P8106 HW4

Lin Yang

```
library(tidyverse)
library(caret)
library(mlbench)
library(caret)
library(rpart)
library(rpart.plot)
library(party)
library(partykit)
library(pROC)
library(randomForest)
library(ranger)
library(gbm)
library(pdp)
```

Problem 1

```
College <- read.csv("data/College.csv") %>%
  janitor::clean_names() %>%
  select(-1)


set.seed(2022)
trainRows <- createDataPartition(y = College$outstate, p = 0.8, list = FALSE)
College_train <- College[trainRows, ]
College_test <- College[-trainRows, ]

summary(College)
```

```
##       apps           accept          enroll        top10perc
##  Min.   :   81   Min.   :   72   Min.   :  35.0   Min.   : 1.00
##  1st Qu.:  619   1st Qu.:  501   1st Qu.: 206.0   1st Qu.:17.00
##  Median : 1133   Median :  859   Median : 328.0   Median :25.00
##  Mean   : 1978   Mean   : 1306   Mean   : 456.9   Mean   :29.33
##  3rd Qu.: 2186   3rd Qu.: 1580   3rd Qu.: 520.0   3rd Qu.:36.00
##  Max.   :20192   Max.   :13007   Max.   :4615.0   Max.   :96.00
##    top25perc       f_undergrad     p_undergrad       outstate
##  Min.   :  9.00   Min.   :  139   Min.   :    1   Min.   : 2340
##  1st Qu.: 42.00   1st Qu.:  840   1st Qu.:   63   1st Qu.: 9100
##  Median : 55.00   Median : 1274   Median :  207   Median :11200
##  Mean   : 56.96   Mean   : 1872   Mean   :  434   Mean   :11802
##  3rd Qu.: 70.00   3rd Qu.: 2018   3rd Qu.:  541   3rd Qu.:13970
```

