MLPS2

Ying Zhou

2020/2/1

1

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ISLR)
library(broom)
library(rsample)
library(rcfss)
library(yardstick)
```

```
## For binary classification, the first factor level is assumed to be the event.
## Set the global option `yardstick.event_first` to `FALSE` to change this.

##
## Attaching package: 'yardstick'

## The following object is masked from 'package:readr':
##
##     spec
```

```
library(ggplot2)
nesdata <- read.csv("C:/Users/zhouy/Desktop/Uchicago/uchicourse/Intro to Machine Learning/PS/PS2/MLPS2_
model1<-lm(biden ~ female+age+educ+dem+rep, data = nesdata)
summary(model1)
```

```
##
## Call:
## lm(formula = biden ~ female + age + educ + dem + rep, data = nesdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.546 -11.295   1.018  12.776  53.977
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  58.81126    3.12444  18.823  < 2e-16 ***
## female        4.10323    0.94823   4.327 1.59e-05 ***
## age           0.04826    0.02825   1.708   0.0877 .
## educ         -0.34533    0.19478  -1.773   0.0764 .
## dem          15.42426    1.06803  14.442  < 2e-16 ***
## rep         -15.84951    1.31136 -12.086  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.91 on 1801 degrees of freedom
## Multiple R-squared:  0.2815, Adjusted R-squared:  0.2795
## F-statistic: 141.1 on 5 and 1801 DF,  p-value: < 2.2e-16
```

```
(mse <- augment(model1, newdata = nesdata)   %>%
mse(truth = biden, estimate = .fitted))
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 mse     standard        395.
```

MSE of the model is 395.27. The average squared difference between the actual values and the predicted values is 395.27. The smaller the MSE, the better the estimators. The MSE here is not small.

2

Split the sample set into a training set (50%) and a holdout set (50%). Be sure to set your seed prior to this part of your code to guarantee reproducibility of results.

```
set.seed(1234)
t_split <- initial_split(data = nesdata,
                         prop=0.5)
t_train <- training(t_split)
t_test <- testing(t_split)
```

Fit the linear regression model using only the training observations

```
t_lm <- lm(biden ~ female+age+educ+dem+rep, data = t_train)
summary(t_lm)
```

```
##
```

```
## Call:
## lm(formula = biden ~ female + age + educ + dem + rep, data = t_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.880 -11.950   1.929  11.899  46.124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  58.68937    4.30323  13.638  < 2e-16 ***
## female        4.41344    1.28889   3.424 0.000644 ***
## age           0.04460    0.03858   1.156 0.247980
## educ         -0.18263    0.26831  -0.681 0.496251
## dem          13.63872    1.45353   9.383  < 2e-16 ***
## rep         -18.76842    1.78349 -10.523  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.11 on 898 degrees of freedom
## Multiple R-squared:  0.3085, Adjusted R-squared:  0.3046
## F-statistic: 80.12 on 5 and 898 DF,  p-value: < 2.2e-16
```

Calculate the MSE using only the test set observations

