Module 3 Assign1

Je’Kolby Worthy

2/8/2021

# Module 3 Assignment 1

## Model Validation

### Je’Kolby Worthy

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.0.5 ✓ dplyr 1.0.3  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 0.1.2 ──

## ✓ broom 0.7.3 ✓ recipes 0.1.15  
## ✓ dials 0.0.9 ✓ rsample 0.0.8   
## ✓ infer 0.5.4 ✓ tune 0.1.2   
## ✓ modeldata 0.1.0 ✓ workflows 0.2.1   
## ✓ parsnip 0.1.5 ✓ yardstick 0.0.7

## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## x scales::discard() masks purrr::discard()  
## x dplyr::filter() masks stats::filter()  
## x recipes::fixed() masks stringr::fixed()  
## x dplyr::lag() masks stats::lag()  
## x yardstick::spec() masks readr::spec()  
## x recipes::step() masks stats::step()

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

bike <- read\_csv("bike\_cleaned-2.csv")

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## instant = col\_double(),  
## dteday = col\_character(),  
## season = col\_character(),  
## mnth = col\_character(),  
## hr = col\_double(),  
## holiday = col\_character(),  
## weekday = col\_character(),  
## workingday = col\_character(),  
## weathersit = col\_character(),  
## temp = col\_double(),  
## atemp = col\_double(),  
## hum = col\_double(),  
## windspeed = col\_double(),  
## casual = col\_double(),  
## registered = col\_double(),  
## count = col\_double()  
## )

bike = bike %>% mutate(dteday = mdy(dteday))   
bike = bike %>% mutate\_if(is.character,as\_factor)  
bike = bike %>% mutate(hr = as.factor(hr))

set.seed(1234)  
bike\_spilt = initial\_split(bike, prob = 0.70, strata = count)  
train = training(bike\_spilt)  
test = testing(bike\_spilt)

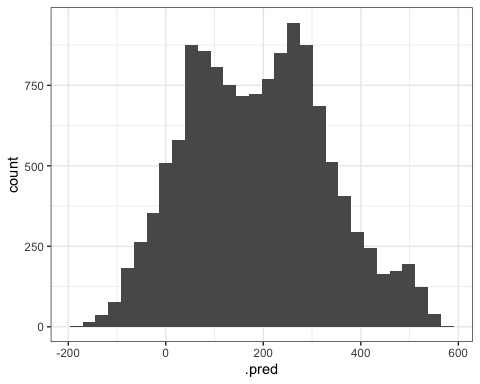
How many rows of data are in each set (training and testing)? **The training set have 13,036 rows, while in the testing set have 4,343 rows.**

bike\_recipe = recipe(count ~ season + mnth + hr + holiday + weekday + temp + weathersit, train)  
  
lm\_model =  
 linear\_reg()%>%  
 set\_engine("lm")  
  
lm\_wflow =  
 workflow()%>%  
 add\_model(lm\_model)%>%  
 add\_recipe(bike\_recipe)  
  
