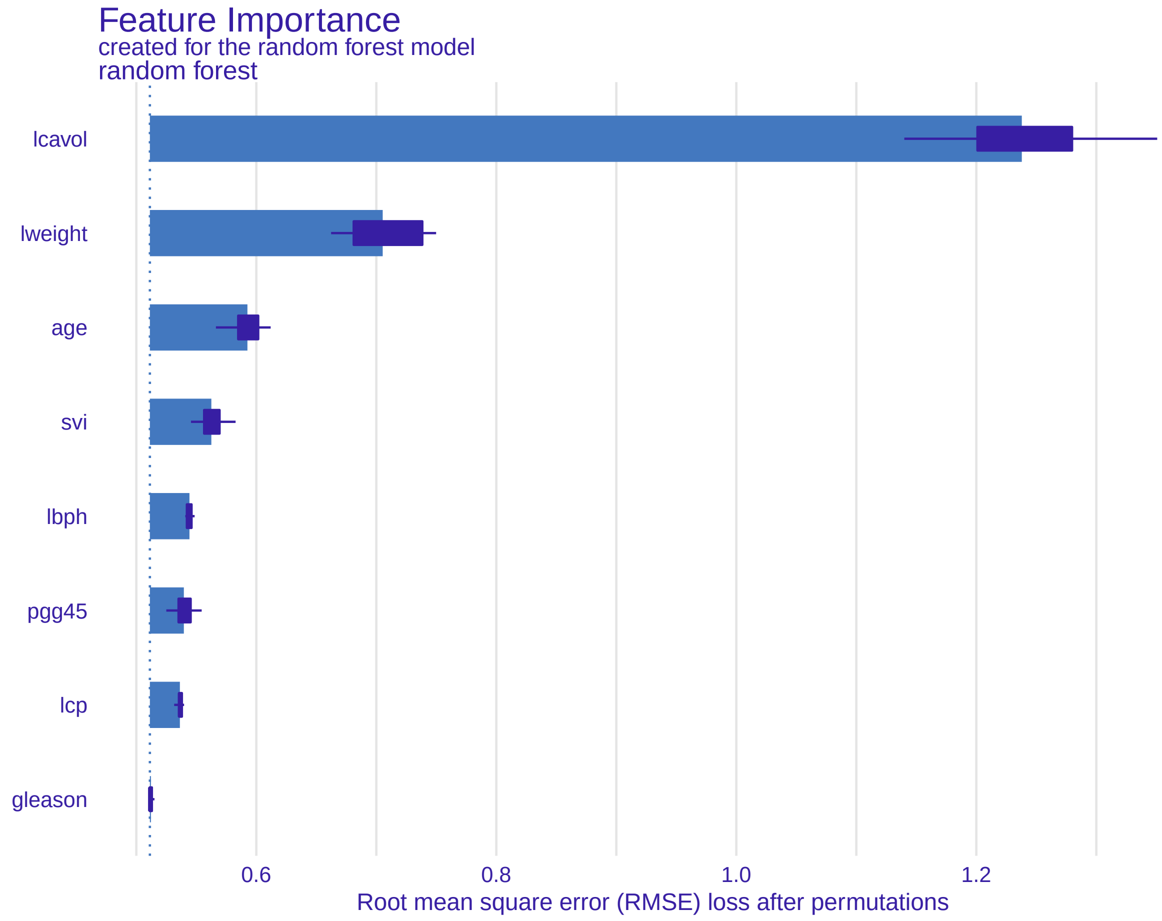
4

```
t1 = Sys.time()
cl = parallel::makePSOCKcluster(5)
doParallel::registerDoParallel(cl)

adab =   train(X_tr,
        Y_tr,
        method = "gbm",
        tuneGrid = expand.grid(n.trees = c(1e+3,2e+3,3e+3),
                               interaction.depth = 1:5,
                               shrinkage = seq(0.01,0.05,len=3),
                               n.minobsinnode = 1:10),
        metric = "RMSE",
        trControl = ctrl,
        preProcess = c("center","scale"),
        verbose = F)
```

```r
ggplot(adab,highlight = T)
```



