Assignment1

Jordan Whitaker

5/25/2020

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.6.3

## -- Attaching packages ------------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.5  
## v tidyr 1.0.3 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.6.3

## Warning: package 'tidyr' was built under R version 3.6.3

## Warning: package 'readr' was built under R version 3.6.3

## Warning: package 'dplyr' was built under R version 3.6.3

## Warning: package 'stringr' was built under R version 3.6.3

## -- 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 3.6.3

##   
## Attaching package: 'MASS'

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

library(caret)

## Warning: package 'caret' was built under R version 3.6.3

## 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"))  
  
#head(bike) #key reference to ensure code snippets below were successful   
  
bike = bike %>% mutate(yr = as\_factor(as.character(yr))) %>%  
mutate(yr = fct\_recode(yr))  
  
  
bike = bike %>% mutate(mnth = as\_factor(as.character(mnth))) %>%  
mutate(mnth = fct\_recode(mnth))  
  
bike = bike %>% mutate(hr = as\_factor(as.character(hr))) %>%  
mutate(hr = fct\_recode(hr))  
  
  
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,  
"Monday" = "1",  
"Tuesday" = "2",  
"Wednesday" = "3",  
"Thursday" = "4",  
"Friday" = "5",  
"Saturday" = "6",  
"Sunday"= "0"))  
  
head(bike)

## # A tibble: 6 x 17  
## instant dteday season yr mnth hr holiday weekday workingday  
## <dbl> <date> <fct> <fct> <fct> <fct> <dbl> <fct> <fct>   
## 1 1 2011-01-01 Spring 0 1 0 0 Saturd~ NotWorkin~  
## 2 2 2011-01-01 Spring 0 1 1 0 Saturd~ NotWorkin~  
## 3 3 2011-01-01 Spring 0 1 2 0 Saturd~ NotWorkin~  
## 4 4 2011-01-01 Spring 0 1 3 0 Saturd~ NotWorkin~  
## 5 5 2011-01-01 Spring 0 1 4 0 Saturd~ NotWorkin~  
## 6 6 2011-01-01 Spring 0 1 5 0 Saturd~ NotWorkin~  
## # ... with 8 more variables: weathersit <fct>, temp <dbl>, atemp <dbl>,  
## # hum <dbl>, windspeed <dbl>, casual <dbl>, registered <dbl>, count <dbl>

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

Task 2: There are 12167 rows in the training dataset and there are 5212 rows in the testing dataset.

train\_mod = lm(count ~ season + mnth + hr + weekday + temp + weathersit, bike)  
summary(train\_mod)

##   
## Call:  
## lm(formula = count ~ season + mnth + hr + weekday + temp + weathersit,   
## data = bike)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -425.28 -62.13 -9.62 51.87 503.60   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -85.325 5.811 -14.682 < 2e-16 \*\*\*  
## seasonSummer 35.886 5.309 6.759 1.44e-11 \*\*\*  
## seasonFall 27.688 6.291 4.401 1.08e-05 \*\*\*  
## seasonWinter 67.275 5.315 12.657 < 2e-16 \*\*\*  
## mnth2 1.784 4.289 0.416 0.677538   
## mnth3 6.517 4.812 1.354 0.175644   
## mnth4 -5.413 7.156 -0.756 0.449390   
## mnth5 -4.911 7.650 -0.642 0.520923   
## mnth6 -16.246 7.846 -2.071 0.038414 \*   
## mnth7 -40.475 8.817 -4.591 4.45e-06 \*\*\*  
## mnth8 -19.528 8.571 -2.278 0.022716 \*   
## mnth9 4.475 7.627 0.587 0.557435   
## mnth10 -5.087 7.080 -0.719 0.472436   
## mnth11 -20.659 6.810 -3.033 0.002422 \*\*   
## mnth12 -15.939 5.412 -2.945 0.003233 \*\*   
## hr1 -17.940 5.853 -3.065 0.002180 \*\*   
## hr2 -26.972 5.872 -4.593 4.40e-06 \*\*\*  
## hr3 -37.823 5.913 -6.397 1.63e-10 \*\*\*  
## hr4 -41.176 5.916 -6.961 3.51e-12 \*\*\*  
## hr5 -24.977 5.877 -4.250 2.15e-05 \*\*\*  
## hr6 33.391 5.862 5.697 1.24e-08 \*\*\*  
## hr7 169.353 5.854 28.929 < 2e-16 \*\*\*  
## hr8 310.664 5.849 53.117 < 2e-16 \*\*\*  
## hr9 164.659 5.849 28.152 < 2e-16 \*\*\*  
## hr10 111.714 5.857 19.074 < 2e-16 \*\*\*  
## hr11 139.231 5.874 23.702 < 2e-16 \*\*\*  
## hr12 180.293 5.893 30.596 < 2e-16 \*\*\*  
## hr13 176.232 5.911 29.816 < 2e-16 \*\*\*  
## hr14 160.570 5.928 27.088 < 2e-16 \*\*\*  
## hr15 170.041 5.934 28.654 < 2e-16 \*\*\*  
## hr16 231.584 5.928 39.064 < 2e-16 \*\*\*  
## hr17 384.699 5.911 65.079 < 2e-16 \*\*\*  
## hr18 352.104 5.895 59.725 < 2e-16 \*\*\*  
## hr19 241.675 5.874 41.144 < 2e-16 \*\*\*  
## hr20 161.218 5.861 27.505 < 2e-16 \*\*\*  
## hr21 110.407 5.852 18.867 < 2e-16 \*\*\*  
## hr22 72.420 5.847 12.386 < 2e-16 \*\*\*  
## hr23 33.245 5.845 5.688 1.31e-08 \*\*\*  
## weekdaySunday -15.871 3.150 -5.039 4.73e-07 \*\*\*  
## weekdayMonday -11.528 3.159 -3.649 0.000264 \*\*\*  
## weekdayTuesday -6.766 3.173 -2.132 0.033012 \*   
## weekdayWednesday -4.057 3.167 -1.281 0.200223   
## weekdayThursday -2.888 3.165 -0.912 0.361561   
## weekdayFriday 1.134 3.154 0.359 0.719313   
## temp 285.462 10.213 27.951 < 2e-16 \*\*\*  
## weathersitMisty -19.253 1.982 -9.714 < 2e-16 \*\*\*  
## weathersitLightPrecip -90.443 3.170 -28.534 < 2e-16 \*\*\*  
## weathersitHeavyPrecip -77.016 64.449 -1.195 0.232106   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 111.4 on 17331 degrees of freedom  
## Multiple R-squared: 0.6237, Adjusted R-squared: 0.6227   
## F-statistic: 611.2 on 47 and 17331 DF, p-value: < 2.2e-16

