Model Validation

Deliverable: All of your work for this assignment should be done in an R Markdown document. Knit your document into a Word file and submit the Word file as the deliverable for this assignment.

Libraries: For this assignment you will need the following libraries: tidyverse, MASS, caret.

Before beginning the assignment tasks, read-in the "hour.csv" file into a data frame called "bike". This is the same data that you used in the Module 2 Multiple Linear Regression and Special Issues assignment. Using the same code you created in that assignment:

Convert "season" so that 1 = spring, 2 = summer, 3 = fall, 4 = winter.
Convert "yr", "mnth", and "hr" to factors. You do NOT need to recode (rename) the levels of these factors.
Convert the "holiday" variable to a factor and recode the levels from 0 to "NotHoliday" and 1 to "Holiday".
Convert "workingday" to a factor and recode the levels from 0 to "NotWorkingDay" and 1 to "WorkingDay".
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".
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".

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.**

Task 2: How many rows of data are in each set (training and testing)?

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.

Task 4: Use the predict functions to make predictions (using your model from Task 3) on the **training** set. **Hint: Be sure to store the predictions in an object, perhaps named "predict_train" or similar.** It can be useful to "sanity check" your predictions. You can do this is one or more of several ways: 1) Use the "head" function to display the first six predictions corresponding to the first six rows in the data. 2) Examine a summary of the predictions. Are there any strange predictions? 3) Examine a histogram of predictions. Does the distribution of predictions seem reasonable? Comment on the predictions.

Task 5: Use the predict functions to make predictions (using your model from Task 3) on the **testing** set. **Hint: Be sure to store the predictions in an object, perhaps named "predict_test" or similar.** As you did in Task 4, comment on the predictions.

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.

Task 7: Describe how k-fold cross-validation differs from model validation via a training/testing split.