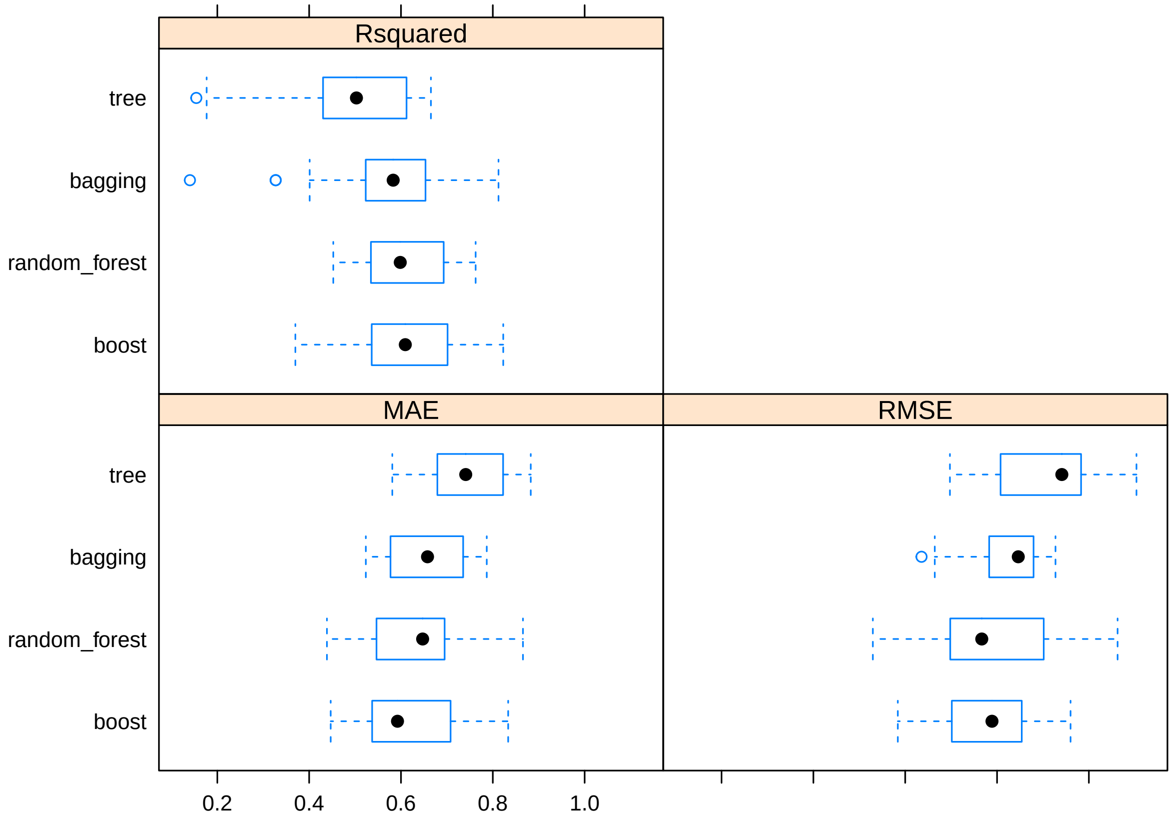
```r
adab_explain = DALEX::explain(adab,
                             label = "random forest",
                             data=X_tr,
                             y= Y_tr,
                            verbose = F)

adab_imp = DALEX::model_parts(adab_explain)

plot(adab_imp)
```



```
parallel::stopCluster(cl)

runt = Sys.time() - t1
```

The order of variable importance are lcavol, lweight, age, lbph, svi, lcp, gleason, pgg45.

```
reg_rsmp =
  resamples(
    list(
      tree = sng_tree,
      bagging = bagging,
      random_forest = random_forest,
      boost  = adab
    )
  )

bwplot(reg_rsmp)
```



All methods perform simlarlly, by choosing the model with minimum loss functions, `Boost` model would be the choice of model.

