Random Forest Capstone

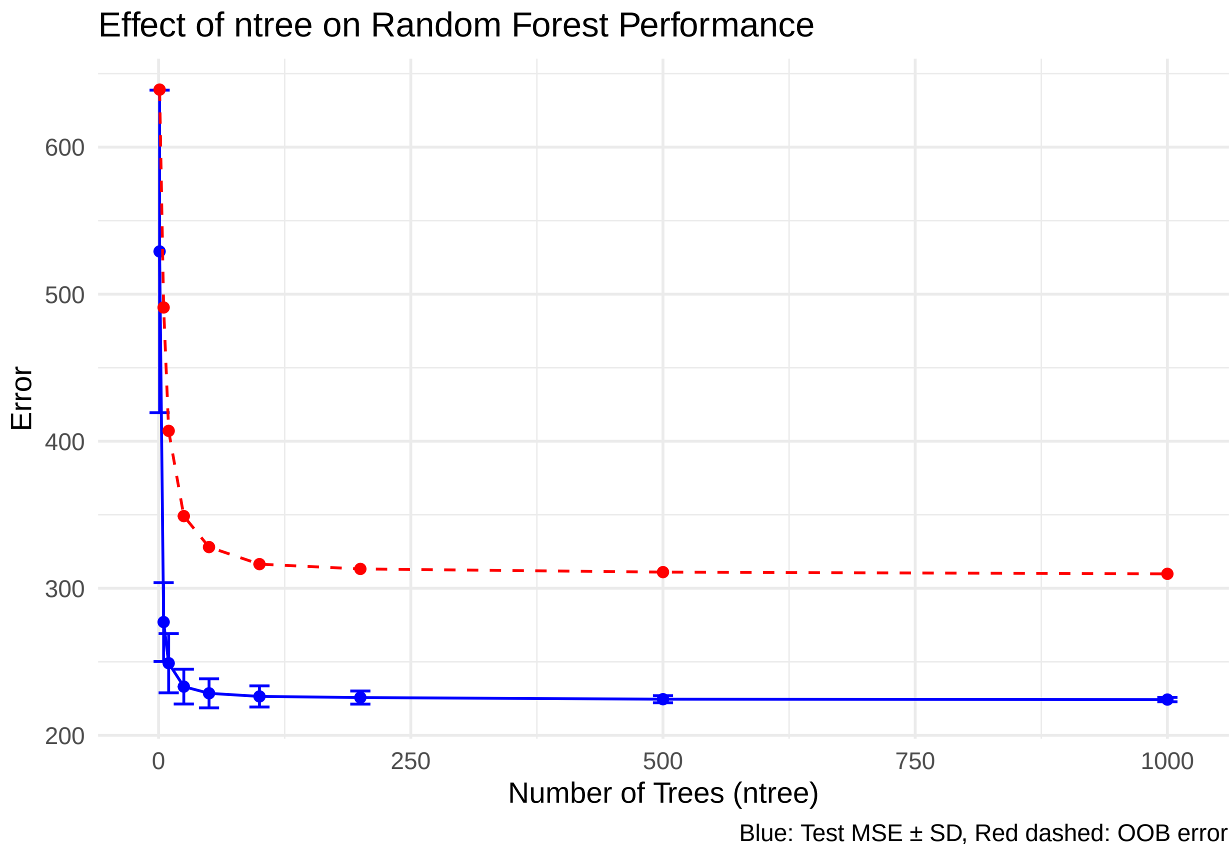
2025-10-23

Running the Code

```
library(MASS)
library(randomForest)
library(tidyverse)

data1 <- read.csv('onlyCompleteData.csv')
data1 <- data1 %>% select(WRC., k_percent, isolated_power, avg_swing_speed, squared_up_contact, avg_swin
                          ideal_angle_rate, exit_velocity_avg, launch_angle_avg, sweet_spot_percent, ba
                          hard_hit_percent, z_swing_percent, oz_swing_percent, meatball_swing_percent, 
                          pull_percent, straightaway_percent, opposite_percent, groundballs_percent, fl
                          linedrives_percent, popups_percent,year)

set.seed(123)
train_idx <- which(data1$year == 2024)
train <- data1[train_idx, ]
train <- train %>% select(-year)
test <- data1[-train_idx, ]
test <- test %>% select(-year)

mtry_val <- 6
ntree_values <- c(1,5,10, 25, 50, 100, 200, 500,1000)
n_reps <- 20

results <- data.frame(ntree=integer(),seed=integer(),test_MSE=double(),OOB_error=double())

for (nt in ntree_values) { # goes through all different ntree values
  for (s in 1:n_reps) { # do many times for each ntree
    set.seed(s)
    rf_model <- randomForest(WRC. ~ ., data=train, mtry=mtry_val, ntree=nt)
