Two Sigma Data Challenge Report

Ben Jakubowski

March 1, 2017

1 Introduction

For my project for the Two Sigma data challenge, I chose to use CitiBike trip data, plus external NOAA daily weather data, to build a predictive model of the number of CitiBike trips taken per day from 2014-2016. This modeling task is motivated by the assumption that the CitiBike system needs to regularly take bikes out of circulation for maintenance. If demand can be forecasted accurately, CitiBike may be able to predict dates with low demand and chose those days to take bikes out of circulation for maintenance.

In constructing my models, I made a number of assumptions:

- Bike trips are logged one week after the trip start date. This seems to be a very conservative assumption (and if more information were available on data availability, new lag features could be tested).
- Weather predictions are essentially the same as actual weather conditions. This is a very liberal assumption, but actual weather condition data was available through a NOAA API, and (given time constraints for this project) robust datasets on past weather forecasts were not found. With additional time, the model would ideally be retrained on weather forecasts (with the appropriate time delta to account for this use case).

2 Data

I used two data sources to complete this project.

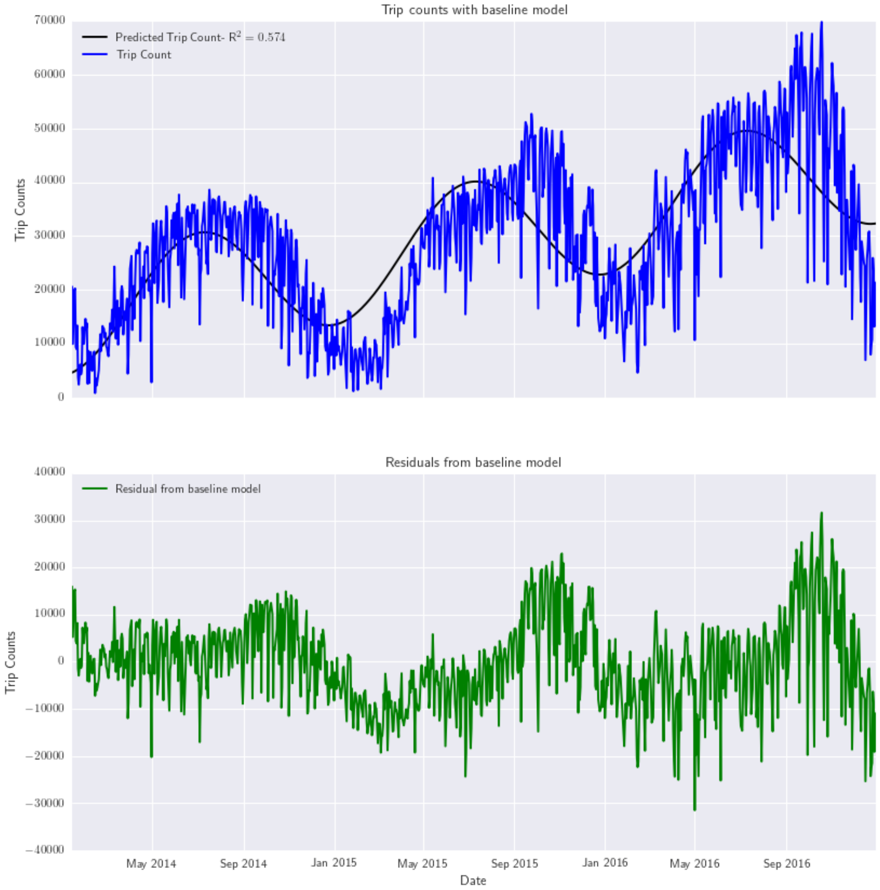
- CitiBike trip data. Available at https://s3.amazonaws.com/tripdata/index.html
  - The only feature computed from this dataset was the number of trips taken on each day from 01/01/2014-12/31/2016.
- NOAA Climate Data Online. The API is documented at https://www.ncdc.noaa.gov/cdo-web/webservices/v2. Features taken from this datasource include:
  - Precipitation
  - Snowfall
  - Snow depth
  - Max temperature
  - Min temperature
  - Average daily wind speed
  - Fastest 2-minute wind speed
  - Fastest 4-second wind speed
  - (Binary): Fog, ice fog, or freezing fog (may include heavy fog)
  - (Binary): Smoke or haze

