Ensemble Methods (Tidymodels)

Yifei Sun, Runze Cui

Contents

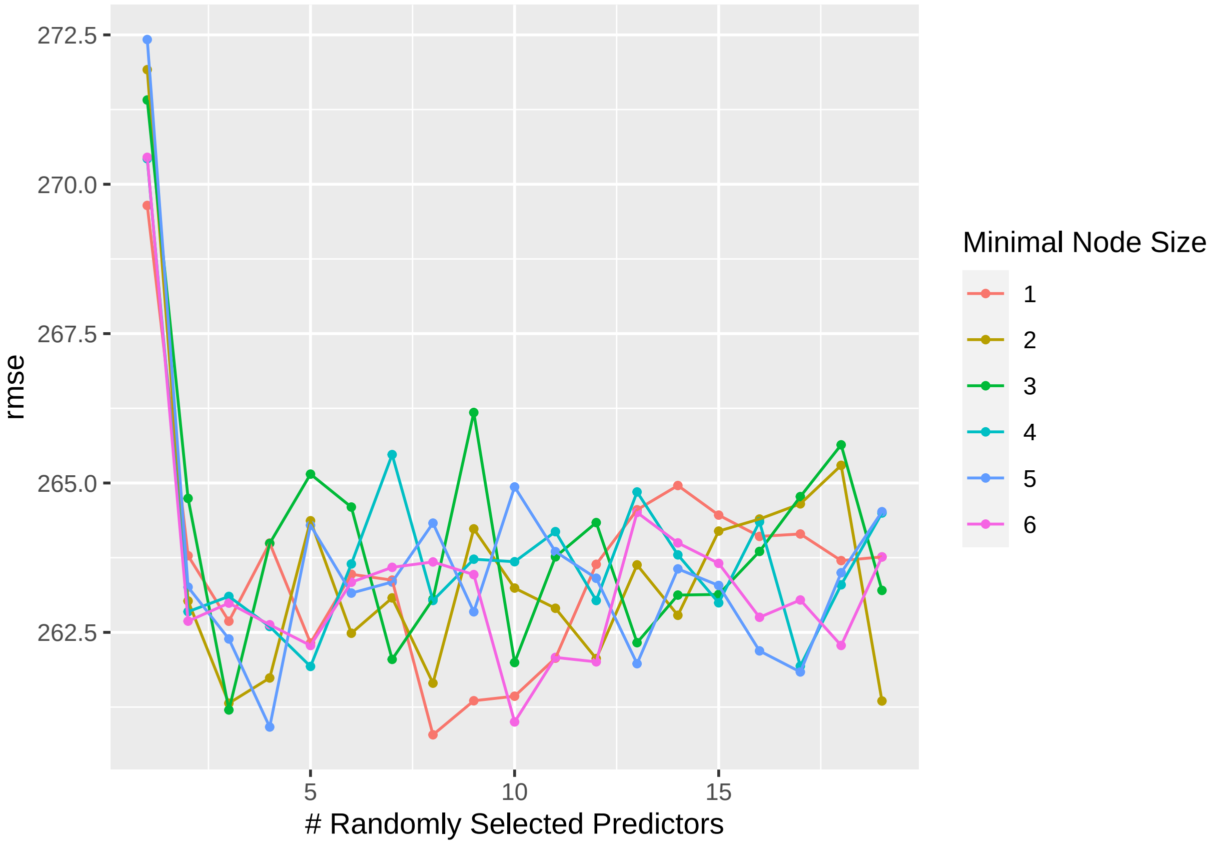
```
library(tidyverse)
library(ISLR)
library(caret)
library(tidymodels)
library(bonsai)
library(lightgbm)
library(ranger)

tidymodels_prefer()
```

Predict a baseball player’s salary on the basis of various statistics associated with performance in the previous year.

```
data(Hitters)
Hitters <- na.omit(Hitters)

set.seed(1)
data_split <- initial_split(Hitters, prop = 0.8)

# Extract the training and test data
training_data <- training(data_split)
testing_data <- testing(data_split)
```

Random forest

```
set.seed(1)
cv_folds <- vfold_cv(training_data)

# Model specification
rf_spec <- rand_forest(mtry = tune(), min_n = tune()) %>%
  set_engine("ranger", splitrule = "variance") %>%
  set_mode("regression")

# Tuning parameters
rf_grid_set <- parameters(mtry(range = c(1, 19)), min_n(range = c(1, 6)))
rf_grid <- grid_regular(rf_grid_set, levels = c(19, 6))

# Set up the workflow
rf_workflow <- workflow() %>%
  add_model(rf_spec) %>%
  add_formula(Salary ~ .)

rf_tune <- rf_workflow %>%
  tune_grid(resamples = cv_folds,
            grid = rf_grid)

autoplot(rf_tune, metric = "rmse")
```

