HW 6 Q5

Ben Howell

4/14/2022

```
suppressMessages(require(tidyverse))
suppressMessages(require(janitor))
suppressMessages(require(purrr))
suppressWarnings(require(leaps))
```

```
## Loading required package: leaps
```

```
suppressMessages(require(ggthemes))
suppressMessages(require(ISLR2))
suppressMessages(require(randomForest))
suppressMessages(require(gbm))
suppressMessages(require(ggthemes))
# generate p = 20, n = 1000

set.seed(123)

df <- Hitters %>%
  dplyr::filter(! is.na(Salary)) %>%
  dplyr::mutate(
    log_salary = log(Salary)
  ) %>%
  dplyr::select(-c(Salary)) # %>%
  # rownames_to_column("player_name")

train <- df[1:200, ]
test <- df[201:nrow(df), ]

shrk <- seq(0.001, 1, by = 0.005)

lst <- list()
n <- 0

for (y in shrk) {
  # print(y)

  n <- n + 1
```

```
  mod <- gbm(log_salary ~ .,
             data = train,
             shrinkage = shrk,
             n.trees = 1000,
             distribution = "gaussian")

  train$pred_sal <- predict(mod, train, n.trees = 1000)
  test$pred_sal <- predict(mod, test, n.trees = 1000)

  mse <- mean((train$log_salary - train$pred_sal)^2)
  tmse <- mean((test$log_salary - test$pred_sal)^2)

  m <- data.frame("shrinkage" = y,
                  "MSE" = mse,
                  "tMSE" = tmse)

  lst[[n]] <- m

  print(paste0(scales::percent(n / length(shrk)), " of models tested."))

  test <- test %>%
    dplyr::select(-c(pred_sal))
  # took me to long to figure out that this was being used in future models past the first on bc I forg
  train <- train %>%
    dplyr::select(-c(pred_sal))
}
```

```
## [1] "0% of models tested."
## [1] "1% of models tested."
## [1] "2% of models tested."
## [1] "2% of models tested."
## [1] "2% of models tested."
## [1] "3% of models tested."
## [1] "4% of models tested."
## [1] "4% of models tested."
## [1] "4% of models tested."
## [1] "5% of models tested."
## [1] "6% of models tested."
## [1] "6% of models tested."
## [1] "6% of models tested."
## [1] "7% of models tested."
## [1] "8% of models tested."
## [1] "8% of models tested."
## [1] "8% of models tested."
## [1] "9% of models tested."
## [1] "10% of models tested."
## [1] "10% of models tested."
## [1] "10% of models tested."
## [1] "11% of models tested."
## [1] "12% of models tested."
## [1] "12% of models tested."
## [1] "12% of models tested."
## [1] "13% of models tested."
```

```
## [1] "14% of models tested."
## [1] "14% of models tested."
## [1] "14% of models tested."
## [1] "15% of models tested."
## [1] "16% of models tested."
## [1] "16% of models tested."
## [1] "16% of models tested."
## [1] "17% of models tested."
## [1] "18% of models tested."
## [1] "18% of models tested."
## [1] "18% of models tested."
## [1] "19% of models tested."
## [1] "20% of models tested."
## [1] "20% of models tested."
## [1] "20% of models tested."
## [1] "21% of models tested."
## [1] "22% of models tested."
## [1] "22% of models tested."
## [1] "22% of models tested."
## [1] "23% of models tested."
## [1] "23% of models tested."
## [1] "24% of models tested."
## [1] "24% of models tested."
## [1] "25% of models tested."
## [1] "26% of models tested."
## [1] "26% of models tested."
## [1] "26% of models tested."
## [1] "27% of models tested."
## [1] "28% of models tested."
## [1] "28% of models tested."
## [1] "28% of models tested."
## [1] "29% of models tested."
## [1] "29% of models tested."
## [1] "30% of models tested."
## [1] "30% of models tested."
## [1] "31% of models tested."
## [1] "32% of models tested."
## [1] "32% of models tested."
## [1] "32% of models tested."
## [1] "33% of models tested."
## [1] "34% of models tested."
## [1] "34% of models tested."
## [1] "34% of models tested."
## [1] "35% of models tested."
## [1] "36% of models tested."
## [1] "36% of models tested."
## [1] "36% of models tested."
## [1] "37% of models tested."
## [1] "38% of models tested."
## [1] "38% of models tested."
## [1] "38% of models tested."
## [1] "39% of models tested."
## [1] "40% of models tested."
## [1] "40% of models tested."
```

