Model Validation

# Model Validation

## MIS502

### Danielle Harris

library(tidyverse)

## ── Attaching packages ────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.0 ✔ purrr 0.2.5  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.7  
## ✔ tidyr 0.8.2 ✔ stringr 1.3.1  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

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

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

bike = read\_csv("hour.csv")

## Parsed with column specification:  
## cols(  
## instant = col\_integer(),  
## dteday = col\_date(format = ""),  
## season = col\_integer(),  
## yr = col\_integer(),  
## mnth = col\_integer(),  
## hr = col\_integer(),  
## holiday = col\_integer(),  
## weekday = col\_integer(),  
## workingday = col\_integer(),  
## weathersit = col\_integer(),  
## temp = col\_double(),  
## atemp = col\_double(),  
## hum = col\_double(),  
## windspeed = col\_double(),  
## casual = col\_integer(),  
## registered = col\_integer(),  
## count = col\_integer()  
## )

bike = bike %>% mutate(season = as\_factor(as.character(season))) %>%  
mutate(season = fct\_recode(season,  
"Spring" = "1",  
"Summer" = "2",  
"Fall" = "3",  
"Winter" = "4"))

set.seed(1234)  
train.rows = createDataPartition(y = bike$count, p= 0.7, list= FALSE)  
train = bike [train.rows,]  
test = bike [-train.rows,]

# There are 12,167 rows in training and 5,212 rows in testing.

mod1 = lm (count ~ season + mnth + hr + holiday + weekday + temp + weathersit, train)  
summary(mod1)

##   
## Call:  
## lm(formula = count ~ season + mnth + hr + holiday + weekday +   
## temp + weathersit, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -331.52 -100.93 -28.72 58.17 708.37   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -72.8723 6.4614 -11.278 < 2e-16 \*\*\*  
## seasonSummer -4.5518 4.8949 -0.930 0.352439   
## seasonFall -42.9103 6.9186 -6.202 5.75e-10 \*\*\*  
## seasonWinter 42.5536 6.9096 6.159 7.57e-10 \*\*\*  
## mnth -1.2529 0.7304 -1.715 0.086332 .   
## hr 8.5906 0.2019 42.543 < 2e-16 \*\*\*  
## holiday -20.1203 8.3752 -2.402 0.016304 \*   
## weekday 2.6391 0.6828 3.865 0.000112 \*\*\*  
## temp 418.7385 11.7083 35.764 < 2e-16 \*\*\*  
## weathersit -29.9031 2.1292 -14.044 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 150 on 12157 degrees of freedom  
## Multiple R-squared: 0.31, Adjusted R-squared: 0.3095   
## F-statistic: 607 on 9 and 12157 DF, p-value: < 2.2e-16

# The quality of the model is low, given that the adjusted R-squared value is closer to 0 than 1. Additionally, the p-value is <0.05.

prediction\_train = predict(mod1, newdata= train)  
head(prediction\_train, 6)

## 1 2 3 4 5 6   
## 12.51940 21.11003 38.07543 46.66606 25.35358 55.47255

# Given the training predictions, count is likely to be higher based on temperature.

prediction\_test = predict(mod1, newdata= test)  
head( prediction\_test, 6)

## 1 2 3 4 5 6   
## 12.30355 123.11829 157.04909 190.76403 186.20108 58.68837

# Given the testing predictions, count is likely to be higher based on holiday.

SSE= sum((test$count - prediction\_test)^2)  
SST= sum((test$count - mean(test$count))^2)  
1-SSE/SST

## [1] 0.3120587

# The R-squared value is only slightly larger than the R-squared value for the training set, meaning that the training set model is likely to pe4rform simiarly on new data.

ctrl = trainControl(method= "cv", number =10)  
set.seed(1234)  
modcv = train(count ~ season + mnth + hr + holiday + weekday + temp + weathersit, bike, method= "lm", trControl = ctrl, metric= "R-squared")

## Warning in train.default(x, y, weights = w, ...): The metric "R-squared"  
## was not in the result set. RMSE will be used instead.

summary(modcv)

##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -332.65 -101.33 -28.85 58.53 709.24   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -74.8912 5.4273 -13.799 < 2e-16 \*\*\*  
## seasonSummer -3.2920 4.1451 -0.794 0.427097   
## seasonFall -42.6974 5.8241 -7.331 2.38e-13 \*\*\*  
## seasonWinter 47.6051 5.7873 8.226 < 2e-16 \*\*\*  
## mnth -1.3772 0.6113 -2.253 0.024275 \*   
## hr 8.6707 0.1698 51.056 < 2e-16 \*\*\*  
## holiday -24.0507 6.8858 -3.493 0.000479 \*\*\*  
## weekday 2.6074 0.5729 4.551 5.37e-06 \*\*\*  
## temp 419.3939 9.8520 42.569 < 2e-16 \*\*\*  
## weathersit -29.4998 1.7995 -16.393 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 150.6 on 17369 degrees of freedom  
## Multiple R-squared: 0.3108, Adjusted R-squared: 0.3104   
## F-statistic: 870.2 on 9 and 17369 DF, p-value: < 2.2e-16

# The adjusted R-squared was slightly larger in the k-fold cross validation than the model validation via testing/training split, but the p-values remained the same. The amount of work for k-fold cross validation was more simple, with fewer steps than the model validation via testing/training.