## Model Validation

### Anthony Sumter

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

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

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 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(MASS)

##   
## Attaching package: 'MASS'

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

library(caret)

## Warning: package 'caret' was built under R version 3.6.2

## 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\_double(),  
## dteday = col\_date(format = ""),  
## season = col\_double(),  
## yr = col\_double(),  
## mnth = col\_double(),  
## hr = col\_double(),  
## holiday = col\_double(),  
## weekday = col\_double(),  
## workingday = col\_double(),  
## weathersit = col\_double(),  
## temp = col\_double(),  
## atemp = col\_double(),  
## hum = col\_double(),  
## windspeed = col\_double(),  
## casual = col\_double(),  
## registered = col\_double(),  
## count = col\_double()  
## )

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

#### 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.**

bike = bike %>% drop\_na()  
str(bike)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 17379 obs. of 17 variables:  
## $ instant : num 1 2 3 4 5 6 7 8 9 10 ...  
## $ dteday : Date, format: "2011-01-01" "2011-01-01" ...  
## $ season : Factor w/ 4 levels "Spring","Summer",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ yr : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ mnth : Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ hr : Factor w/ 24 levels "0","1","2","3",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ holiday : Factor w/ 2 levels "NotHoliday","Holiday": 1 1 1 1 1 1 1 1 1 1 ...  
## $ weekday : Factor w/ 7 levels "Saturday","Sunday",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ workingday: Factor w/ 2 levels "NotWorkingDay",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ weathersit: Factor w/ 4 levels "NoPrecip","Misty",..: 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 : num 3 8 5 3 0 0 2 1 1 8 ...  
## $ registered: num 13 32 27 10 1 1 0 2 7 6 ...  
## $ count : num 16 40 32 13 1 1 2 3 8 14 ...

set.seed(1234)  
train.rows1 = createDataPartition(y = bike$count, p=0.7, list = FALSE)  
train1 = bike[train.rows1,]   
test1 = bike[-train.rows1,]

#### Task 2

**How many rows of data are in each set (training and testing)?** In the training set there are 12,167 rows of data versus the testing set which has 5,212 rows of data.

#### 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.**

The model quality is pretty good based on the r squared value being .62 and the p value being less than .05.

mod1 = lm(count ~ season + mnth + hr + holiday + weekday + temp + weathersit, train1)   
summary(mod1)

