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:

a. Retrain the `cars_log_lm` model on just the training dataset (call the new model: `lm_trained`); Show the coefficients of the trained model
b. Use the `lm_trained` model to predict the `log.mpg.` of the test dataset
   What is the in-sample mean-square fitting error (MSE_{IS}) of the trained model?
   What is the out-of-sample mean-square prediction error (MSE_{OOS}) of the test dataset?
c. 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!

a. Report the MSE_{IS} 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?

b. 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 MSE_{OOS} of the operation. Your function may must accept a dataset and number of folds (`k`) but can also have whatever other parameters you wish.

   i. Use/modify your k-fold cross-validation function to find and report the MSE_{OOS} for `cars_lm` – recall that this non-transformed data/model has non-linearities

   ii. Use/modify your k-fold cross-validation function to find and report the MSE_{OOS} for `cars_log_lm` – does it predict better than `cars_lm`? Was non-linearity harming predictions?

   iii. Use/modify your k-fold cross-validation function to find and report the MSE_{OOS} for `cars_log_lm_full` – this model has collinear terms; so does multicollinearity seem to harm the predictions?

c. 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 MSE_{OOS} 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.