p8106_HW4_qz3366

Qing Zhou

2023-04-20

Question 1

Data preparation

In this exercise, we will build tree-based models using the College data. The dataset contains statistics for 565 US Colleges from a previous issue of US News and World Report. The response variable is the out-of-state tuition (Outstate).

```
# Data Import
college_df =
  read.csv("data/College.csv") %>%
  na.omit() %>%
  janitor::clean_names() %>%
  relocate("outstate", .after = "grad_rate") %>%
  select(-college)
```

Partition the dataset into two parts: training data (80%) and test data (20%).

```
set.seed(1)
trainRows <- createDataPartition(y = college_df$outstate, p = 0.8, list = FALSE)

# Training data
college_train = college_df[trainRows, ]

# Testing data
college_test = college_df[-trainRows, ]

# create a cross-validation object
ctrl1 <- trainControl(method = "cv")
```

(a) Regression tree

Build a regression tree on the training data to predict the response. Create a plot of the tree.

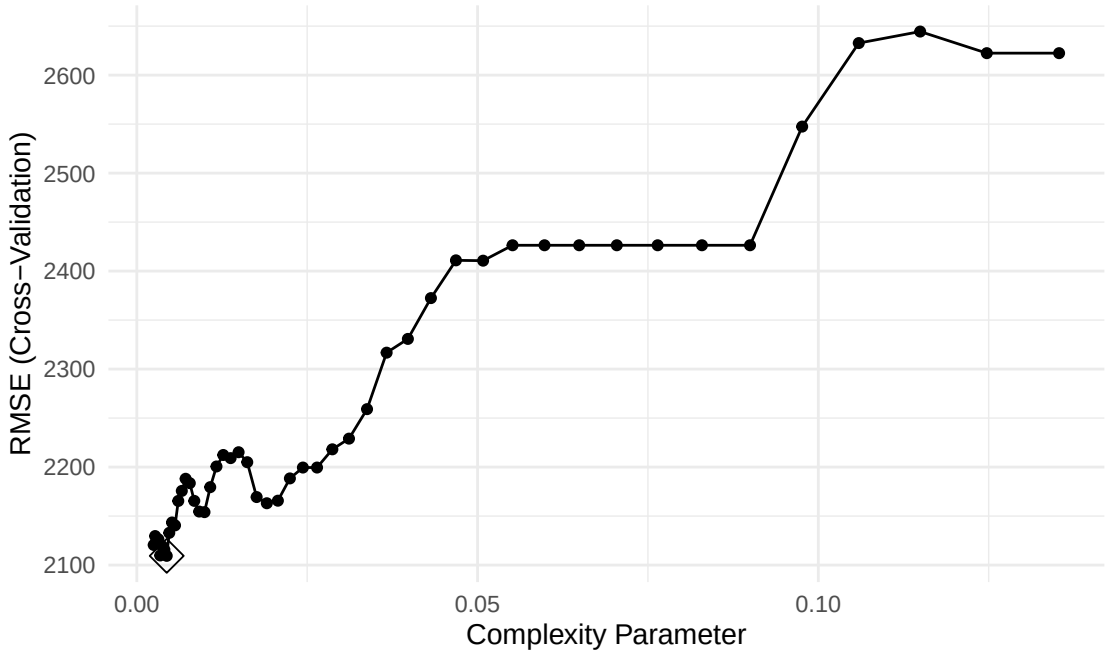
```
set.seed(1)

# build a regression tree on the training data
rpart.fit <- train(outstate ~ . ,
                   data = college_train,
```