##   
## Call:  
## lm(formula = count ~ season + mnth + hr + holiday + weekday +   
## temp + weathersit, data = train1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -419.31 -61.93 -9.98 52.57 504.24   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -81.2946 6.9356 -11.721 < 2e-16 \*\*\*  
## seasonSummer 28.8486 6.4074 4.502 6.78e-06 \*\*\*  
## seasonFall 19.7865 7.6029 2.602 0.009266 \*\*   
## seasonWinter 62.0339 6.4333 9.643 < 2e-16 \*\*\*  
## mnth2 -0.8013 5.1396 -0.156 0.876114   
## mnth3 2.5584 5.7973 0.441 0.659003   
## mnth4 -1.2250 8.6334 -0.142 0.887166   
## mnth5 -1.5879 9.2279 -0.172 0.863382   
## mnth6 -15.3992 9.4846 -1.624 0.104485   
## mnth7 -38.8277 10.6085 -3.660 0.000253 \*\*\*  
## mnth8 -16.8557 10.3542 -1.628 0.103569   
## mnth9 5.4060 9.2152 0.587 0.557459   
## mnth10 -2.7341 8.5079 -0.321 0.747943   
## mnth11 -12.8043 8.2169 -1.558 0.119193   
## mnth12 -15.3615 6.5409 -2.349 0.018864 \*   
## hr1 -19.7855 6.9722 -2.838 0.004550 \*\*   
## hr2 -28.2440 6.9696 -4.052 5.10e-05 \*\*\*  
## hr3 -40.3146 7.0910 -5.685 1.34e-08 \*\*\*  
## hr4 -40.5469 7.0249 -5.772 8.03e-09 \*\*\*  
## hr5 -26.7454 6.9592 -3.843 0.000122 \*\*\*  
## hr6 32.8518 7.0435 4.664 3.13e-06 \*\*\*  
## hr7 161.3872 6.9925 23.080 < 2e-16 \*\*\*  
## hr8 312.2263 6.9502 44.923 < 2e-16 \*\*\*  
## hr9 164.2556 7.0163 23.411 < 2e-16 \*\*\*  
## hr10 107.1856 6.9552 15.411 < 2e-16 \*\*\*  
## hr11 139.6256 7.0057 19.930 < 2e-16 \*\*\*  
## hr12 179.7448 6.9778 25.760 < 2e-16 \*\*\*  
## hr13 178.6812 7.0201 25.453 < 2e-16 \*\*\*  
## hr14 156.2811 7.0628 22.127 < 2e-16 \*\*\*  
## hr15 168.7543 7.0939 23.788 < 2e-16 \*\*\*  
## hr16 228.1106 7.0881 32.182 < 2e-16 \*\*\*  
## hr17 377.6085 7.0185 53.802 < 2e-16 \*\*\*  
## hr18 347.7287 6.9806 49.813 < 2e-16 \*\*\*  
## hr19 238.7339 7.0128 34.043 < 2e-16 \*\*\*  
## hr20 159.7394 7.0231 22.745 < 2e-16 \*\*\*  
## hr21 108.1070 6.9494 15.556 < 2e-16 \*\*\*  
## hr22 72.3808 6.9874 10.359 < 2e-16 \*\*\*  
## hr23 32.5734 6.9996 4.654 3.30e-06 \*\*\*  
## holidayHoliday -29.0249 6.4088 -4.529 5.98e-06 \*\*\*  
## weekdaySunday -14.0349 3.7638 -3.729 0.000193 \*\*\*  
## weekdayMonday -6.5302 3.8944 -1.677 0.093604 .   
## weekdayTuesday -7.2790 3.8319 -1.900 0.057509 .   
## weekdayWednesday -3.2707 3.7984 -0.861 0.389212   
## weekdayThursday -1.7267 3.8053 -0.454 0.650004   
## weekdayFriday 1.3251 3.7744 0.351 0.725539   
## temp 288.1743 12.1860 23.648 < 2e-16 \*\*\*  
## weathersitMisty -19.6696 2.3717 -8.293 < 2e-16 \*\*\*  
## weathersitLightPrecip -94.1331 3.8166 -24.664 < 2e-16 \*\*\*  
## weathersitHeavyPrecip -80.2490 64.7672 -1.239 0.215356   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 111.9 on 12118 degrees of freedom  
## Multiple R-squared: 0.6217, Adjusted R-squared: 0.6202   
## F-statistic: 414.8 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 first six predictions. Hint: Be sure to store the predictions in an object, perhaps named “predict\_train” or similar. Comment on the predictions.**

Based upon the predictions it appears the training set using the identified variables would produce a negative count of bike rides per day more times than not.

predict\_train = predict(mod1, newdata = train1, interval = "predict")  
head(predict\_train,6)

## fit lwr upr  
## 1 -37.68169 -257.3449 181.9815  
## 2 -46.14026 -265.8041 173.5236  
## 3 -52.44730 -272.1354 167.2408  
## 4 -52.67962 -272.3518 166.9925  
## 5 -58.54772 -278.2314 161.1359  
## 6 14.95557 -204.7171 234.6282

#### 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 first six predictions. Hint: Be sure to store the predictions in an object, perhaps named “predict\_test” or similar. Comment on the predictions.**

Based upon the predictions it appears the testing set using the identified variables would produce a positive count of bike rides per day more times than not.

predict\_test = predict(mod1, newdata = test1, interval = "predict")  
head(predict\_test,6)

## fit lwr upr  
## 1 -12.13272 -231.79110 207.5257  
## 2 137.72755 -81.94175 357.3968  
## 3 174.04493 -45.67523 393.7651  
## 4 17.56108 -202.19963 237.3218  
## 5 -22.20993 -241.96034 197.5405  
## 6 168.48847 -51.19453 388.1715

#### 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.**

The manual R sqaured value compared to the model’s performance on the training set is significantly less and this would indicate that it will less than likely perform well on a new set of data.

SSE = sum((test1$count - predict\_test)^2)  
SST = sum((test1$count - mean(test1$count))^2)   
1 - SSE/SST

## [1] -3.058344

#### Task 7

**Describe how k-fold cross-validation differs from model validation via a training/testing split.**

K-fold cross- validation differs from a training/testing split because it generally results in a less biased result and allows you to do the validation process over and over again to see how model performace may differ across various partitions.