    yhat.bag <- predict(rf_model, newdata=test)
    test_mse <- mean((yhat.bag - test$WRC.)^2)
    results <- rbind(results, data.frame(ntree=nt,test_MSE=test_mse,OOB_error=rf_model$mse[nt]))
  }
}

results_all <- results %>% group_by(ntree) %>%summarize(mean_test_MSE = mean(test_MSE),sd_test_MSE = sd
            mean_OOB_error = mean(OOB_error),sd_OOB_error = sd(OOB_error))
```

```
ggplot(results_all, aes(x=ntree)) + geom_errorbar(aes(ymin=mean_test_MSE-sd_test_MSE, ymax=mean_test_MS
                width=20, color="blue") +geom_line(aes(y=mean_test_MSE), color="blue") + geom_point(aes
  geom_line(aes(y=mean_OOB_error), color="red",linetype="dashed") +
  geom_point(aes(y=mean_OOB_error), color="red") +
```

```r
  labs(title="Effect of ntree on Random Forest Performance",x="Number of Trees (ntree)",y="Error",captio
  theme_minimal()
```




```r
results_all
```

```
## # A tibble: 9 x 5
##   ntree mean_test_MSE sd_test_MSE mean_OOB_error sd_OOB_error
##   <dbl>         <dbl>       <dbl>          <dbl>        <dbl>
## 1     1          529.      110.             639.       213.
## 2     5          277.       26.8            491.        67.9
## 3    10          249.       