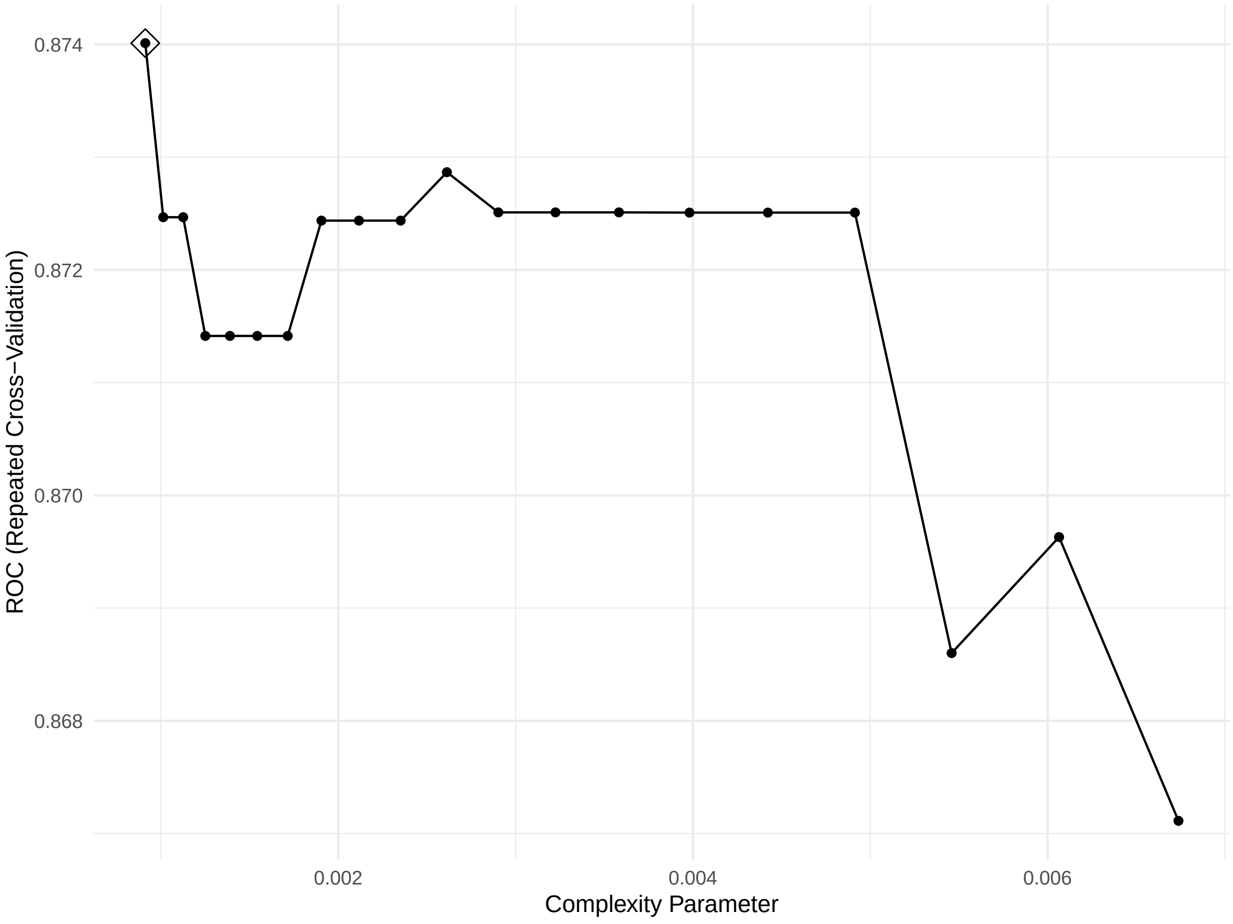
2

1

```
library(ISLR)
data(OJ)

trainindex = createDataPartition(OJ$Purchase,p=0.8,list = F)

X_tr = model.matrix(Purchase~.,OJ[trainindex,])[,-1]

Y_tr = OJ %>% as.matrix %>% .[trainindex,1] %>% as.factor()


X_ts = model.matrix(Purchase~.,OJ[-trainindex,])[,-1]

Y_ts = OJ %>% as.matrix %>% .[-trainindex,"Purchase"] %>% as.factor()

ctrl = trainControl(method = "repeatedcv",number = 5, repeats = 5,
                    summaryFunction = twoClassSummary,
                    classProbs = TRUE)
```

Tree

```
cl = parallel::makePSOCKcluster(5)
doParallel::registerDoParallel(cl)

sng_tree = train(X_tr,
                 Y_tr,
                 method = "rpart",
                 tuneGrid = expand.grid(cp = exp(seq(-5,-7,len = 20))),
                 preProcess = pre,
                 metric = "ROC",
                 trControl = ctrl
                 )

ggplot(sng_tree,highlight = T)
```



```
Y_pr = predict(sng_tree, newdata = X_ts, type = "raw") %>%
  as.factor()


print("the test error is ")
```

```
## [1] "the test error is "
```

