Homework 1: Linear Regression and k-Nearest Neighbors Regression

STAT 665

due Fri. Jan. 27th 5PM

Background

Citi Bike is a public bicycle sharing system in New York City. There are hundreds of bike stations scattered throughout the city, with more stations continually added over time. Customers can check out a bike at any station and and return it at any station (where of course, the start and end stations could be different). Citi Bike caters to both commuters (who use the bike to get to and from work) as well as tourists (who use the bike for sight-seeing in the city). Details on this program can be found at:

https://www.citibikenyc.com/

For this assignment, you will build models for predicting Citi Bike usage (in # of trips per day) for the six months April through September, 2016.

The Data

The dataset consists of two parts: Citi Bike usage information as well as weather data (as recorded from Central Park). Intuitively, poor weather might negatively impact tourist usage but probably won't change commuter usage.

In the `citibike_*.csv` files, we see:

1. `date`
2. `trips`: the total number of Citi Bike trips (not in test set)
3. `n_stations`: the total number of Citi Bike stations in service (estimated using the total number of stations used on the given date)
4. `holiday`: whether or not the day is a work holiday

In the `weather.csv` file, we have:

1. `date`
2. `PRCP`: amount precipitation (i.e. rainfall amount) in inches
3. `SNWD`: snow depth in inches
4. `SNOW`: snowfall in inches
5. `TMAX`: maximum temperature for the day, in degrees F
6. `TMIN`: minimum temperature for the day, in degrees F
7. `AWND`: average windspeed

You are provided a training set consisting of data from 7/1/2013 to 3/31/2016 and a test set consisting of data after 4/1/2016. The weather file contains weather data for both periods, combined. You will need to predict `trips` for the test set, so obviously you will not have the `trips` column in the test data frame!

Your Task

Part 1 (20%)

For this part, you will write a function that performs k-nearest neighbors regression. In short, k-nearest neighbors regression estimates y (associated with vector of covariates x) using:

$$\hat{y} = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

where $N_k(x)$ is the set of the k-nearest points to x in the training set. We will use Euclidean distance to determine $N_k(x)$. In other words, the distance between vectors x_i and x_l is defined by:

$$d(x_i, x_l) = \sqrt{\sum_{j=1}^{n} (\tilde{x}_{ij} - x_{lj})^2}$$

Note: If you struggle with this part of the assignment, skip to Part 2.

Write a function that performs k-nearest neighbor regression. The function should take in:

- a set of predictors in the training set `xtrain` (a matrix)
- a set of predictors in the test set `xtest` (a matrix)
- a vector of responses in the training set `ytrain`, and
- `k` the number of nearest neighbors considered

The function should then output a vector of predicted responses for the test set. Should you choose to undertake this in R, a skeleton of the function has already been prepared for you, currently returning an appropriately-sized vector of NAs (missing values).

```
myknn <- function(xtrain, xtest, ytrain, k) {

  return(rep(NA, nrow(xtest)))
}
```

Helpful Notes:

1. You can safely assume that `xtrain` and `xtest` are matrix-like objects with the same columns, that is, you won't be attempting k-nearest neighbor regression with a single predictor.
2. For a single row in `xtest` (say $\tilde{x}_i$), you should compute its Euclidean distance to each row in `xtrain` $(x_1, \ldots, x_n)$. This should yield a total of n distances:

$$d(\tilde{x}_i, x_1), \ldots, d(\tilde{x}_i, x_n)$$

Then take an average of the `k` values in `ytrain` that correspond to the `k` smallest $d(\tilde{x}_i, x_l)$, $l = 1, \ldots, n$. This is your predicted value for $\tilde{y}_i$.

Part 2 (80%)

Now, let's return to the Citi Bike data and our goal of modeling the number of daily trips. You will consider 2 possible ways of predicting the number of trips.

1. k-nearest neighbor regression
2. linear regression

(Note: If you could not get your implementation of k-nearest neighbors working in Part 1, you may use one from a built-in package. For example, if you are using R, you can use the `knn.reg()` function in the "FNN" R package.)

First, you'll need to merge the weather data and the Citi Bike data, since it's possible that weather data can be helpful for predicting bike usage.

You should randomly split the training set into your own training set and a validation set in order to do model selection for both of these approaches. Use Mean Squared Error (MSE) on the validation set as the performance metric (see Section 2.2 in textbook for details).

In the first approach (k-NN), you'll want to see which value of `k` works best, i.e. you should compute the validation MSE for a range of values for k. In the second approach, you will want to determine which variables (with possible transformations or newly-derived features) to include in the least squares model; again, you should utilize validation MSE to compare models linear regression models.

Once you have identified an optimal k-nearest neighbors model and an optimal linear regression model, compare the two optimal approaches to each other – which one does a better job of predicting `trips` in your validation set? Using what you deemed to be the better approach, refit the model now on the entire training set and generate predictions for the test set.

Note: Although you should not use other outside data (besides the included weather data) for this analysis, you are welcome (and encouraged) to transform and create new predictors from the existing data.

What to Submit

You must submit the output of your compiled .Rmd or .ipynb file in either Word or PDF format via the Assignments section on Canvas **plus** a .csv file related to Part 2. (Or, if you're using Python, then submit a .ipynb notebook compiled into a PDF format). The original .Rmd or .ipynb scripts do not need to be submitted. It is permissible to compile to either HTML or Word format first, and then save the file as a PDF in your browser or MS Word.

In the spirit of reproducible analysis, your submitted Word or PDF file should show the full code used to complete Parts 1 and 2, including code that imports the data. The submitted document should be easy-to-read, contain few grammatic and typographic errors, and be nicely-formatted.

Part 1 (20%)

Include code for your R or Python function `myknn()`. You should not use any outside packages for this part.

Part 2 (80%)

Describe your best k-nearest neighbor model and best linear regression model. That is, you should state what the predictors are in each case, and in the case of k-nearest neighbors, state the value of k.

Then, attach a .csv file containing your predicted daily trip counts for the test set. This file should have 2 columns (`date`, `trips`). Call this file `HW1_netid.csv`, obviously with your actual Yale netid in place of 'netid'.

Academic Integrity (for This Assignment and Beyond)

While you are welcome to discuss parts of the homework with your fellow classmates, you should write up your solutions separately. Please include the names of your collaborators at the top of your assignment.