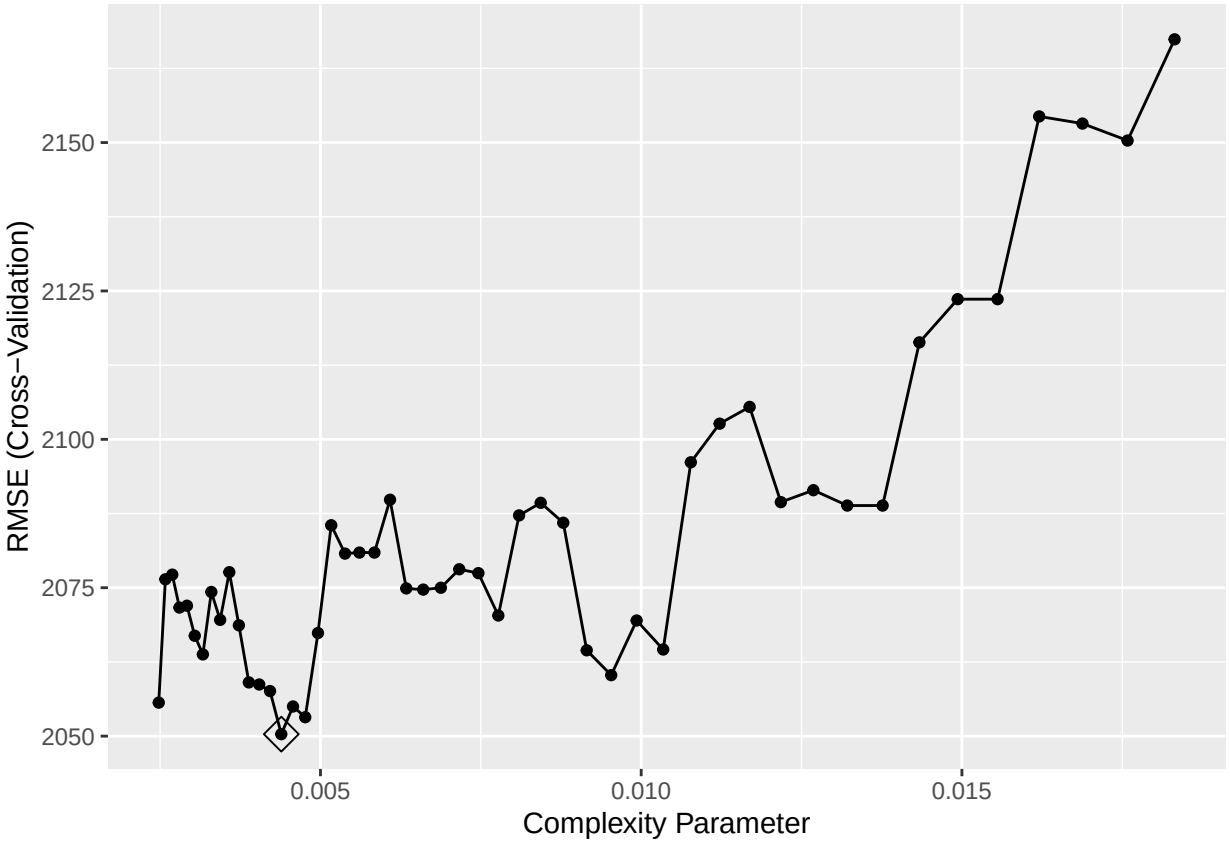
```
##  Max.   :100.00   Max.   :27378   Max.   :10221   Max.   :21700
##    room_board       books           personal         ph_d
##  Min.   :2370   Min.   : 250.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3736   1st Qu.: 450.0   1st Qu.: 800   1st Qu.: 60.00
##  Median :4400   Median : 500.0   Median :1100   Median : 73.00
##  Mean   :4586   Mean   : 547.5   Mean   :1214   Mean   : 71.09
##  3rd Qu.:5400   3rd Qu.: 600.0   3rd Qu.:1500   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :100.00
##     terminal        s_f_ratio      perc_alumni        expend        grad_rate
##  Min.   : 24.00   Min.   : 2.50   Min.   : 2.00   Min.   : 3186   Min.   : 15
##  1st Qu.: 68.00   1st Qu.:11.10   1st Qu.:16.00   1st Qu.: 7477   1st Qu.: 58
##  Median : 81.00   Median :12.70   Median :25.00   Median : 8954   Median : 69
##  Mean   : 78.53   Mean   :12.95   Mean   :25.89   Mean   :10486   Mean   : 69
##  3rd Qu.: 92.00   3rd Qu.:14.50   3rd Qu.:34.00   3rd Qu.:11625   3rd Qu.: 81
##  Max.   :100.00   Max.   :39.80   Max.   :64.00   Max.   :56233   Max.   :118
```

a. Regression Tree

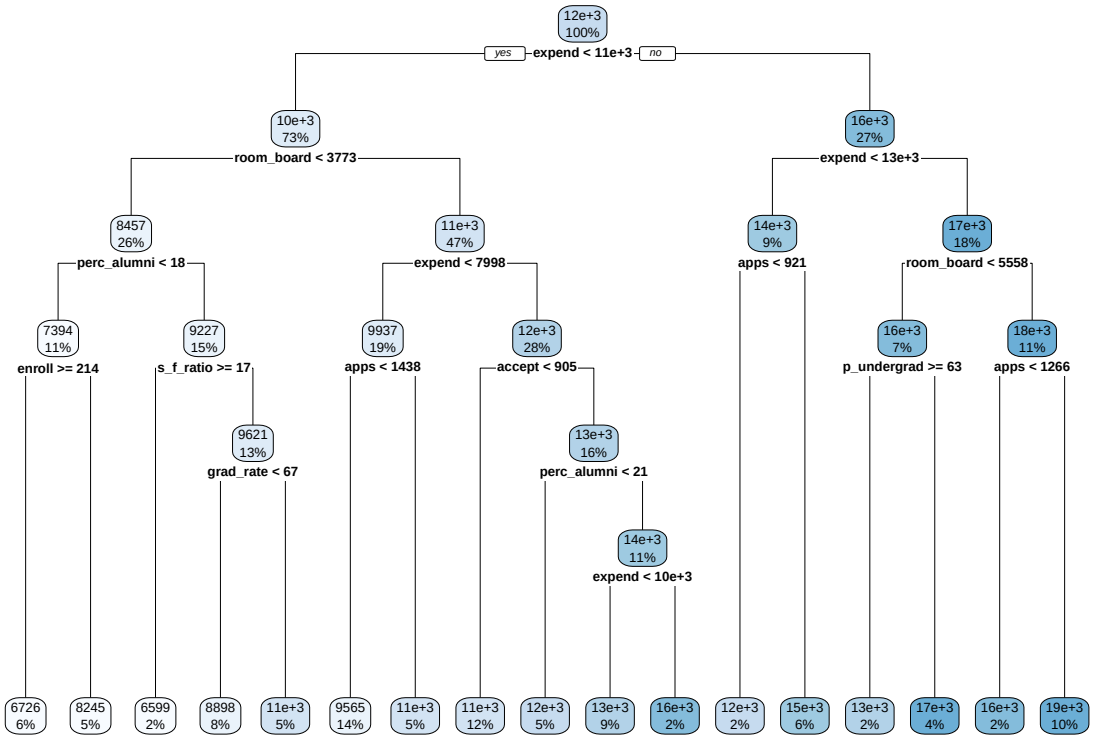
```
ctrl <- trainControl(method = "cv")
set.seed(2022)
r.tree <- train(outstate ~ . ,
                College_train,
                method = "rpart",
                tuneGrid = data.frame(cp = exp(seq(-6,-4, length = 50))),
                trControl = ctrl)
r.tree$bestTune
```

```
##             cp
## 15 0.004389362
```

```
ggplot(r.tree, highlight = TRUE)
```




```
rpart.plot(r.tree$finalModel)
```