20.1            407.        33.6
## 4    25          233.       11.8            349.        25.1
## 5    50          229.        9.88           328.        19.2
## 6   100          226.        7.19           316.        11.6
## 7   200          226.        4.46           313.         8.89
## 8   500          225.        2.41           311.         5.45
## 9  1000          224.        1.47           310.         3.16
```

```r
# Packages
library(MASS)
library(randomForest)
library(ggplot2)
```

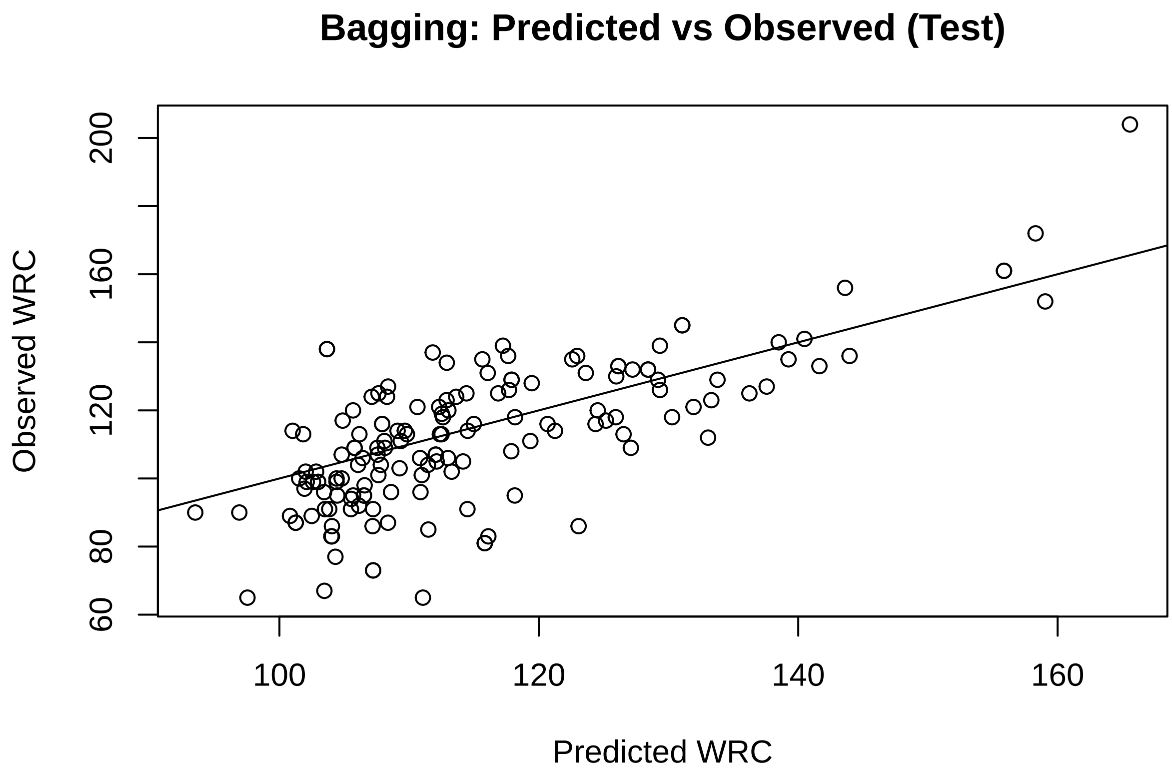
```r
set.seed(1)
N <- nrow(data1)
train <- which(data1$year == 2024)

mlb.train <- data1[train, ]
mlb.test  <- data1[-train, ]

# Response vectors
y.test <- mlb.test$WRC.

set.seed(1)
bag.mlb <- randomForest(WRC. ~ ., data = (data1 %>% select(-year)),
                           subset = train, mtry = 12,
                           importance = TRUE)
bag.mlb
```

```
##
## Call:
##  randomForest(formula = WRC. ~ ., data = (data1 %>% select(-year)),      mtry = 12, importance = TRUI
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 12
##
##           Mean of squared residuals: 307.8028
##                     % Var explained: 36.2
```

```r