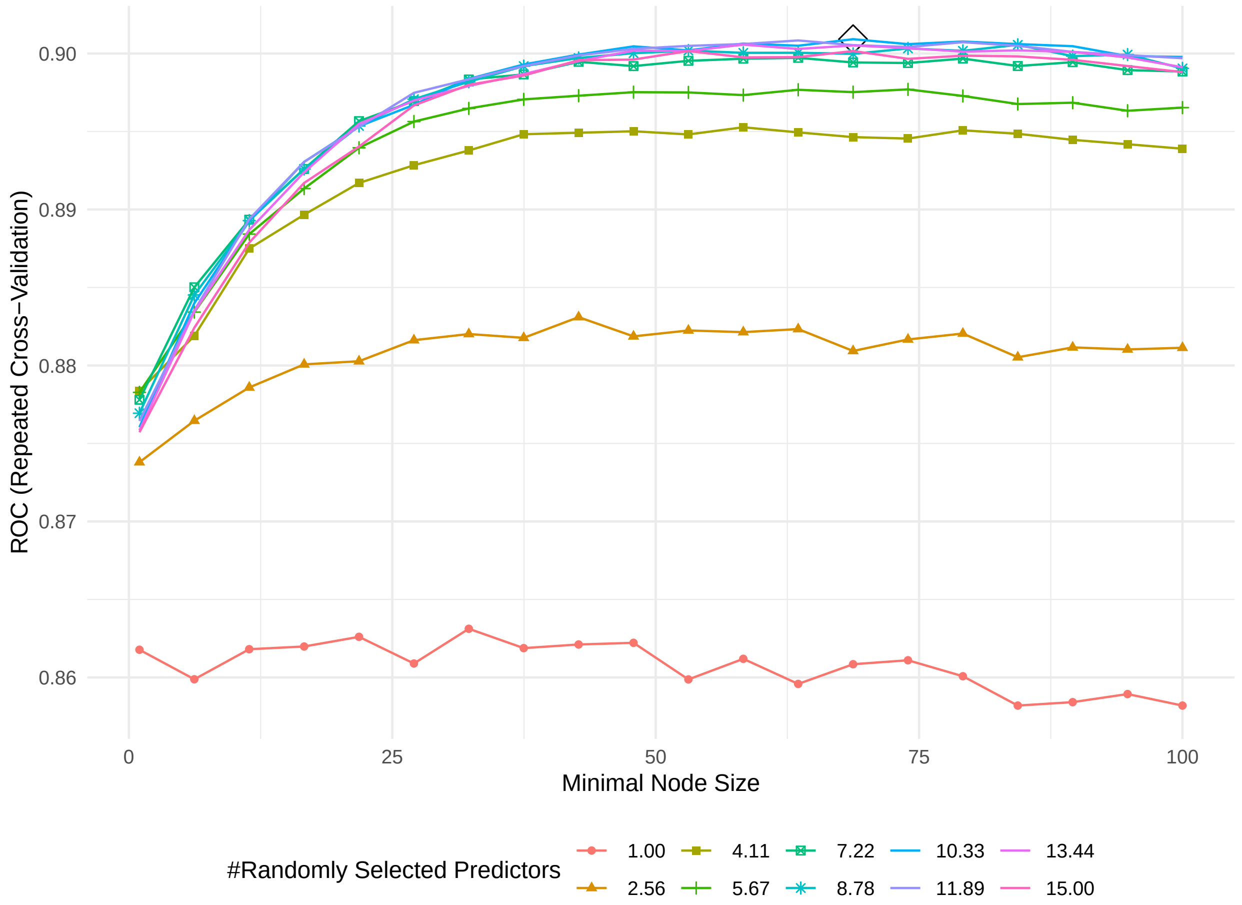
```
sum(Y_ts != Y_pr)/length(Y_pr)
```

```
## [1] 0.23
```

```
parallel::stopCluster(cl)
```

random forest

```
cl = parallel::makePSOCKcluster(5)
doParallel::registerDoParallel(cl)

random_forest =
  train(X_tr,
        Y_tr,
        method = "ranger",
        tuneGrid = expand.grid(mtry = seq(1,15,len=10),
                               splitrule = "gini",
                               min.node.size = seq(1,100,len=20)),
        metric = "ROC",
        trControl = ctrl,
        preProcess = c("center","scale"))

ggplot(random_forest,highlight = T)
```



