## BAN 502

## Module 3: Model Validation

## Mandy Hawkins

# 

library(tidyverse)

## -- 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(readr)  
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("~/BAN502/Module 3/Model Validation Assignment/Model Validation Assignment/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()  
## )

View(bike)

bike = bike %>% drop\_na() #delete any row with an NA value   
str(bike) #check structure after the drop

## Classes 'tbl\_df', 'tbl' and 'data.frame': 17379 obs. of 17 variables:  
## $ instant : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ dteday : Date, format: "2011-01-01" "2011-01-01" ...  
## $ season : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ yr : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ mnth : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ hr : int 0 1 2 3 4 5 6 7 8 9 ...  
## $ holiday : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ weekday : int 6 6 6 6 6 6 6 6 6 6 ...  
## $ workingday: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ weathersit: int 1 1 1 1 1 2 1 1 1 1 ...  
## $ temp : num 0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...  
## $ atemp : num 0.288 0.273 0.273 0.288 0.288 ...  
## $ hum : num 0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...  
## $ windspeed : num 0 0 0 0 0 0.0896 0 0 0 0 ...  
## $ casual : int 3 8 5 3 0 0 2 1 1 8 ...  
## $ registered: int 13 32 27 10 1 1 0 2 7 6 ...  
## $ count : int 16 40 32 13 1 1 2 3 8 14 ...

# convert "season" to a factor and to rename the factor levels  
bike = bike %>% mutate(season = as\_factor(as.character(season))) %>% mutate(season = fct\_recode(season, "Spring" = "1", "Summer" = "2", "Fall" = "3", "Winter" = "4"))

#Convert "yr", "mnth", and "hr" to factors  
bike = bike %>% mutate(mnth = as\_factor(as.character(mnth)))  
bike = bike %>% mutate(hr = as\_factor(as.character(hr)))  
bike = bike %>% mutate(yr = as\_factor(as.character(yr)))

#Convert the "holiday" variable to a factor and recode the levels from 0 to "NotHoliday" and 1 to "Holiday".  
bike = bike %>% mutate(holiday = as\_factor(as.character(holiday))) %>% mutate(holiday = fct\_recode(holiday, "NotHoliday" = "0", "Holiday" = "1"))

#Convert "workingday" to a factor and recode the levels from 0 to "NotWorkingDay" and 1 to "WorkingDay".   
  
bike = bike %>% mutate(workingday = as\_factor(as.character(workingday))) %>% mutate(workingday = fct\_recode(workingday, "NotWorkingday" = "0", "Workingday" = "1"))  
  
#Convert "weathersit" to a factor and recode the levels. Level 1 should be "NoPrecip", 2 should become "Misty", 3 should become "LightPrecip", and 4 should become "HeavyPrecip".   
  
bike = bike %>% mutate(weathersit = as\_factor(as.character(weathersit))) %>% mutate(weathersit = fct\_recode(weathersit, "NoPrecip" = "1", "Misty" = "2", "LightPrecip" = "3", "HeavyPrecip" = "4"))  
  
#Convert the "weekday" variable to a factor and recode the levels. Note that 6 is "Saturday" and 0 is "Sunday". The rest of the days of the week are from 1 to 5, starting with "Monday".  
  
bike = bike %>% mutate(weekday = as\_factor(as.character(weekday))) %>% mutate(weekday = fct\_recode(weekday, "Monday" = "1", "Tuesday" = "2", "Wednesday" = "3", "Thursday" = "4", "Friday" = "5", "Saturday" = "6", "Sunday" = "0"))

## Task 1

# Split the data into training and testing sets. Your training set should have 70% of the data. Use a random number (set.seed) of 1234. Hint: Remember to specify the response variable when using the createDataPartition function.

#Split the data (training and testing)   
train.rows = createDataPartition(y = bike$count, p=0.7, list = FALSE) #70% in training  
train = bike[train.rows,]   
test = bike[-train.rows,]

## Task 2

How many rows of data are in each set (training and testing)? traning=12167 and test=5212 #

## Task 3: Build a linear regression model (using the training set) to predict “count” using the variables “season”, “mnth”, “hr”, “holiday”, and “weekday”, “temp”, and “weathersit”. Comment on the quality of the model. Be sure to note the Adjusted R-squared value.

Good model. The negative values make sense. Adjusted R-squared: 0.626

ctrl = trainControl(method = "cv",number = 10) #set up caret 10 fold cross validation  
  
set.seed(1234) #set random number seed   
mod1 = train(count ~ season + mnth + hr+ holiday + weekday + temp + weathersit, train, method = "lm", trControl = ctrl, metric="Rsquared")  
summary(mod1)