yhat.bag <- predict(bag.mlb, newdata = mlb.test)

# Test MSE
mse_bag <- mean((yhat.bag - y.test)^2)
mse_bag
```

```
## [1] 217.4684
```

```r
# Plot predicted vs observed
plot(yhat.bag, y.test,
     xlab = "Predicted WRC",
     ylab = "Observed WRC",
     main = "Bagging: Predicted vs Observed (Test)")
abline(0, 1)
```




```r
set.seed(1)
rf.mlb <- randomForest(WRC. ~ ., data = (data1 %>% select(-year)),
                          subset = train, mtry = 6,
                          importance = TRUE)

yhat.rf <- predict(rf.mlb, newdata = mlb.test)
mse_rf <- mean((yhat.rf - y.test)^2)
c(mse_bag = mse_bag, mse_rf = mse_rf)
```

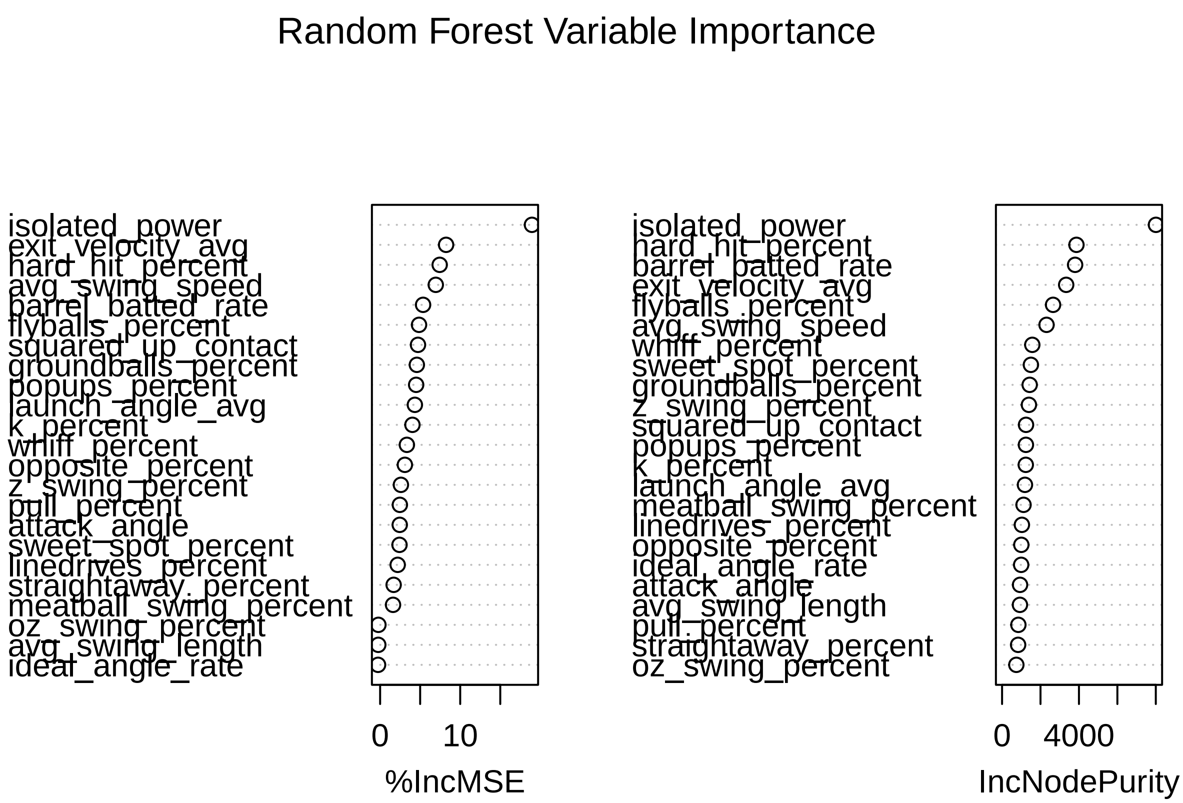
```
##  mse_bag   mse_rf
## 217.4684 225.4536
```

```r
importance(rf.mlb)
```

```
##                         %IncMSE IncNodePurity
## k_percent             4.0342178     1234.0679
## isolated_power       18.9553344     8007.4562
## avg_swing_speed       6.9482535     2314.1854
## squared_up_contact    4.7277869     1246.6261
## avg_swing_length     -0.2266313      929.3655
## attack_angle          2.4506400      935.4998
## ideal_angle_rate     -0.2542535      993.0373
## exit_velocity_avg     8.2369485     3335.9887
## launch_angle_avg      4.3392748     1190.4286
```