lm\_fit = fit(lm\_wflow, train)  
summary(lm\_fit$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -413.11 -61.65 -10.20 52.16 493.99   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -85.3666 6.7480 -12.651 < 2e-16 \*\*\*  
## seasonSpring 36.4869 6.0975 5.984 2.24e-09 \*\*\*  
## seasonSummer 31.2531 7.2079 4.336 1.46e-05 \*\*\*  
## seasonFall 62.3139 6.1169 10.187 < 2e-16 \*\*\*  
## mnthFeb 1.1497 4.9544 0.232 0.816495   
## mnthMar 6.1995 5.6065 1.106 0.268842   
## mnthApr -8.0391 8.2890 -0.970 0.332141   
## mnthMay -7.9182 8.8342 -0.896 0.370100   
## mnthJun -24.0790 9.0886 -2.649 0.008074 \*\*   
## mnthJul -46.7354 10.1901 -4.586 4.55e-06 \*\*\*  
## mnthAug -27.7663 9.8890 -2.808 0.004996 \*\*   
## mnthSep 0.3262 8.7760 0.037 0.970353   
## mnthOct -2.8693 8.1729 -0.351 0.725537   
## mnthNov -17.9871 7.8670 -2.286 0.022247 \*   
## mnthDec -12.6331 6.2366 -2.026 0.042824 \*   
## hr1 -17.6394 6.7829 -2.601 0.009318 \*\*   
## hr2 -24.7408 6.7860 -3.646 0.000268 \*\*\*  
## hr3 -36.3172 6.7857 -5.352 8.85e-08 \*\*\*  
## hr4 -39.8317 6.8741 -5.794 7.01e-09 \*\*\*  
## hr5 -23.5341 6.8326 -3.444 0.000574 \*\*\*  
## hr6 34.9075 6.7378 5.181 2.24e-07 \*\*\*  
## hr7 170.4187 6.7576 25.219 < 2e-16 \*\*\*  
## hr8 310.2081 6.7874 45.703 < 2e-16 \*\*\*  
## hr9 167.5555 6.6896 25.047 < 2e-16 \*\*\*  
## hr10 112.2824 6.7742 16.575 < 2e-16 \*\*\*  
## hr11 139.9731 6.7959 20.597 < 2e-16 \*\*\*  
## hr12 180.4694 6.8816 26.225 < 2e-16 \*\*\*  
## hr13 182.6847 6.8514 26.664 < 2e-16 \*\*\*  
## hr14 163.6753 6.8350 23.947 < 2e-16 \*\*\*  
## hr15 168.7255 6.8956 24.469 < 2e-16 \*\*\*  
## hr16 228.8081 6.8944 33.187 < 2e-16 \*\*\*  
## hr17 380.6338 6.8048 55.936 < 2e-16 \*\*\*  
## hr18 355.7561 6.8635 51.833 < 2e-16 \*\*\*  
## hr19 244.4088 6.7834 36.031 < 2e-16 \*\*\*  
## hr20 160.9975 6.8198 23.607 < 2e-16 \*\*\*  
## hr21 110.3631 6.7372 16.381 < 2e-16 \*\*\*  
## hr22 73.3439 6.7251 10.906 < 2e-16 \*\*\*  
## hr23 34.8460 6.7667 5.150 2.65e-07 \*\*\*  
## holidayHoliday -27.9348 6.1853 -4.516 6.35e-06 \*\*\*  
## weekdaySunday -18.8199 3.6318 -5.182 2.23e-07 \*\*\*  
## weekdayMonday -7.3613 3.7407 -1.968 0.049102 \*   
## weekdayTuesday -6.4130 3.6577 -1.753 0.079581 .   
## weekdayWednesday -3.1261 3.6405 -0.859 0.390522   
## weekdayThursday -3.2735 3.6387 -0.900 0.368340   
## weekdayFriday 1.2942 3.6295 0.357 0.721414   
## temp 289.3663 11.7834 24.557 < 2e-16 \*\*\*  
## weathersitMisty -20.7095 2.2908 -9.040 < 2e-16 \*\*\*  
## weathersitLightPrecip -92.1176 3.6283 -25.389 < 2e-16 \*\*\*  
## weathersitHeavyPrecip -41.2406 78.7048 -0.524 0.600294   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 111.1 on 12987 degrees of freedom  
## Multiple R-squared: 0.6243, Adjusted R-squared: 0.6229   
## F-statistic: 449.6 on 48 and 12987 DF, p-value: < 2.2e-16

predict\_train = predict(lm\_fit, train)  
  
ggplot(predict\_train, aes(x=.pred)) + geom\_histogram() + theme\_bw()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



lm\_fit %>% predict(test) %>% bind\_cols(test) %>% metrics(truth = count, estimate = ".pred")

## # A tibble: 3 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 112.   
## 2 rsq standard 0.623  
## 3 mae standard 81.6

Determine the R-squared value of the model on the testing set. Comment on how this value compares to the model’s performance on the training set. **On the testing set, the R-Squared value is 0.6229, while the R-Squared value on the training set is also 0.6229. This means that the model does not over fit, making it a strong model.**