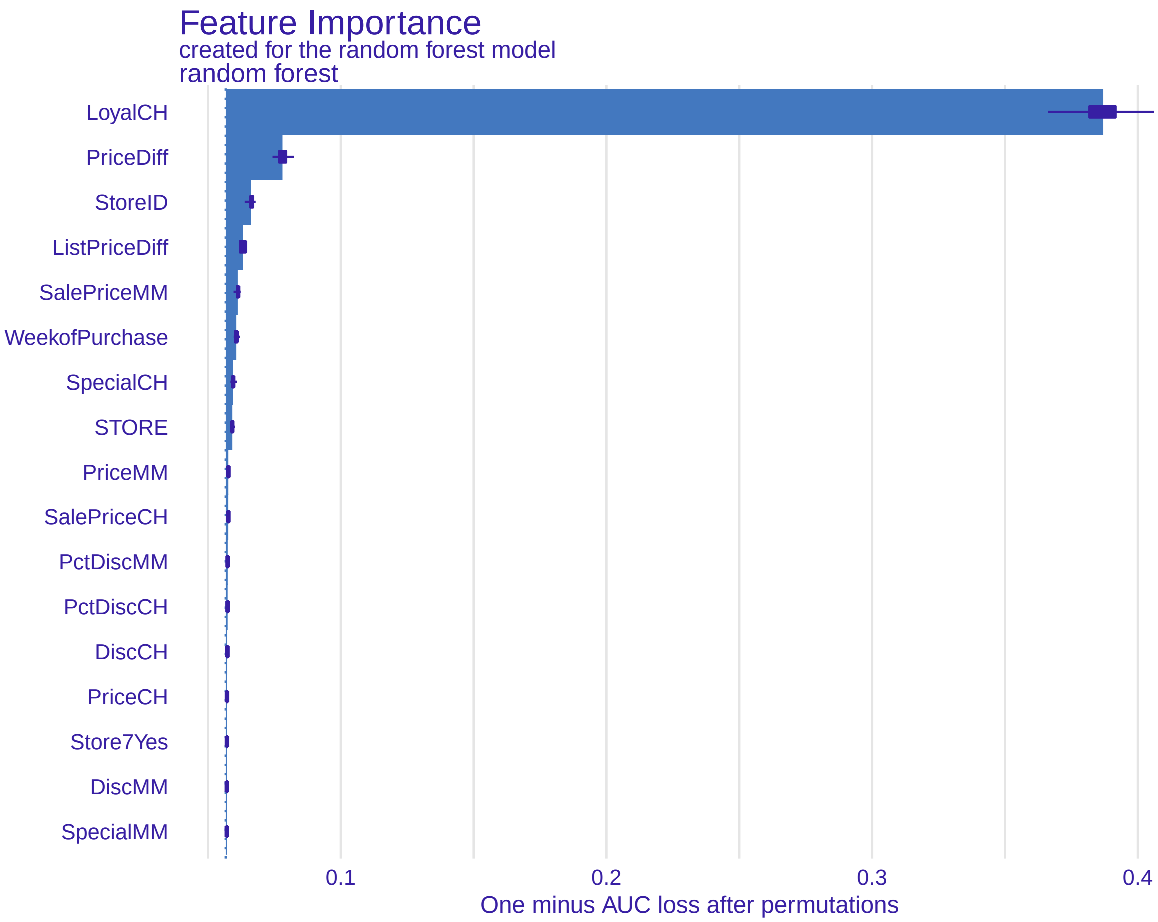
```
Y_pr = predict(random_forest, newdata = X_ts, type = "raw") %>%
  as.factor()


print("the test error is ")
```

```
## [1] "the test error is "
```

```
sum(Y_ts != Y_pr)/length(Y_pr)
```

```
## [1] 0.202
```