```
(test_mse <- augment(t_lm, newdata = t_test) %>%
    mse(truth = biden, estimate = .fitted))
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 mse     standard        432.
```

How does this value compare to the training MSE from question 1? Present numeric comparison and discuss a bit.

The test MSE in this question is 431.6 which is greater than the MSE in question 1. It means that the simple 1-stage holdout validation approach is problematic, and the model does not fit the test set observations as well as question 1.

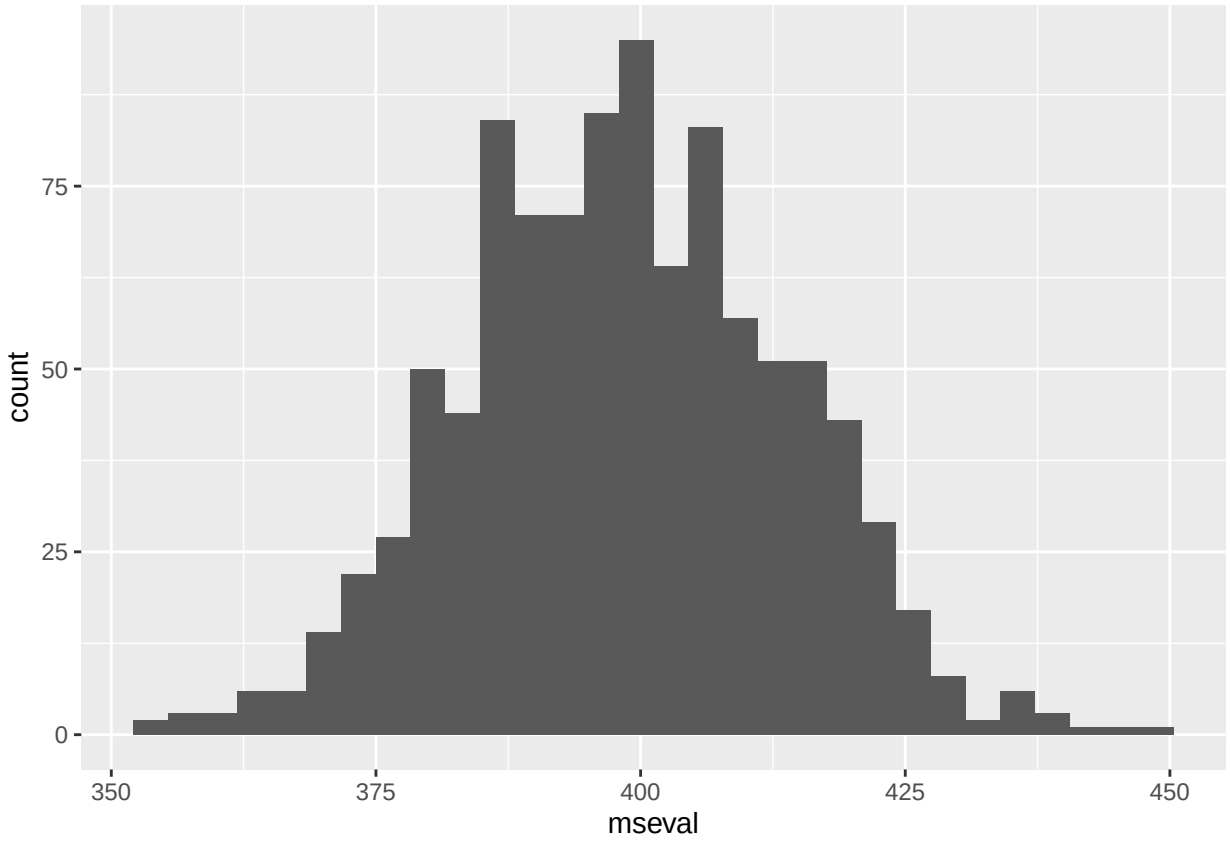
3

```
mseval<-c()
for (i in 1:1000){
  val_split<-initial_split(data = nesdata,
                           prop=0.5)
  val_train<-training(val_split)
  val_test<-testing(val_split)
  val_lm <- lm(biden ~ female+age+educ+dem+rep, data = val_train)
  test_mseval <- augment(val_lm, newdata = val_test) %>%
    mse(truth = biden, estimate = .fitted)
  mseval[[i]]<-test_mseval$.estimate
}
```

```
msedf <- data.frame(matrix(mseval))

ggplot(msedf, aes(x = mseval)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The distribution is nearly the normal distribution with mean roughly equal to 400, which is closed to the MSE in question 1. And the MSE in question 2 falls in the tail of this normal distribution. It means the repeated holdout validation can help to obtain a more robust performance estimate.

4

```
# traditional parameter estimates and standard errors
nes <- as_tibble(nesdata)
model2<-lm(biden ~ female+age+educ+dem+rep, data = nes)
tidy(model2)
```

```
## # A tibble: 6 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  58.8       3.12       18.8  2.69e-72
## 2 female        4.10      0.948       4.33 1.59e- 5
```

```
## 3 age           0.0483    0.0282      1.71 8.77e- 2
## 4 educ         -0.345     0.195      -1.77 7.64e- 2
## 5 dem          15.4       1.07       14.4  8.14e-45
## 6 rep         -15.8       1.31      -12.1  2.16e-32
```

```
# bootstrapped estimates of the parameter estimates and standard errors
lm_coefs <- function(splits, ...) {
  ## use `analysis` or `as.data.frame` to get the analysis data
  mod <- lm(..., data = analysis(splits))
  tidy(mod)
}

nes_boot <- nes %>%
  bootstraps(1000) %>%
  mutate(coef = map(splits, lm_coefs, as.formula(biden ~ female+age+educ+dem+rep)))

nes_boot %>%
  unnest(coef) %>%
  group_by(term) %>%
  summarize(.estimate = mean(estimate),
            .se = sd(estimate, na.rm = TRUE))
```

```
## # A tibble: 6 x 3
##   term        .estimate    .se
##   <chr>           <dbl>  <dbl>
## 1 (Intercept)   58.9    2.97
## 2 age            0.0468 0.0297
## 3 dem           15.4    1.03
## 4 educ          -0.347  0.187
## 5 female         4.12   0.920
## 6 rep          -15.8    1.41
```

The estimated parameters and standard errors from the original model in question 1 and the parameters using the bootstrap are nearly the same. Small differences on parameters and standard errors are caused by the random resampling from the empirical distribution. In bootstrap, we resampling from the sample while in question 1 we just use original dataset. The impact of bootstrap is that it may cause problems like biased (non-generalizable) samples.