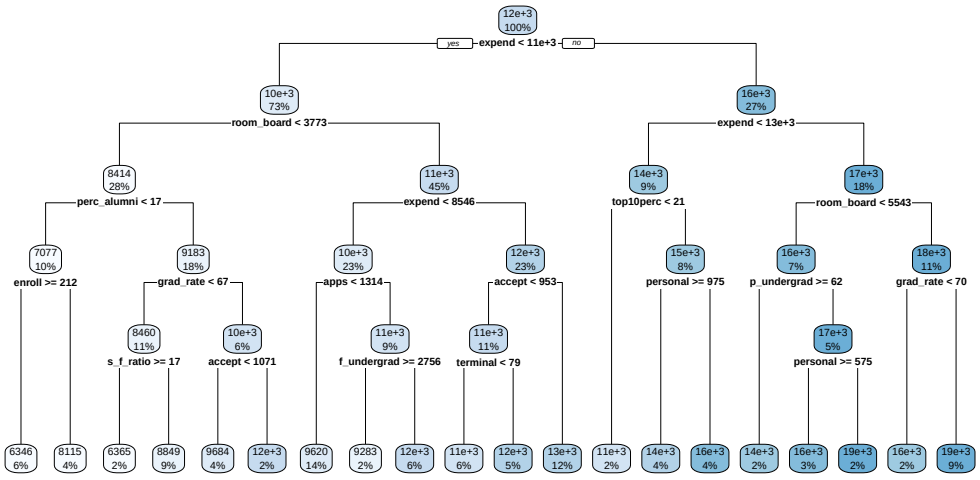
```
                    method = "rpart",
                    tuneGrid = data.frame(cp = exp(seq(-6,-2, length = 50))),
                    trControl = ctrl1)
rpart.fit$bestTune
```

```
##            cp
## 8 0.004389362
```

```
# plot of the complexity parameter
ggplot(rpart.fit, highlight = TRUE)
```




```
# create a plot of the tree
rpart.plot(rpart.fit$finalModel)
```



- The root node is `expend` over or under 11K.
- The optimal cp is 0.004389362.
- The pruned tree based on the optimal cp value is plotted as above. It's quite complicated with 20 terminal nodes and 19 splits.

