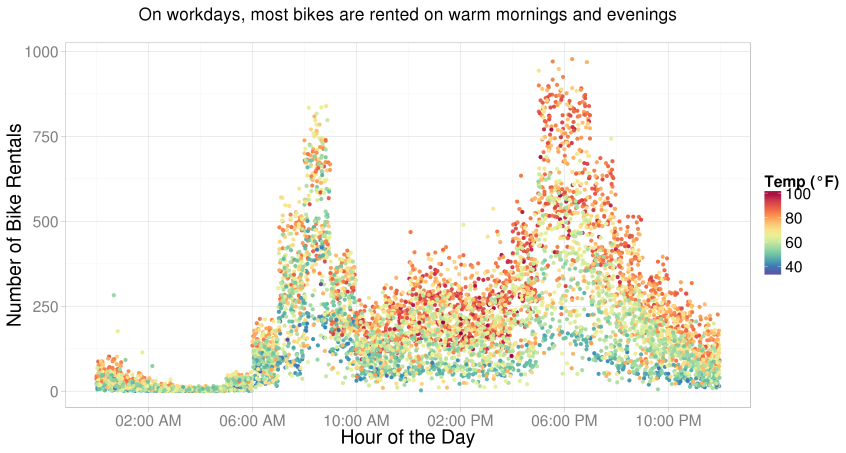
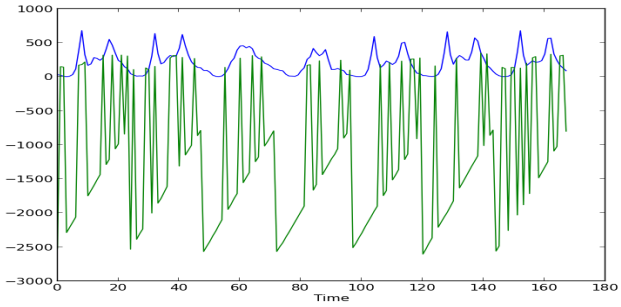
Benjamin Chen
a take-home interview exercise from DataRobot for Data Science intern
http://www.kaggle.com/c/bike-sharing-demand

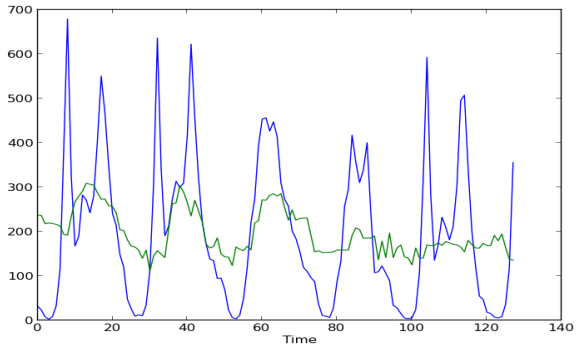


The data is split for analysis into a training set of the first 90% of the data, instead of randomly because the application is forecasting. Conforming to Google Prediction API input guidelines, the columns that are not the output nor in the training set (number of non-registered user rentals initiated and number of registered user rentals initiated) are removed. The blue is our dataset and green is the predicted point. 0 is Thursday midnight.

Fitted Week 1 of Model using Date and Hour as separate variables



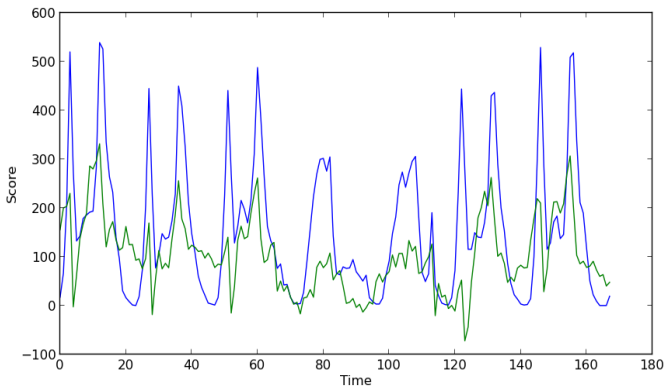
Fitted Week 1 of Model using a unified Time variable



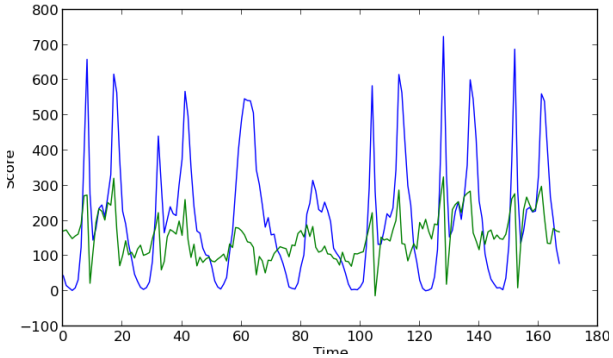
After going through the Google Prediction API, the results go into results.csv, the results of a representative few are as shown above. Using date and hour as separate numeric variables, Google does not learn that those two variables are correlated. Clearly, with the zigzagging pattern, it is using a linear component to hour of the day. Using a unified time variable (ex: 1337.5 being noon 1337 days after 2011-01-01), we see a better model that seems to somewhat understand the cyclic component of the data on Thursday through Saturday, but captures neither the spikes around 9am and 5pm, nor the dips between days.

Taking a different direction, encoding time as the number of rentals from one hour prior and one week (168 hours) prior. We see that it is more representative of the data, but has similar problems when trying to generalize. The weekends in most predictions are particularly poor. The next step would be to place each hour of the week into it's own column.

Fitted Week 35 using a Model Including Last Hour and Last Week



Sample Predictions of Model including Last Hour and Last Week on Test Data



Overall, I am unsatisfied with the results but am unsure Google Prediction API is the right tool for the job. Considering its opaqueness and how much processing needs to occur beforehand, I question how much work it is actually saving, if any. Maybe there are more convenient specialized libraries. Perhaps the real contribution of this API is the scalability and RESTful calls.