## Stuart Broach

## Model Validation

#install.packages(MASS)  
#install.packages(caret)  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.5.2

## -- Attaching packages ---------------------------------------------------------------- tidyverse 1.2.1 --

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

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

library(MASS)

## Warning: package 'MASS' was built under R version 3.5.2

##   
## Attaching package: 'MASS'

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

library(caret)

## Warning: package 'caret' was built under R version 3.5.2

## 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"))  
  
bike = bike %>% mutate(yr = as\_factor(as.character(yr))) %>% mutate(mnth = as\_factor(as.character(mnth))) %>% mutate(hr = as\_factor(as.character(hr)))  
  
bike = bike %>% mutate(holiday = as\_factor(as.character(holiday))) %>%  
 mutate(holiday = fct\_recode(holiday,  
 "NotHoliday" = "0",  
"Holiday" = "1"))  
  
bike = bike %>% mutate(workingday = as\_factor(as.character(workingday))) %>%  
 mutate(workingday = fct\_recode(workingday,  
"NotWorkingDay" = "0",  
"WorkingDay" = "1"))  
  
bike = bike %>% mutate(weathersit = as\_factor(as.character(weathersit))) %>%  
 mutate(weathersit = fct\_recode(weathersit,  
"NoPrecip" = "1",  
"Misty" = "2",  
"LightPrecip" = "3",  
 "HeavyPrecip" = "4"))  
  
bike = bike %>% mutate(weekday = as\_factor(as.character(weekday))) %>%  
 mutate(weekday = fct\_recode(weekday,  
"Sunday" = "0",  
"Monday" = "1",  
"Tuesday" = "2",  
"Wednesday" = "3",  
"Thursday" = "4",  
"Friday" = "5",  
"Saturday" = "6"))  
glimpse(bike)

## Observations: 17,379  
## Variables: 17  
## $ instant <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ...  
## $ dteday <date> 2011-01-01, 2011-01-01, 2011-01-01, 2011-01-01, 20...  
## $ season <fct> Spring, Spring, Spring, Spring, Spring, Spring, Spr...  
## $ yr <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...  
## $ mnth <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...  
## $ hr <fct> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1...  
## $ holiday <fct> NotHoliday, NotHoliday, NotHoliday, NotHoliday, Not...  
## $ weekday <fct> Saturday, Saturday, Saturday, Saturday, Saturday, S...  
## $ workingday <fct> NotWorkingDay, NotWorkingDay, NotWorkingDay, NotWor...  
## $ weathersit <fct> NoPrecip, NoPrecip, NoPrecip, NoPrecip, NoPrecip, M...  
## $ temp <dbl> 0.24, 0.22, 0.22, 0.24, 0.24, 0.24, 0.22, 0.20, 0.2...  
## $ atemp <dbl> 0.2879, 0.2727, 0.2727, 0.2879, 0.2879, 0.2576, 0.2...  
## $ hum <dbl> 0.81, 0.80, 0.80, 0.75, 0.75, 0.75, 0.80, 0.86, 0.7...  
## $ windspeed <dbl> 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0896, 0.0...  
## $ casual <int> 3, 8, 5, 3, 0, 0, 2, 1, 1, 8, 12, 26, 29, 47, 35, 4...  
## $ registered <int> 13, 32, 27, 10, 1, 1, 0, 2, 7, 6, 24, 30, 55, 47, 7...  
## $ count <int> 16, 40, 32, 13, 1, 1, 2, 3, 8, 14, 36, 56, 84, 94, ...

set.seed(123)  
train.rows = createDataPartition(y = bike$count, p=0.7, list = FALSE)  
train = bike[train.rows,]  
test = bike[-train.rows,]

There are 5212 rows of data in the test set and 12167 in the train set.

