**BACS - HW (Week 16)**

Let’s return yet again to the cars dataset we now understand quite well. Recall that it had several interesting issues such as non-linearity and multicollinearity. How do these issues affect prediction?  
  
Let’s **setup** all the models we need for this assignment using:

|  |
| --- |
| *# Load the data and remove missing values* cars <- read.table("auto-data.txt", header=FALSE, na.strings = "?") names(cars) <- c("mpg", "cylinders", "displacement", "horsepower", "weight", "acceleration",   "model\_year", "origin", "car\_name") cars$car\_name <- NULL cars <- na.omit(cars)  *# Shuffle the rows of cars* set.seed(27935752) cars <- cars[sample(1:nrow(cars)),]  *# Create a log transformed dataset also* cars\_log <- with(cars, data.frame(log(mpg), log(cylinders), log(displacement), log(horsepower), log(weight), log(acceleration), model\_year, origin))  *# Linear model of mpg over all the variables that don’t have multicollinearity* cars\_lm <- lm(mpg ~ weight + acceleration + model\_year + factor(origin), data=cars)  *# Linear model of log mpg over all the log variables that don’t have multicollinearity* cars\_log\_lm <- lm(log.mpg. ~ log.weight. + log.acceleration. + model\_year + factor(origin),   data=cars\_log)  *# Linear model of log mpg over all the log variables, including multicollinear terms!* cars\_log\_full\_lm <- lm(log.mpg. ~ log.cylinders. + log.displacement. + log.horsepower. +   log.weight. + log.acceleration. + model\_year + factor(origin),  data=cars\_log) |

**Question 1)** Let’s work with the cars\_log model and test some basic prediction. Split the data into train and test sets (70:30) and try to predict log.mpg. for the smaller test set:

1. Retrain the cars\_log\_lm model on just the training dataset (call the new model: lm\_trained);  
   Show the coefficients of the trained model
2. Use the lm\_trained model to predict the log.mpg. of the test dataset  
   What is the in-sample mean-square fitting error (MSEIS) of the trained model?  
   What is the out-of-sample mean-square prediction error (MSEOOS) of the test dataset?
3. Show a data frame of the test set’s actual log.mpg., the predicted values, and the difference of the two (predictive error); *Just show us the first several rows*

*(see next page for Question 2)*

**Question 2)** Let’s see how our three large models described in the setup at the top perform predictively!

1. Report the MSEIS of the cars\_lm, cars\_log\_lm, and cars\_log\_full\_lm; Which model has the best (lowest) mean-square fitting error? Which has the worst?
2. Try writing a function that performs k-fold cross-validation (see class notes and ask in Teams for hints!). Name your function k\_fold\_mse(dataset, k=10, …) – it should return the MSEOOS of the operation. Your function may must accept a dataset and number of folds (k) but can also have whatever other parameters you wish.
   1. Use/modify your k-fold cross-validation function to find and report the MSEOOS for cars\_lm – recall that this non-transformed data/model has non-linearities
   2. Use/modify your k-fold cross-validation function to find and report the MSEOOS for cars\_log\_lm – does it predict better than cars\_lm? Was non-linearity harming predictions?
   3. Use/modify your k-fold cross-validation function to find and report the MSEOOS for cars\_log\_lm\_full – this model has collinear terms; so does multicollinearity seem to harm the predictions?
3. Check if your k\_fold\_mse function can do as many folds as there are rows in the data (i.e., k=392). Report the MSEOOS for the cars\_log\_lm model with k=392.

We will take a deeper dive into predictions and machine learning in our next (and final) class.