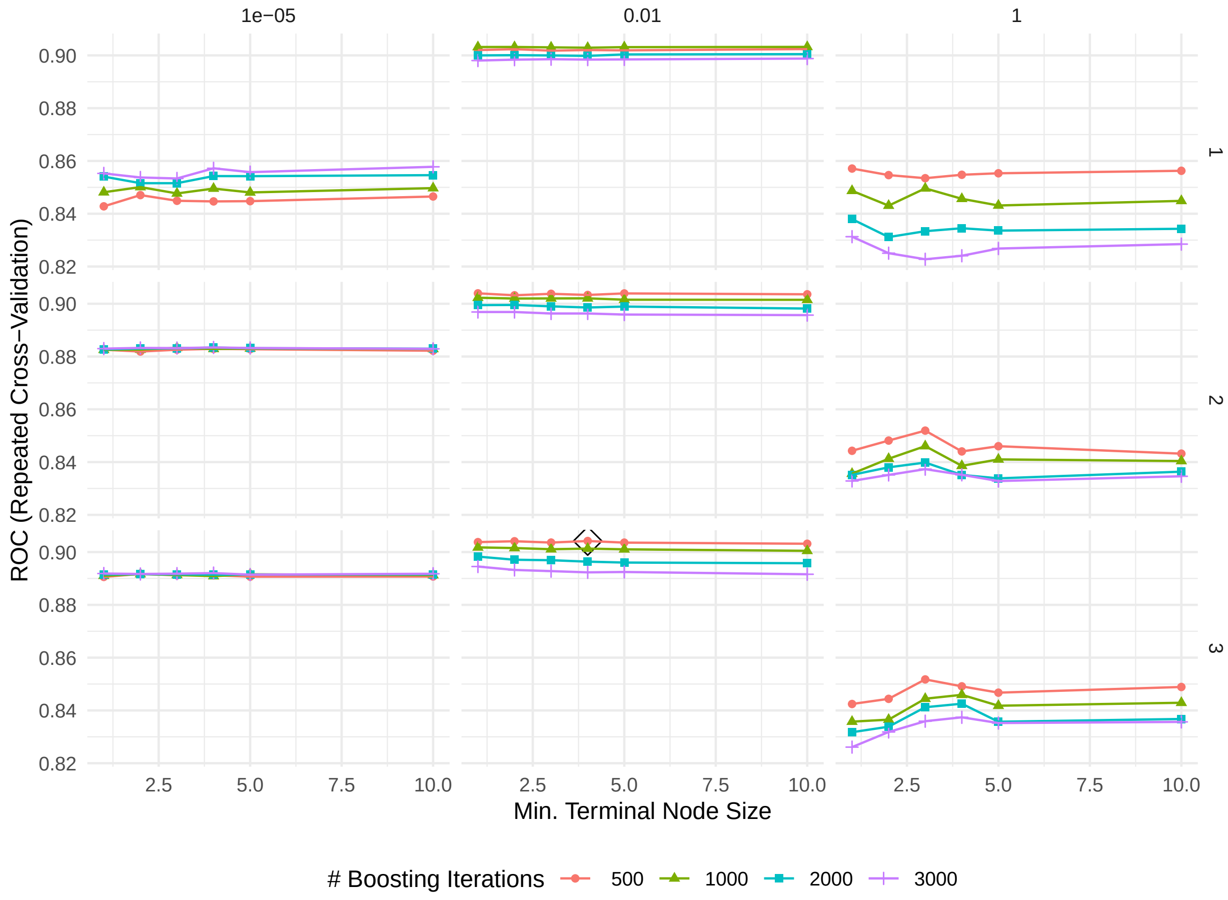
```
rf_explain = DALEX::explain(random_forest,
                             label = "random forest",
                             data=X_tr,
                             y= Y_tr,
                            verbose = F)

rf_imp = DALEX::model_parts(rf_explain)

plot(rf_imp)
```



```
parallel::stopCluster(cl)
```

Boost

```
cl = parallel::makePSOCKcluster(5)
doParallel::registerDoParallel(cl)

ada = train(
  X_tr,
  Y_tr,
  method = "gbm",
  distribution = "adaboost",
  tuneGrid = expand.grid(
    n.trees = c(500, 1e+3, 2e+3, 3e+3),
    interaction.depth = 1:3,
    shrinkage = c(1e-5,1e-2,1),
    n.minobsinnode = c(1:5,10)
  ),
  metric = "ROC",
  verbose = F,
  preProcess = c("center", "scale"),
  trControl = ctrl
)

ggplot(ada,highlight = T)
```



```
Y_pr = predict(ada, newdata = X_ts, type = "raw") %>%
  as.factor()

print("the test error is ")
```

```
## [1] "the test error is "
```

```
sum(Y_ts != Y_pr)/length(Y_pr)
```

```
## [1] 0.178
```

```
parallel::stopCluster(cl)
```