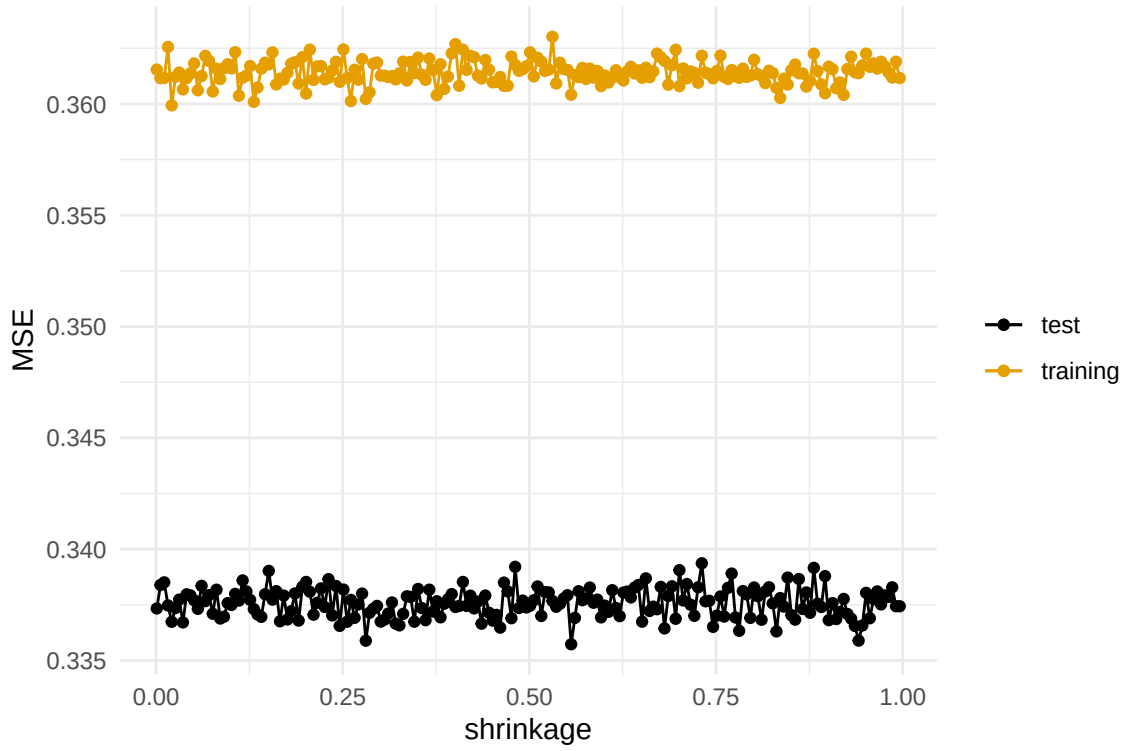
```
## [1] "40% of models tested."
## [1] "41% of models tested."
## [1] "42% of models tested."
## [1] "42% of models tested."
## [1] "42% of models tested."
## [1] "43% of models tested."
## [1] "44% of models tested."
## [1] "44% of models tested."
## [1] "44% of models tested."
## [1] "45% of models tested."
## [1] "46% of models tested."
## [1] "46% of models tested."
## [1] "46% of models tested."
## [1] "47% of models tested."
## [1] "48% of models tested."
## [1] "48% of models tested."
## [1] "48% of models tested."
## [1] "49% of models tested."
## [1] "50% of models tested."
## [1] "50% of models tested."
## [1] "50% of models tested."
## [1] "51% of models tested."
## [1] "52% of models tested."
## [1] "52% of models tested."
## [1] "52% of models tested."
## [1] "53% of models tested."
## [1] "54% of models tested."
## [1] "54% of models tested."
## [1] "54% of models tested."
## [1] "55% of models tested."
## [1] "56% of models tested."
## [1] "56% of models tested."
## [1] "56% of models tested."
## [1] "57% of models tested."
## [1] "57% of models tested."
## [1] "58% of models tested."
## [1] "58% of models tested."
## [1] "59% of models tested."
## [1] "59% of models tested."
## [1] "60% of models tested."
## [1] "60% of models tested."
## [1] "61% of models tested."
## [1] "62% of models tested."
## [1] "62% of models tested."
## [1] "62% of models tested."
## [1] "63% of models tested."
## [1] "64% of models tested."
## [1] "64% of models tested."
## [1] "64% of models tested."
## [1] "65% of models tested."
## [1] "66% of models tested."
## [1] "66% of models tested."
## [1] "66% of models tested."
## [1] "67% of models tested."
```

```
## [1] "68% of models tested."
## [1] "68% of models tested."
## [1] "68% of models tested."
## [1] "69% of models tested."
## [1] "70% of models tested."
## [1] "70% of models tested."
## [1] "70% of models tested."
## [1] "71% of models tested."
## [1] "72% of models tested."
## [1] "72% of models tested."
## [1] "72% of models tested."
## [1] "73% of models tested."
## [1] "74% of models tested."
## [1] "74% of models tested."
## [1] "74% of models tested."
## [1] "75% of models tested."
## [1] "76% of models tested."
## [1] "76% of models tested."
## [1] "76% of models tested."
## [1] "77% of models tested."
## [1] "78% of models tested."
## [1] "78% of models tested."
## [1] "78% of models tested."
## [1] "79% of models tested."
## [1] "80% of models tested."
## [1] "80% of models tested."
## [1] "80% of models tested."
## [1] "81% of models tested."
## [1] "82% of models tested."
## [1] "82% of models tested."
## [1] "82% of models tested."
## [1] "83% of models tested."
## [1] "84% of models tested."
## [1] "84% of models tested."
## [1] "84% of models tested."
## [1] "85% of models tested."
## [1] "86% of models tested."
## [1] "86% of models tested."
## [1] "86% of models tested."
## [1] "87% of models tested."
## [1] "88% of models tested."
## [1] "88% of models tested."
## [1] "88% of models tested."
## [1] "89% of models tested."
## [1] "90% of models tested."
## [1] "90% of models tested."
## [1] "90% of models tested."
## [1] "91% of models tested."
## [1] "92% of models tested."
## [1] "92% of models tested."
## [1] "92% of models tested."
## [1] "93% of models tested."
## [1] "94% of models tested."
## [1] "94% of models tested."
```

```
## [1] "94% of models tested."
## [1] "95% of models tested."
## [1] "96% of models tested."
## [1] "96% of models tested."
## [1] "96% of models tested."
## [1] "97% of models tested."
## [1] "98% of models tested."
## [1] "98% of models tested."
## [1] "98% of models tested."
## [1] "99% of models tested."
## [1] "100% of models tested."
## [1] "100% of models tested."
```

```
res <- dplyr::bind_rows(lst)

res %>%
  ggplot() +
  geom_point(aes(x = shrinkage, y = MSE, color = "training")) +
  geom_line(aes(x = shrinkage, y = MSE, color = "training")) +
  geom_point(aes(x = shrinkage, y = tMSE, color = "test")) +
  geom_line(aes(x = shrinkage, y = tMSE, color = "test")) +
  scale_color_colorblind() +
  theme_minimal() +
  theme(legend.title = element_blank())
```



