## Week3 Assignment 1

### DEY, Sankha

#### Model Validation

# message=FALSE done before final knit  
library(tidyverse)  
library(MASS)  
library(caret)

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()  
## )

#str(bike) # Commeneted before final knit

#Convert Season  
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  
bike = bike %>% mutate(yr = as\_factor(yr))  
bike = bike %>% mutate(mnth = as\_factor(mnth))  
bike = bike %>% mutate(hr = as\_factor(hr))  
  
#convert holiday  
bike = bike %>% mutate(holiday = as\_factor(as.character(holiday))) %>%  
mutate(holiday = fct\_recode(holiday,  
"NotHoliday" = "0",  
"Holiday" = "1"))  
  
#convert workingday  
bike = bike %>% mutate(workingday = as\_factor(as.character(workingday))) %>%  
mutate(workingday = fct\_recode(workingday,  
"NotWorkingDay" = "0",  
"WorkingDay" = "1"))  
  
#convert weathersit  
bike = bike %>% mutate(weathersit = as\_factor(as.character(weathersit))) %>%  
mutate(weathersit = fct\_recode(weathersit,  
"NoPrecip" = "1",  
"Misty" = "2",  
"LightPrecip" = "3",  
"HeavyPrecip" = "4"))  
  
#convert weekday  
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"))  
str(bike)

## Classes 'spec\_tbl\_df', '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 ...

#### Task 1

set.seed(1234) #set random number seed for cross validation  
trainrows1 = createDataPartition(y = bike$count, p=0.7, list = FALSE) #70% in training  
train1 = slice(bike,trainrows1)  
test1 = slice(bike,-trainrows1)

#### Task 2

Number of rows in training data set = 12167  
Number of rows in testing data set = 5212

#### Task 3

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

The model has high adjusted R-squared value (0.6202). Overall quality of the model is good with a good R-squared value and all variables (some dummy variables in cases) have p-values less than 0.05.  
This model also matches with our intuition for most of the variables except season and mnth. The slope of SeasonWinter is greater than the same of SeasonSummer or SeasonFall. This is kind of odd. Similarly slope of mnth7 became negative which ideally shouldn’t be.

#### Task 4

predict\_train = predict(mod1, newdata = train1)

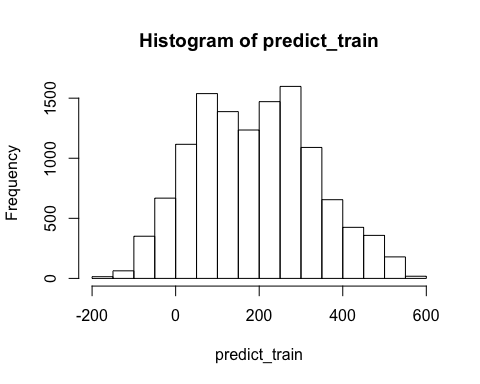
head(predict\_train)

## 1 2 3 4 5 6   
## -37.68169 -46.14026 -52.44730 -52.67962 -58.54772 14.95557

summary(predict\_train)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -183.90 75.04 188.25 189.33 289.19 584.44

hist(predict\_train)

 At a glance, Predictions seem reasonable with median and mean are very close and the histogram is equally distributed. However, a good number of count is predicted to be negative. That doesn’t see reasonable.

#### Task 5

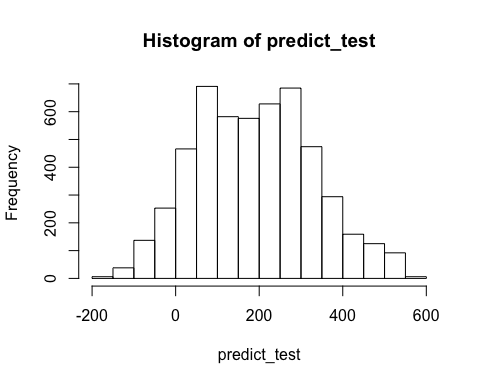
predict\_test = predict(mod1, newdata = test1)  
head(predict\_test)

## 1 2 3 4 5 6   
## -12.13272 137.72755 174.04493 17.56108 -22.20993 168.48847

summary(predict\_test)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -187.00 78.66 188.34 189.41 288.72 567.39

hist(predict\_test)

 Similar observation with test data. At a glance, Predictions seem reasonable with median and mean are very close and the histogram is equally distributed. However, a good number of count is predicted to be negative. That doesn’t see reasonable.

#### Task 6

SSE = sum((test1$count - predict\_test)^2) #sum of squared residuals from model  
SST = sum((test1$count - mean(test1$count))^2) #sum of squared residuals from a "naive" model  
1 - SSE/SST #definition of R squared

## [1] 0.6289223

R-Squared value at testing dataset is 0.629 which is very close to the same of training data set. The model built on training dataset is likely to perform well on new data at testing dataset. So, we should feel comofrtable on deploying this model to real world new data.