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

##   
## Call:  
## lm(formula = count ~ season + mnth + hr + holiday + weekday +   
## temp + weathersit, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -381.09 -62.17 -9.89 52.38 497.92   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -88.2288 6.8830 -12.818 < 2e-16 \*\*\*  
## seasonSummer 36.0728 6.3700 5.663 1.52e-08 \*\*\*  
## seasonFall 24.8948 7.5477 3.298 0.000975 \*\*\*  
## seasonWinter 65.8226 6.3920 10.298 < 2e-16 \*\*\*  
## mnth2 0.9124 5.1243 0.178 0.858688   
## mnth3 3.9284 5.6909 0.690 0.490029   
## mnth4 -5.3658 8.5409 -0.628 0.529855   
## mnth5 -6.5966 9.1165 -0.724 0.469332   
## mnth6 -19.9880 9.3745 -2.132 0.033014 \*   
## mnth7 -44.4010 10.5230 -4.219 2.47e-05 \*\*\*  
## mnth8 -19.5075 10.2459 -1.904 0.056942 .   
## mnth9 4.1128 9.1124 0.451 0.651753   
## mnth10 -7.0933 8.4691 -0.838 0.402301   
## mnth11 -21.9296 8.1614 -2.687 0.007220 \*\*   
## mnth12 -18.1554 6.4929 -2.796 0.005179 \*\*   
## hr1 -13.3717 6.9674 -1.919 0.054986 .   
## hr2 -25.7624 6.8773 -3.746 0.000181 \*\*\*  
## hr3 -39.2527 7.0707 -5.551 2.89e-08 \*\*\*  
## hr4 -40.1235 7.0046 -5.728 1.04e-08 \*\*\*  
## hr5 -23.1163 7.0131 -3.296 0.000983 \*\*\*  
## hr6 33.6188 6.9286 4.852 1.24e-06 \*\*\*  
## hr7 169.8889 6.9686 24.379 < 2e-16 \*\*\*  
## hr8 307.5194 6.8944 44.604 < 2e-16 \*\*\*  
## hr9 164.2925 6.9360 23.687 < 2e-16 \*\*\*  
## hr10 114.0912 7.0011 16.296 < 2e-16 \*\*\*  
## hr11 138.1606 7.0536 19.587 < 2e-16 \*\*\*  
## hr12 183.8968 6.9477 26.469 < 2e-16 \*\*\*  
## hr13 176.8968 7.0068 25.246 < 2e-16 \*\*\*  
## hr14 155.3241 7.0204 22.125 < 2e-16 \*\*\*  
## hr15 171.0530 7.0412 24.293 < 2e-16 \*\*\*  
## hr16 227.1267 6.9243 32.801 < 2e-16 \*\*\*  
## hr17 380.5829 6.9879 54.463 < 2e-16 \*\*\*  
## hr18 353.0159 6.9669 50.671 < 2e-16 \*\*\*  
## hr19 237.9453 6.9013 34.479 < 2e-16 \*\*\*  
## hr20 163.0001 6.9654 23.402 < 2e-16 \*\*\*  
## hr21 109.3346 6.9480 15.736 < 2e-16 \*\*\*  
## hr22 72.1642 6.8967 10.464 < 2e-16 \*\*\*  
## hr23 34.6697 7.0092 4.946 7.66e-07 \*\*\*  
## holidayHoliday -25.0434 6.2847 -3.985 6.79e-05 \*\*\*  
## weekdaySunday -11.4516 3.7685 -3.039 0.002380 \*\*   
## weekdayMonday -3.1214 3.8861 -0.803 0.421854   
## weekdayTuesday -1.9526 3.7660 -0.518 0.604131   
## weekdayWednesday 1.5959 3.7657 0.424 0.671714   
## weekdayThursday 0.3010 3.7738 0.080 0.936422   
## weekdayFriday 5.3773 3.7720 1.426 0.154016   
## temp 289.3605 12.1362 23.843 < 2e-16 \*\*\*  
## weathersitMisty -21.0867 2.3771 -8.871 < 2e-16 \*\*\*  
## weathersitLightPrecip -93.3272 3.7526 -24.870 < 2e-16 \*\*\*  
## weathersitHeavyPrecip -42.0316 78.9520 -0.532 0.594480   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 111.4 on 12118 degrees of freedom  
## Multiple R-squared: 0.6236, Adjusted R-squared: 0.6221   
## F-statistic: 418.2 on 48 and 12118 DF, p-value: < 2.2e-16

This is a good model right here. .62 r squared value and a p-value less then .05.

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

## 1 2 3 4 5 6   
## -37.94115 -50.33188 -58.03498 -58.90576 -62.98525 139.53224

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

## 1 2 3 4 5 6   
## -18.782265 9.049333 168.659049 217.199397 179.114378 398.586025

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

## [1] 0.6244417

The two models matched up erfectly at .62.

K-fold cross-validation differs from model validation via a training/testing split by splitting the data into partitions. It evaluates a model k times while leacing out one of the folds each iteration. This helps to reduce overfitting.