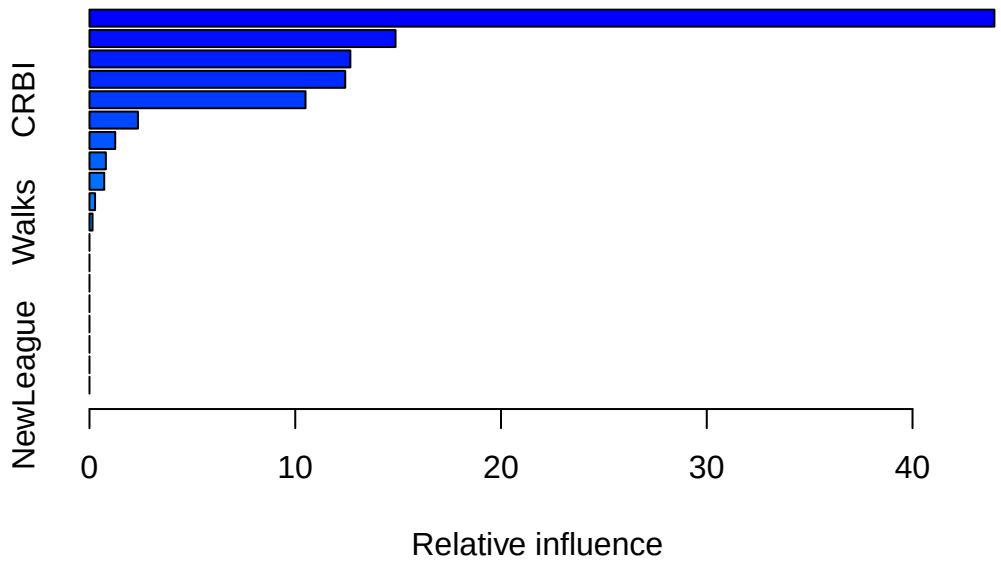

```
ml <- lm(log_salary ~ .,
         data = train)
test$lm_sal <- predict(ml, newdata = test)
```

```
reg_m <- regsubsets(log_salary ~.,
                    data = train,
                    method = "exhaustive")
# summary(reg_m)
five_m <- glm(log_salary ~ .,
              data = train %>%
                dplyr::select(log_salary, AtBat, Hits, Walks, Years, PutOuts))
test$sub_sal <- predict(five_m, newdata = test)

simple_mse <- mean((test$log_salary - test$lm_sal)^2)
subset_mse <- mean((test$log_salary - test$sub_sal)^2)
```