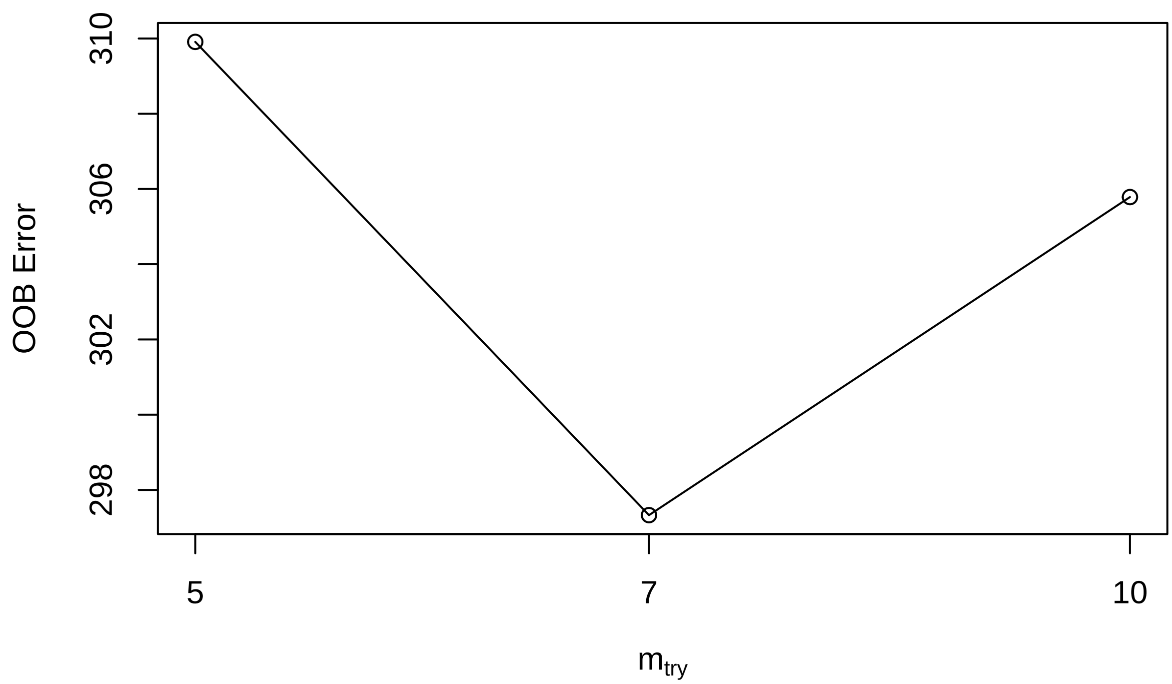
```
## sweet_spot_percent       2.4218317     1499.3724
## barrel_batted_rate       5.3643746     3800.6089
## hard_hit_percent         7.4387125     3870.8463
## z_swing_percent          2.5955117     1397.6026
## oz_swing_percent        -0.2127629      742.7318
## meatball_swing_percent   1.6135249     1117.3377
## whiff_percent            3.3409023     1571.7090
## pull_percent             2.4646820      848.0517
## straightaway_percent     1.6897713      835.0008
## opposite_percent         3.0916951     1002.2378
## groundballs_percent      4.5849651     1437.3837
## flyballs_percent         4.8595681     2653.8116
## linedrives_percent       2.1947863     1030.7874
## popups_percent           4.4976473     1240.8634
```

```r
varImpPlot(rf.mlb, main = "Random Forest Variable Importance")
```




```r
set.seed(2)
tune.out <- tuneRF(x = mlb.train[, setdiff(names(data1 %>% select(-year)), "WRC.")],
                   y = mlb.train$WRC.,
                   stepFactor = 1.5,
                   improve = 0.01,
                   ntreeTry = 500,
                   trace = TRUE,
                   plot = TRUE)
```