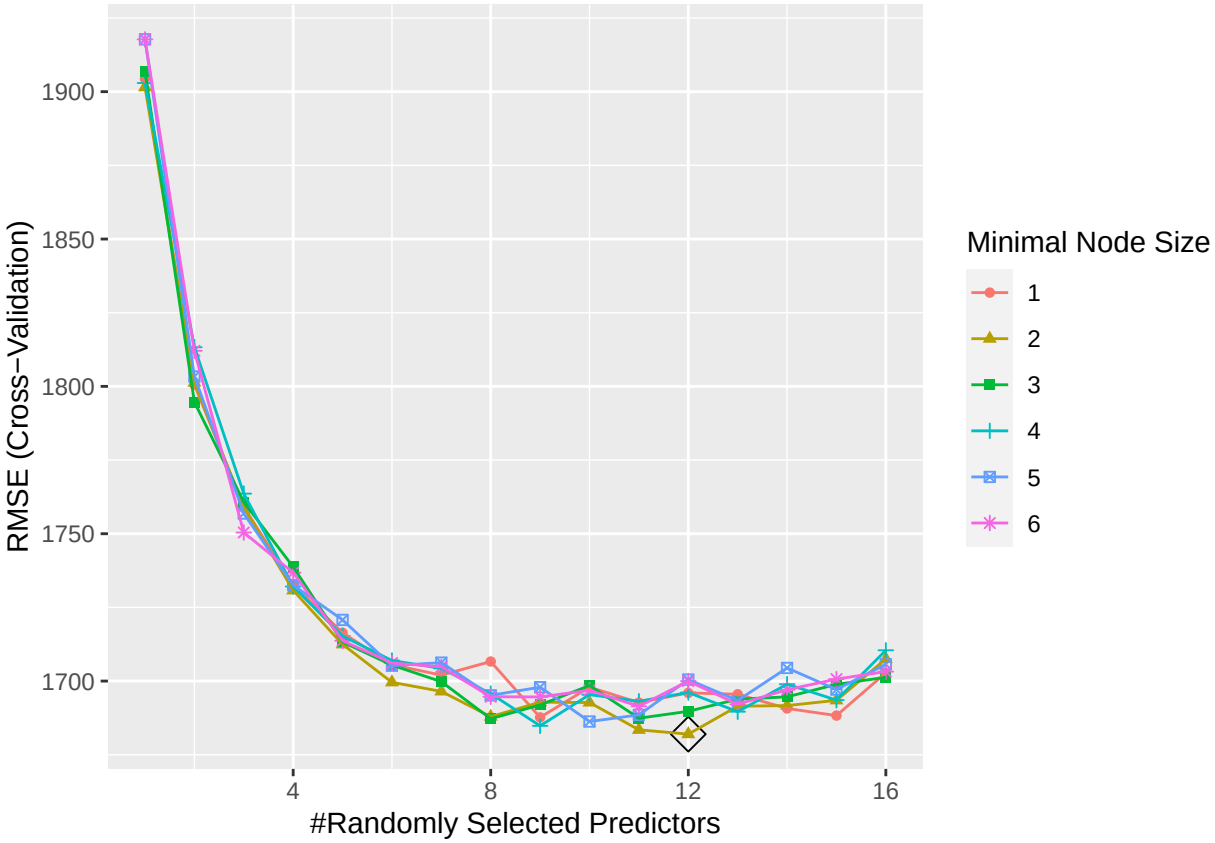
The best cp is selected to be 0.00438936184277844. The root node is **expend** less than 11000 or not. There are 17 terminal nodes, thus this is a fairly large tree.