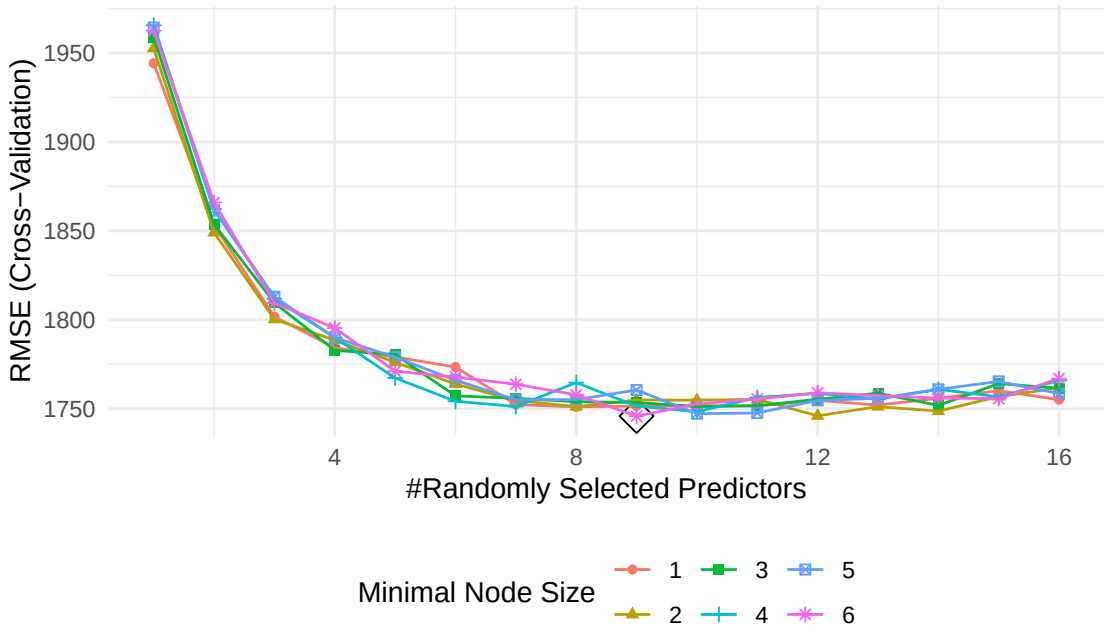
(b) Random forest

Perform random forest on the training data. Report the variable importance and the test error.

```
rf.grid <- expand.grid(mtry = 1:16,
                       splitrule = "variance",
                       min.node.size = 1:6)

set.seed(1)
# train a random forest model on the training data
rf.fit <- train(outstate ~ .,
                data = college_train,
                 method = "ranger",
                 tuneGrid = rf.grid,
                 trControl = ctrl1)

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

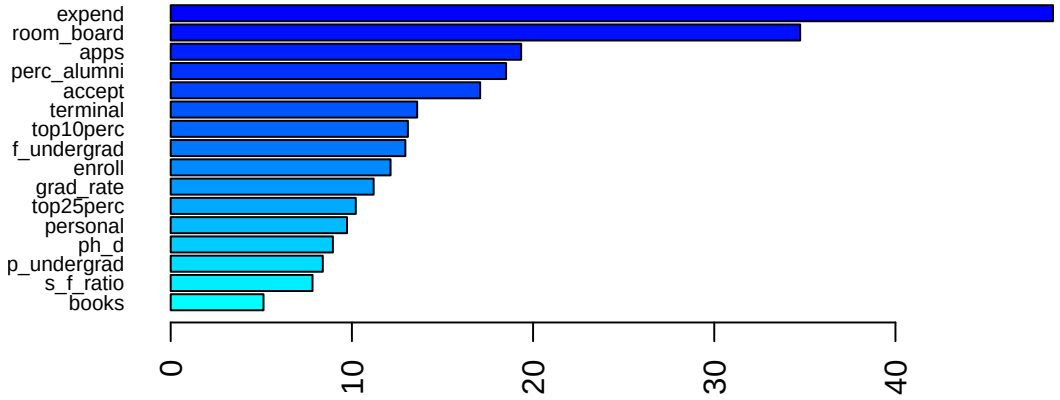



```
rf.fit$bestTune
```

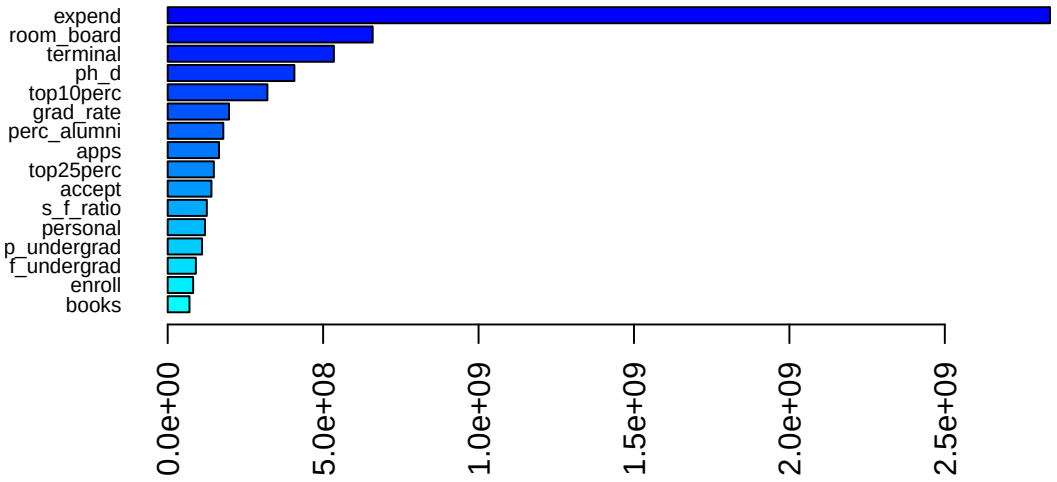
```
##    mtry splitrule min.node.size
## 54    9  variance             6
```

- Using ranger method, we perform Random Forest algorithm with minimum node size 6 and 9 selected predictors.
- Once a random forest model is trained, it is common to inquire about the variables that have the most predictive ability. Variables with a high degree of importance are instrumental in determining the outcome and their values can significantly affect the outcome. On the other hand, variables with low importance may be excluded from the model, which can simplify the model and improve its efficiency in terms of fitting and prediction.

```
# variable importance using permutation methods
set.seed(1)
rf.perm = ranger(outstate ~ .,
                      data = college_train,
                      mtry = rf.fit$bestTune[[1]],
                      splitrule = "variance",
                      min.node.size = rf.fit$bestTune[[3]],
                      importance = "permutation",
                      scale.permutation.importance = TRUE)
# Report variable importance
barplot(sort(ranger::importance(rf.perm), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan", "blue"))(16))
```




```
# variable importance using impurity methods
set.seed(1)
rf.impu <- ranger(outstate ~ .,
                          data = college_train,
                          mtry = rf.fit$bestTune[[1]],
                          splitrule = "variance",
                          min.node.size = rf.fit$bestTune[[3]],
                          importance = "impurity")
# Report variable importance
barplot(sort(ranger::importance(rf.impu), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan", "blue"))(16))
```



- The model indicated that the variables expend and room_board had the highest predictive power and their values were the most significant in determining the out-of-state tuition cost (Outstate). This suggests that these variables play a crucial role in influencing the Outstate cost.

```
# Test error
pred.rf <- predict(rf.fit, newdata = college_test)
RMSE(pred.rf, college_test$outstate)
```

```
## [1] 1651.307
```

- The test error of the model is 1651.307