Task 3: The model above is pretty good with an adjusted R-Squared value of .62, most variables are significant.

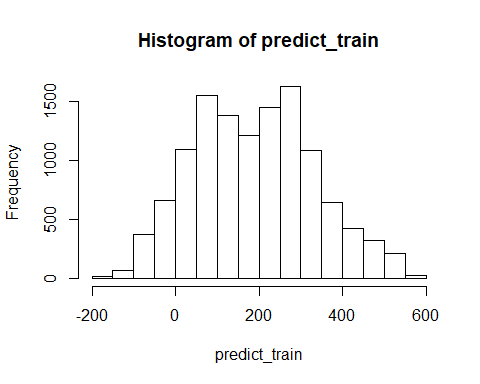
predict\_train = predict(train\_mod, newdata = train)  
  
head(predict\_train)

## 1 2 3 4 5 6   
## -40.46297 -49.49554 -54.63665 -57.98999 -61.04450 10.86837

View(predict\_train)  
summary(predict\_train)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -187.1 74.9 189.3 189.4 289.1 591.7

hist(predict\_train)



The model has some interesting predictions. There are roughly 1000 combined instances where there is a negative count prediction. The distribution is mostly normal.

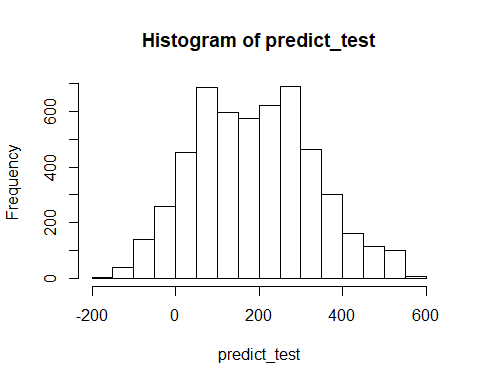
predict\_test = predict(train\_mod, newdata = test)  
  
head(predict\_test)

## 1 2 3 4 5 6   
## -16.81384 141.12050 170.82455 10.86317 -27.52776 162.60997

View(predict\_test)  
summary(predict\_test)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -184.96 78.79 188.06 189.54 288.28 573.04

hist(predict\_test)



Task 5: The testing data has a similar distrubtion compared to the training data predictions. There are still some negative predictions but the overall trend of the distrubtion is closely related.

SSE = sum((test$count - predict\_test)^2) #sum of squared errors  
  
SST = sum((test$count - mean(test$count))^2) #sum of squared residuals from a "naive" model  
  
r\_squared = 1 - SSE/SST  
r\_squared # .6309

## [1] 0.6309314

Task 6: Manually calculating R-Squared value The manually calculated R-squared value on the testing data is .6309 compared to .6227 on the training data model. These closely related r-squared values point to good fitting model. It is slightly above, so there may be some overfitting, but not a tremendous amount.

Task 7: K-fold cross validation is a technique that splits the data into partitions, typically k is either 3, 5, or 10. Using these partitions, the model will evaluate k times and upon each iteration will hold out 1 fold. This fold will rotate between partitions until each partition has been evaluated as a part of the training set. Compared to the training/testing split technique which just takes a majority of the data at a random seed and designates it as training data and holds out a minority percentage that will be used to evaluate the training data model.