In addition to raw features, a number of additional features were constructed for each day, including

- A day number feature (where $t_{01/01/2014} = 0,\ t_{01/02/2014} = 1,\ \cdots$). This feature was constructed under the hypothesis that the CitiBike system is relatively new, and probably has been growing (linearly) with time.
- Trig features ($sin\left(\frac{2\pi t}{365.25}\right)$ and $cos\left(\frac{2\pi t}{365.25}\right)$) to capture hypothesized annual cycle in usage. Note these features were constructed with fixed one-year frequencies.
- Average daily trip count over a 7-day window ending a week earlier.
- Day of the week
- Month number (to account for potential misspecification in the fixed seasonal cycle).

3 Modeling

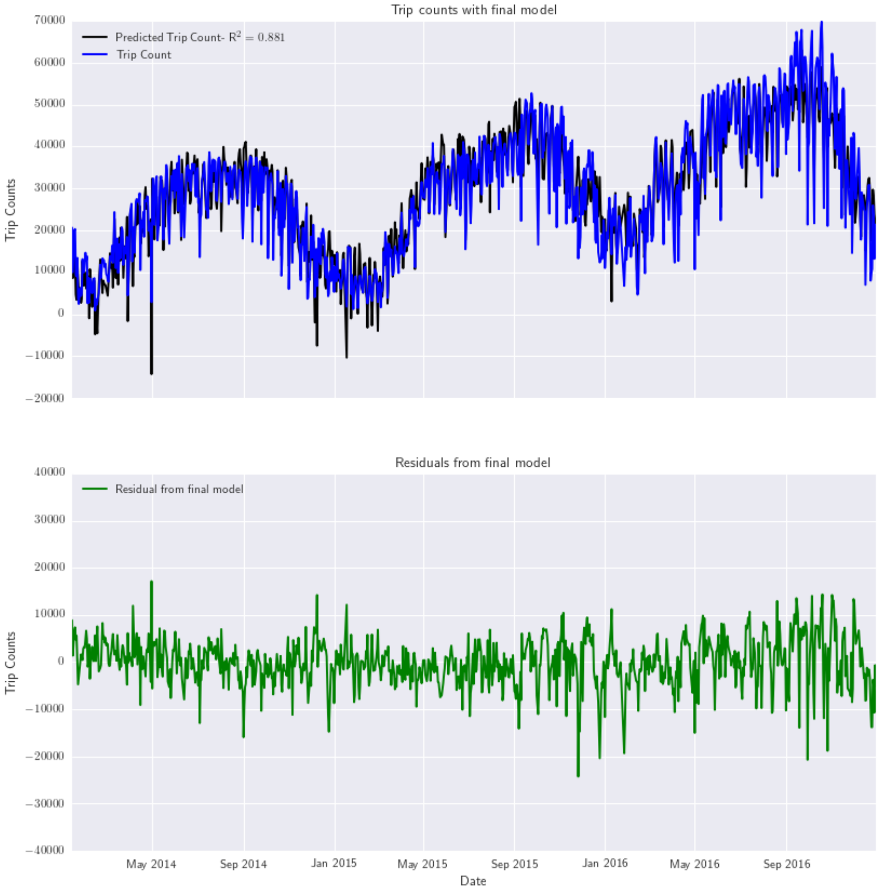
First, we used a 90-10 training test split. Then two models were learned over the training set. The first (our baseline model) was an unregularized linear model including only t, sin, and cos features (plus an intercept). The fit model is plotted below, along with the residual.



Trip counts with baseline model

Residuals from baseline model

Next, 5-fold cross validation was used to optimize ElasticNet hyperparameters for linear model including all the features. The final fit model is plotted below, along with the residual.



4 Results and Discussion

To evaluate the models, we present R^2 values for the training and test sets:

Model	Train	Test
Baseline (Simple Linear)	0.574035414422	0.530425946396
Final (Elastic Net)	0.880643189036	0.825660091537

Based on this training/test performance, we draw a couple of conclusions:

- The decrease in performance (for both models) from training to testing suggests overfitting. However, given the small sample size, it may also be random noise (due to the training/test split). With additional time, this could be explored further (for example by looking at the standard error in the estimated out-of-sample R^2 from cross-validation).

- Regardless, it is apparent that the final model is a dramatic improvement over the simple baseline in predicting the number of CitiBike rides taken on a given day.