#### task 7

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

##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -424.63 -62.16 -9.71 51.92 499.33   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -85.278 5.808 -14.684 < 2e-16 \*\*\*  
## seasonSummer 35.545 5.306 6.699 2.17e-11 \*\*\*  
## seasonFall 26.998 6.289 4.293 1.77e-05 \*\*\*  
## seasonWinter 65.129 5.330 12.219 < 2e-16 \*\*\*  
## mnth2 1.323 4.287 0.309 0.75768   
## mnth3 5.078 4.818 1.054 0.29187   
## mnth4 -6.014 7.152 -0.841 0.40041   
## mnth5 -5.832 7.647 -0.763 0.44566   
## mnth6 -18.097 7.850 -2.305 0.02116 \*   
## mnth7 -41.455 8.813 -4.704 2.58e-06 \*\*\*  
## mnth8 -21.251 8.572 -2.479 0.01318 \*   
## mnth9 4.316 7.622 0.566 0.57119   
## mnth10 -3.922 7.079 -0.554 0.57954   
## mnth11 -18.304 6.823 -2.683 0.00731 \*\*   
## mnth12 -15.180 5.411 -2.805 0.00503 \*\*   
## hr1 -17.912 5.849 -3.062 0.00220 \*\*   
## hr2 -26.901 5.868 -4.584 4.59e-06 \*\*\*  
## hr3 -37.809 5.909 -6.399 1.61e-10 \*\*\*  
## hr4 -41.087 5.912 -6.950 3.78e-12 \*\*\*  
## hr5 -24.888 5.873 -4.238 2.27e-05 \*\*\*  
## hr6 33.488 5.858 5.717 1.10e-08 \*\*\*  
## hr7 169.440 5.850 28.963 < 2e-16 \*\*\*  
## hr8 310.710 5.845 53.160 < 2e-16 \*\*\*  
## hr9 164.653 5.845 28.170 < 2e-16 \*\*\*  
## hr10 111.648 5.853 19.075 < 2e-16 \*\*\*  
## hr11 139.110 5.870 23.697 < 2e-16 \*\*\*  
## hr12 180.131 5.889 30.588 < 2e-16 \*\*\*  
## hr13 176.032 5.907 29.801 < 2e-16 \*\*\*  
## hr14 160.344 5.924 27.067 < 2e-16 \*\*\*  
## hr15 169.807 5.931 28.632 < 2e-16 \*\*\*  
## hr16 231.354 5.925 39.050 < 2e-16 \*\*\*  
## hr17 384.495 5.907 65.086 < 2e-16 \*\*\*  
## hr18 351.933 5.892 59.735 < 2e-16 \*\*\*  
## hr19 241.539 5.870 41.147 < 2e-16 \*\*\*  
## hr20 161.120 5.858 27.506 < 2e-16 \*\*\*  
## hr21 110.339 5.848 18.868 < 2e-16 \*\*\*  
## hr22 72.378 5.843 12.387 < 2e-16 \*\*\*  
## hr23 33.232 5.841 5.689 1.30e-08 \*\*\*  
## holidayHoliday -26.140 5.335 -4.899 9.71e-07 \*\*\*  
## weekdaySunday -15.873 3.148 -5.043 4.64e-07 \*\*\*  
## weekdayMonday -7.779 3.248 -2.395 0.01663 \*   
## weekdayTuesday -6.528 3.172 -2.058 0.03960 \*   
## weekdayWednesday -3.805 3.166 -1.202 0.22940   
## weekdayThursday -2.393 3.165 -0.756 0.44960   
## weekdayFriday 1.631 3.154 0.517 0.60515   
## temp 287.864 10.218 28.173 < 2e-16 \*\*\*  
## weathersitMisty -19.377 1.981 -9.782 < 2e-16 \*\*\*  
## weathersitLightPrecip -90.772 3.168 -28.650 < 2e-16 \*\*\*  
## weathersitHeavyPrecip -78.721 64.407 -1.222 0.22163   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 111.3 on 17330 degrees of freedom  
## Multiple R-squared: 0.6242, Adjusted R-squared: 0.6232   
## F-statistic: 599.8 on 48 and 17330 DF, p-value: < 2.2e-16

Ran a k-fold (10 fold) cross-validation on bike dataset with count is the response variable. The R-Squared value is 0.6232 which is very closed with the value we got after train/test model.  
In Train/Test model, dataset is splt with training (70%-80%) and testing (20%-30%). The model runs on training set and the predicted on new test data. This model is good for very large data set.  
In k-fold, dataset is randomly split up into ‘k’ groups, typically (k=3,5, or 10). One of the k groups is used as the test set and the rest are used as the training set. Then the process is repeated until each unique group as been used as the test set.Then the best model is kept as the final one. This model is good for large dataset.