b. Random Forest

```
rf.grid <- expand.grid(mtry = 1:16,
                       splitrule = "variance",
                       min.node.size = 1:6)
set.seed(2022)
rf.fit <- train(outstate ~ . ,
                College_train,
                method = "ranger",
                tuneGrid = rf.grid,
                trControl = ctrl)
rf.fit$bestTune
```

```
##    mtry splitrule min.node.size
## 68   12  variance             2
```

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



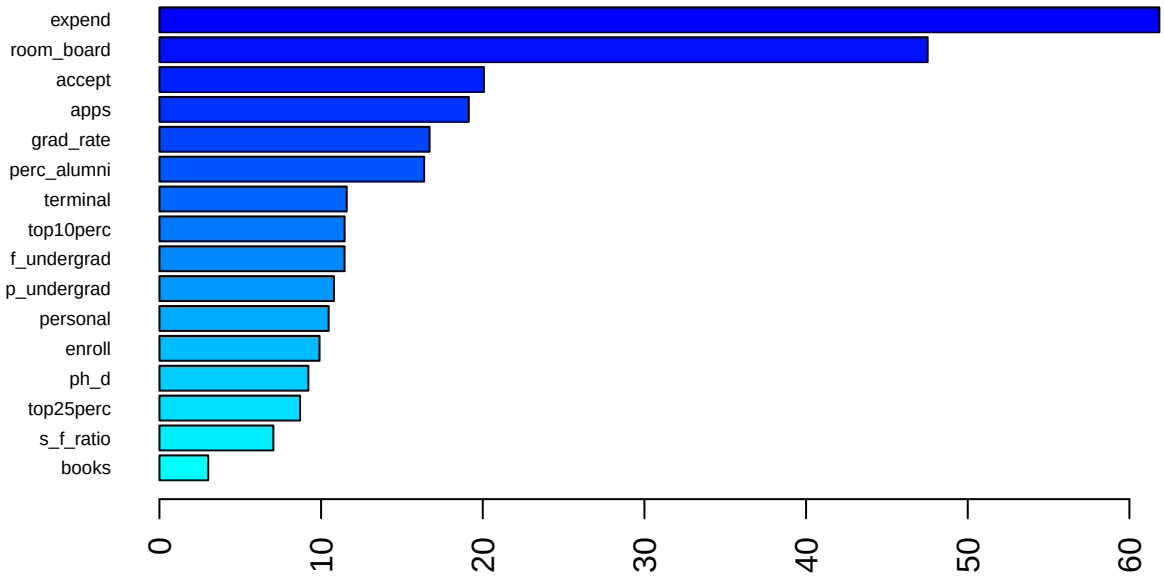

```
pred.rf <- predict(rf.fit, newdata = College_test)
test_error <- RMSE(pred.rf, College_test$outstate)
test_error
```

```
## [1] 1980.006
```

The best tuning parameters are found to be m = 12 and minimum node size = 2. The test error is 1980.0060684.

```
set.seed(2022)
rf.per <- ranger(outstate ~ . ,
                 College_train,
                 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.per), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.6,
        col = colorRampPalette(colors = c("cyan","blue"))(16))
```