```
## mtry = 7  OOB error = 297.33
## Searching left ...
## mtry = 5     OOB error = 309.9098
## -0.04230923 0.01
## Searching right ...
## mtry = 10    OOB error = 305.7852
## -0.02843695 0.01
```



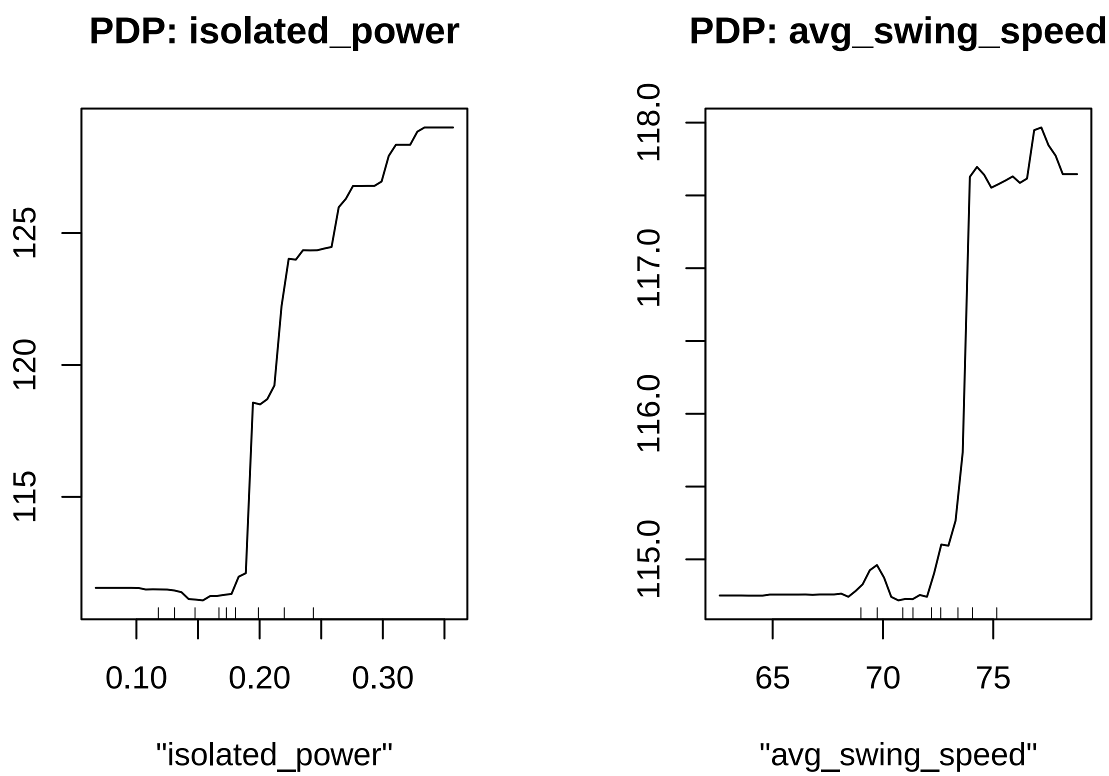

```r
# Best mtry from tuneRF (lower OOB error is better)
best.mtry <- tune.out[which.min(tune.out[, "OOBError"]), "mtry"]
best.mtry
```

```
## [1] 7
```

```r
set.seed(3)
rf.tuned <- randomForest(WRC. ~ ., data = data1 %>% select(-year),
                         subset = train, mtry = best.mtry,
                         ntree = 500, importance = TRUE)

mse_rf_tuned <- mean( (predict(rf.tuned, mlb.test) - y.test)^2 )
c(mse_bag = mse_bag, mse_rf = mse_rf, mse_rf_tuned = mse_rf_tuned)
```

```
##      mse_bag       mse_rf mse_rf_tuned
##     217.4684     225.4536     223.0752
```

```r
# Partial dependence plots give the marginal effect of a feature
par(mfrow = c(1, 2))
partialPlot(rf.tuned, pred.data = mlb.test, x.var = "isolated_power",
            main = "PDP: isolated_power")
partialPlot(rf.tuned, pred.data = mlb.test, x.var = "avg_swing_speed",
            main = "PDP: avg_swing_speed")
```




```r
#using 2025 data

predictions <- predict(rf.tuned, mlb.test)
data1 <- read.csv('onlyCompleteData.csv')
players <- (data1 %>% filter(year==2025))$player.name
actuals <- (data1 %>% filter(year==2025))$WRC.

predict_df <- data.frame(player = players, prediction = predictions, actual = actuals)
predict_df <- predict_df %>% mutate(diff = actuals - predictions)
```