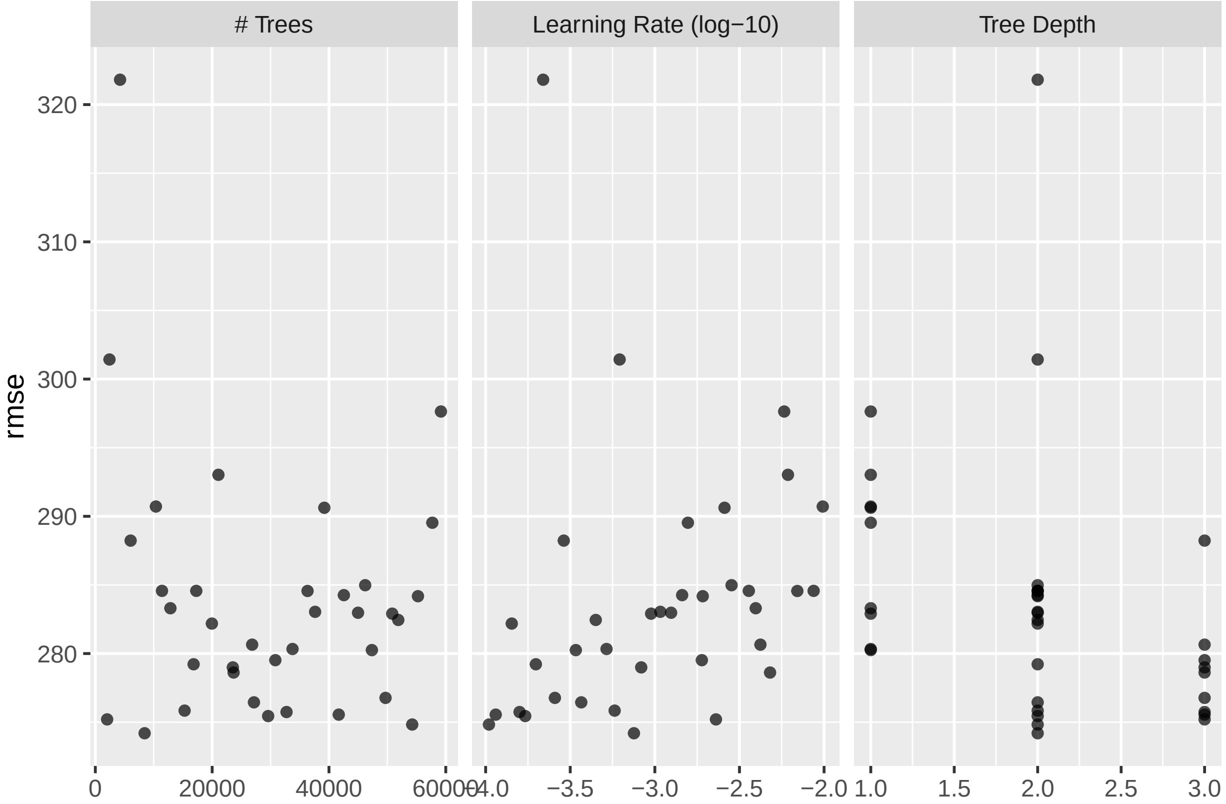


```r
rf_best <- select_best(rf_tune, metric = "rmse")

# Update the model spec
final_rf_spec <- rf_spec %>%
  update(mtry = rf_best$mtry, min_n = rf_best$min_n)

rf_fit <- fit(final_rf_spec, formula = Salary ~ ., data = training_data)

pred.rf <- predict(rf_fit, new_data = testing_data)

RMSE(pred.rf$.pred, testing_data$Salary)
```

```
## [1] 313.9583
```

Boosting

```r
# Model specification
gbm_spec <- boost_tree(trees = tune(), tree_depth = tune(),
                       learn_rate = tune(), min_n = tune()) %>%
  set_engine("lightgbm") %>%
  set_mode("regression")

# Tuning parameters Setup
gbm_grid_set <- parameters(trees(range = c(500, 60000)),
                           tree_depth(range = c(1, 3)),
                           learn_rate(range = c(-4, -2)),
                               min_n(range = c(1, 1)))

gbm_grid <- grid_regular(gbm_grid_set, levels = c(6, 3, 2, 1))
gbm_grid2 <- grid_latin_hypercube(gbm_grid_set, size = 36)

# Set up the workflow
gbm_workflow <- workflow() %>%
  add_model(gbm_spec) %>%
  add_formula(Salary ~ .)

gbm_tune <- gbm_workflow %>%
  tune_grid(resamples = cv_folds,
            grid = gbm_grid2)

autoplot(gbm_tune, metric = "rmse")
```



```
gbm_best <- select_best(gbm_tune, metric = "rmse")

# Update the model spec
final_gbm_spec <- gbm_spec %>%
  update(trees = gbm_best$trees, tree_depth = gbm_best$tree_depth,
         learn_rate = gbm_best$learn_rate, min_n = gbm_best$min_n)

gbm_fit <- parsnip::fit(final_gbm_spec, formula = Salary ~ ., data = training_data)


pred.gbm <- predict(gbm_fit, new_data = testing_data)
```

```r
RMSE(pred.gbm$.pred, testing_data$Salary)
```

```
## [1] 309.2751
```