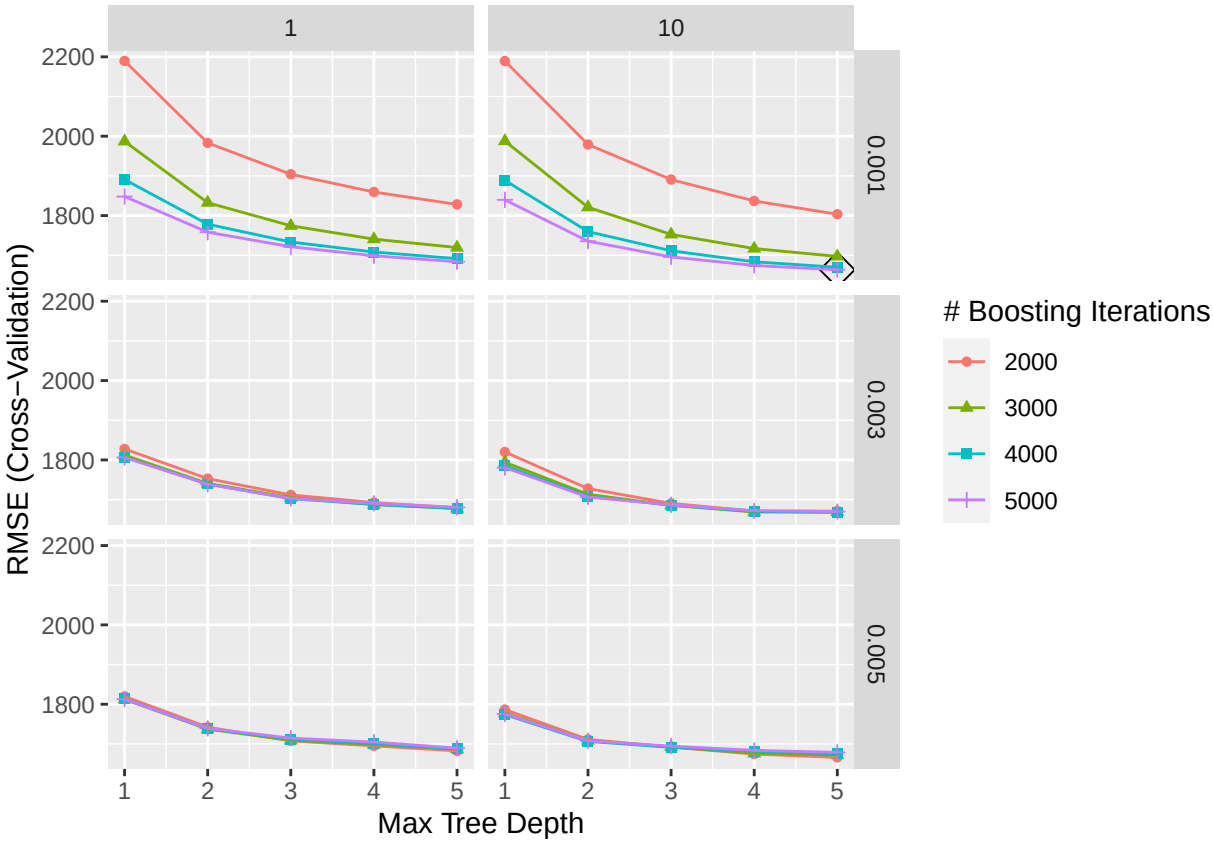
The variable importance plot is based on permutation importance. The most important variables are found to be `expend` and `room_board`. `accept`, `apps`, `grad_rate`, and `perc_alumni` are relatively important.

c. Boosting

```
gbm.grid <- expand.grid(n.trees = c(2000,3000,4000,5000),
                        interaction.depth = 1:5,
                        shrinkage = c(0.001,0.003,0.005),
                        n.minobsinnode = c(1,10))

set.seed(2022)
gbm.fit <- train(outstate ~ . ,
                 College_train,
                 method = "gbm",
                 tuneGrid = gbm.grid,
                 trControl = ctrl,
                 verbose = FALSE)

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


1
10
0.001
0.003
0.005
2200
2000
1800
RMSE (Cross−Validation)
1
2
3
4
5
Max Tree Depth
Boosting Iterations
2000
3000
4000
5000