##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -417.79 -62.25 -9.57 52.32 511.40   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -81.54659 6.91459 -11.793 < 2e-16 \*\*\*  
## seasonSummer 39.83302 6.36213 6.261 3.96e-10 \*\*\*  
## seasonFall 26.57658 7.51795 3.535 0.000409 \*\*\*  
## seasonWinter 57.74388 6.37050 9.064 < 2e-16 \*\*\*  
## mnth2 -0.05773 5.09951 -0.011 0.990968   
## mnth3 1.20926 5.66526 0.213 0.830978   
## mnth4 -13.62135 8.52255 -1.598 0.110009   
## mnth5 -14.13829 9.07623 -1.558 0.119324   
## mnth6 -19.88242 9.32445 -2.132 0.033003 \*   
## mnth7 -44.86379 10.47159 -4.284 1.85e-05 \*\*\*  
## mnth8 -22.64566 10.19776 -2.221 0.026393 \*   
## mnth9 1.01020 9.03128 0.112 0.910939   
## mnth10 0.43080 8.40359 0.051 0.959116   
## mnth11 -13.07681 8.10999 -1.612 0.106894   
## mnth12 -12.24110 6.47253 -1.891 0.058617 .   
## hr1 -15.61469 7.00016 -2.231 0.025725 \*   
## hr2 -25.37874 6.96690 -3.643 0.000271 \*\*\*  
## hr3 -35.80016 6.99444 -5.118 3.13e-07 \*\*\*  
## hr4 -39.88317 7.04855 -5.658 1.56e-08 \*\*\*  
## hr5 -24.77114 7.02303 -3.527 0.000422 \*\*\*  
## hr6 31.85734 7.08471 4.497 6.97e-06 \*\*\*  
## hr7 167.85416 6.98365 24.035 < 2e-16 \*\*\*  
## hr8 315.22318 7.03872 44.784 < 2e-16 \*\*\*  
## hr9 168.57896 6.95414 24.242 < 2e-16 \*\*\*  
## hr10 111.65469 6.99212 15.969 < 2e-16 \*\*\*  
## hr11 138.36094 6.95458 19.895 < 2e-16 \*\*\*  
## hr12 179.69456 7.01227 25.626 < 2e-16 \*\*\*  
## hr13 176.26614 6.96819 25.296 < 2e-16 \*\*\*  
## hr14 163.78712 7.00281 23.389 < 2e-16 \*\*\*  
## hr15 174.44085 7.05401 24.729 < 2e-16 \*\*\*  
## hr16 230.18551 7.05601 32.623 < 2e-16 \*\*\*  
## hr17 389.20670 7.06191 55.113 < 2e-16 \*\*\*  
## hr18 341.56312 6.98754 48.882 < 2e-16 \*\*\*  
## hr19 243.24377 7.06572 34.426 < 2e-16 \*\*\*  
## hr20 158.73496 6.95724 22.816 < 2e-16 \*\*\*  
## hr21 112.23447 7.02912 15.967 < 2e-16 \*\*\*  
## hr22 73.03043 6.93556 10.530 < 2e-16 \*\*\*  
## hr23 31.46595 6.93451 4.538 5.74e-06 \*\*\*  
## holidayHoliday -22.42645 6.46884 -3.467 0.000528 \*\*\*  
## weekdaySunday -14.53968 3.74758 -3.880 0.000105 \*\*\*  
## weekdayMonday -8.91737 3.86206 -2.309 0.020962 \*   
## weekdayTuesday -8.49977 3.76078 -2.260 0.023832 \*   
## weekdayWednesday -5.81754 3.77448 -1.541 0.123275   
## weekdayThursday -3.76919 3.78463 -0.996 0.319309   
## weekdayFriday 0.27680 3.75509 0.074 0.941240   
## temp 286.96367 12.12547 23.666 < 2e-16 \*\*\*  
## weathersitMisty -22.08086 2.35026 -9.395 < 2e-16 \*\*\*  
## weathersitLightPrecip -91.28259 3.79118 -24.078 < 2e-16 \*\*\*  
## weathersitHeavyPrecip -44.98351 78.58787 -0.572 0.567063   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 110.9 on 12118 degrees of freedom  
## Multiple R-squared: 0.6237, Adjusted R-squared: 0.6222   
## F-statistic: 418.5 on 48 and 12118 DF, p-value: < 2.2e-16

## Task 4: Use the predict functions to make predictions (using your model from Task 3) on the training set. Use the “head” function to display the ???rst six predictions. Hint: Be sure to store the predictions in an object, perhaps named “predict\_train” or similar. Comment on the predictions

# The first 5 predictions of count are negative.

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

## 1 2 3 4 5 6   
## -34.02927 -43.79333 -48.47547 -52.55848 -59.52731 13.44275

## Task 5: Task 5: Use the predict functions to make predictions (using your model from Task 3) on the testing set. Use the “head” function to display the ???rst six predictions. Hint: Be sure to store the predictions in an object, perhaps named “predict\_test” or similar. Comment on the predictions.

# Only the first prediction of count is negative.

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

## 1 2 3 4 5 6   
## -12.67531 302.54787 178.86075 204.64198 247.08280 411.84327

## Task 6: Task 6: Manually calculate the R squared value on the testing set. Comment on how this value compares to the model’s performance on the training set.

# This R squared is slightly lower that the R squred in the training set. The model we built on the training set is likely to perform similarly on new data that it has never seen before so we should feel comfortable deploying this into a real world setting with new data.

#manually calculate the R squared value.   
SSE = sum((test$count - predict\_test)^2) #sum of squared residuals from model  
SST = sum((test$count - mean(test$count))^2) #sum of squared residuals from a "naive" model  
1 - SSE/SST #definition of R squared

## [1] 0.6235819

```