I tried both a simple linear regression model and a linear regression with the five most important variables that I determined through an exhaustive search. The smallest MSE of the test dataset was 0.336, which was significantly lower than the simple MSE of 0.492 and the subset MSE of 0.498.

```
summary.gbm(mod)
```




```
##                 var    rel.inf
## CAtBat       CAtBat 43.9797029
## CHits         CHits 14.8749685
## CWalks       CWalks 12.6785222
## CRuns         CRuns 12.4296907
## CRBI           CRBI 10.4991888
## CHmRun       CHmRun  2.3550572
## Hits           Hits  1.2506969
## Years         Years  0.7963886
```

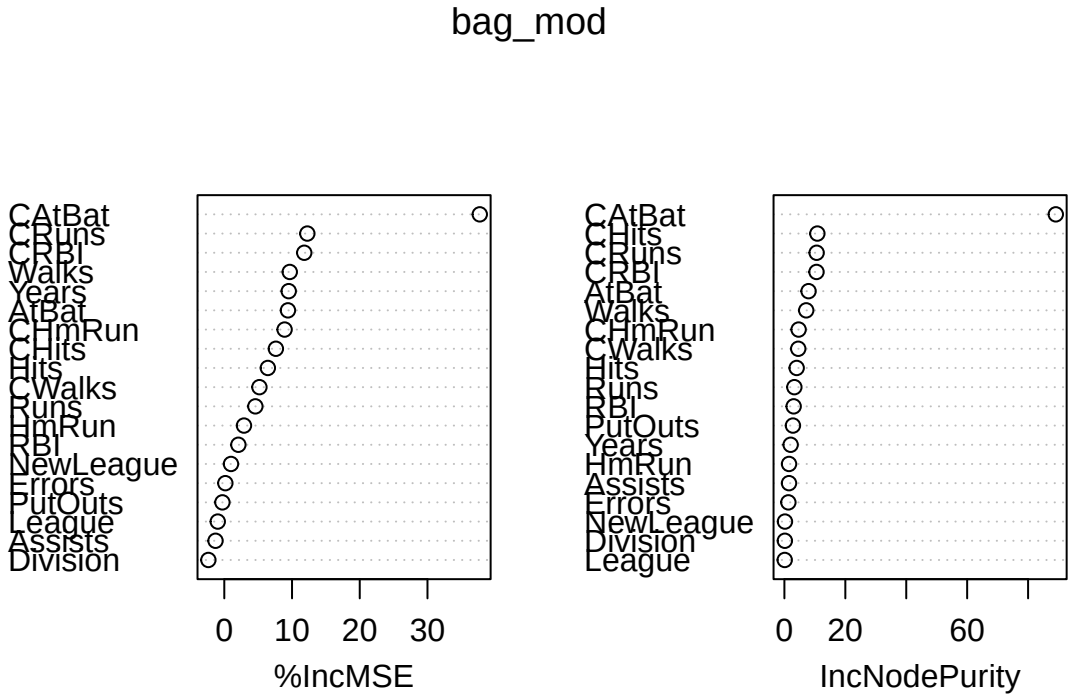
```
## RBI             RBI  0.7161090
## AtBat         AtBat  0.2703954
## Walks         Walks  0.1492798
## HmRun         HmRun  0.0000000
## Runs           Runs  0.0000000
## League       League  0.0000000
## Division   Division  0.0000000
## PutOuts     PutOuts  0.0000000
## Assists     Assists  0.0000000
## Errors       Errors  0.0000000
## NewLeague NewLeague  0.0000000
```

Some of the most important variables are ABs, Walks, Hits, RBI over the course of a career, which was interesting to see that those contributions were valued over the performance in their season.

```
bag_mod <- randomForest(log_salary ~ .,
                        data = train,
                        mtry = ncol(train) - 1,
                        ntree = 500,
                        importance = TRUE)

test$bag_pred <- predict(bag_mod, test)

bag_mse <- mean((test$log_salary - test$bag_pred)^2)
varImpPlot(bag_mod)
```



The bagging MSE of 0.23 is quite a bit lower than the MSE that we got for the GBM using the boosted method, which was certainly interesting to see. Many of the same variables were rated as the most important

across the bagging and boosted method, which seems to imply that the bagging method just did a better job of picking out the specific interactions across features.