Estimating Citibike Demand to Predict Up-and-Coming Neighborhoods

Vittorio Bisin (vb704@nyu.edu) Raghav Singhal (rs4070@nyu.edu)
Pranjal Thapar (pt1089@nyu.edu) Gita Ventyana (gita.ventyana@nyu.edu)

May 20, 2017

Abstract

Bike sharing programs have been gaining popularity throughout the world. But a major cost for most systems is the re-allocations of bikes on a daily basis, due to the naturally asymmetric flow of traffic. To that effect we attempt to build a demand function for each station which can predict the daily demand for each bike station in the network. Furthermore we also intend to investigate the relationship between real estate prices and changes in Citibike station demand - to predict up and coming neighborhoods. As of mid-May 2017 we have modeled the bike station demand function using a generalized linear regression, with a test set average error of 20%.

1 Introduction

Our project is divided into two parts:

1) build a model to predict the station demand (i.e. number of bikes left that station) starting at any of the 603 stations
2) understand if changes in Citibike demand are causality related to fluctuations in rent prices for that neighborhood.

Each part of the problem could provide services to Citibank and real estate companies. Firstly our demand function could be used by Citibank to optimize their bike station network - minimizing the number of unused bikes by setting demand equal to supply, thereby increasing customer utility. Secondly (and perhaps more interestingly) if we determine that there is causation (or even just a correlation) between changes in real estate prices and changes in bike station demand, then this will be useful information for a real estate company on deciding whether to invest in a particular neighborhood or not. Furthermore, this information could also be used by Citibike to determine which neighborhoods to install new bike stations - we hypothesize that our model will suggest that neighborhoods with higher rent prices will have a higher demand for Citibikes.

The first part of our project, building the demand model for the Citibike station network, is more technically complicated - though it is also the more researched problem in the literature. This problem is an interdisciplinary one - spanning fields of data science, mathematics, computer science, and economics. The literature can generally be divided into two approaches: the more data science/mathematics and the more optimization/economics related (as explained in this paper, we opted for a combination of the two).

1.1 Literature Review

Initially we had thought of approaching this problem from a mathematical - signal processing perspective, mainly following the work of [HBFR13]. The idea in their paper is to treat the Citibike stations as a changing (dynamic) graph (where each node is a station and vertex a route between two stations), then using a frequency analysis of the graph to treat the graph as a group of non-stationary signals. We found this approach to be very mathematically interesting, but less

practically useful that it becomes harder to apply multiple features (e.g. weather) in this model.

Another direction commonly found in the bike sharing literature is a more economic-optimization approach (see [CJ15], [LLQ+15], [OS15]). Although the authors in these types of papers (e.g. see [LSCX16]) do create a demand function that estimates future bike consumption, their objective is to equalize demand and supply - to minimize the number of unused bikes and maximize user satisfaction (it is appropriately called in the literature the "Bike Sharing Problem"). On the other hand, we simply wanted to design a demand function to then be able to predict demand and change in demand to then apply this to our real estate data, and predict up-and-coming neighborhoods. An interesting application of the Bike Sharing Problem is [BGK] where they use a biologically inspired ant colony optimization algorithm technique to solve the bike re-balancing problem. Despite [BGK] unique approach to the re-balancing problem, generally neural nets and regressions (e.g. linear and weighted K-nearest neighbor) seem to be the algorithms of choice.

2 Methods

2.1 Data Preparation and Feature Extraction

We firstly worked with the large dataset (more than 35 million trips) taken directly from Citibike's website. This dataset contained the following features: trip duration, start time and date, end time and date, bike ID, user type,gender, and year of birth; however, for our project we only needed the first four features, leaving out user demographics. The dataset was already quite well-organized and involved only fixing small errors, such as bike trips that lasted over 24 hours (presumably these trips were either data input errors or the bikes were stolen). Finally from the above four Citibike features we calculated the explicit features used in our model: week number, weekend,weekday, bike demand between all pairs of stations, and mean/median trip duration.

Our second data set was downloaded from the National Oceanic and Atmospheric Administration (NOAA) and consisted of 47,335 hourly surface weather observations in Central Park, each with 1,009 features. Unlike the above described Citibike dataset, it was not well-organized - containing a large amount of incomplete and inconsistent observations. Adding to this there were also numerous discrepancies between the data and its explanatory text file. Firstly to clean this data we removed the indicator variables, which were not relevant for our research. We then fixed the high number of incomplete observations by using a mean binning method, in many such cases the feature had to be removed because the number of incomplete observations was too high. Furthermore, we then applied two feature reduction algorithms to further reduce the number of features: high correlation threshold (FCT) and Singular Value Decomposition (SVD). Our feature reduction results provided only two possible intuitively relevant features: temperature and rain.

Our third data set was downloaded from Zillow, an on-line real estate database. Like the Citibike dataset, the observations are rarely incomplete or inconsistent and required limited cleaning. The dataset consisted of monthly median values (per square feet) of homes in 160 New York neighborhoods, with a particular emphasis on neighborhoods in Brooklyn. Of course, since we are trying to estimate up-and-coming neighborhoods, Zillow's emphasis on neighborhoods in Brooklyn will become quite useful. Furthermore, on top of monthly median values of homes, the Zillow dataset also includes multiple other variables of home prices: list prices, foreclosures, and estimated rent prices.

2.2 General Linear Model

To build the model for demand prediction, we decided upon implementing a Generalized linear Regression technique. The reason for this choice was that we can choose a target distribution for the response variable (demand per station), which we assume to be from the exponential family of distributions (Poisson, Gamma, Binomial,etc). Particularly for our case, this was a vital choice due to the particular nature of our problem, count data, double peaked as evening and morning traffic due to office-goers was particularly high. Hence, unlike a simple regression model which assumes a Gaussian distribution for the response variable Y_i, we could access several other distributions which suited our data.

Another interesting feature of General Linear Models is that it links the response variable Y_i to the linear model via a link function which is also related to the distribution which we choose to model the data on. This leads to a particular format for the distribution function, which is defined as follow, $\forall x \in \mathbf{R}^n$:

$$f_Y(y|\theta,\tau) = h(y,\tau)\exp(\frac{b(\theta)^T y - A(\theta)}{d(\tau)})$$

where the function $h(\theta), d(\theta), A(\theta)$ are known functions, which are related to the distribution of our choosing, and τ is called the dispersion parameter and is related to the variance of the distribution, and finally the vital parameter is θ which is related to the mean of the distribution given by the link function which we choose. We intended to the Gaussian Distribution and the Poisson Distribution for our model, the link function and other functions are detailed below. The θ parameter is defined as follows:

$$\mu = \mathbf{E}(Y) = \nabla A(\theta)$$

and the variance of the distribution is as follows:

$$\mathbf{Var}(Y) = \nabla\nabla^T A(\theta) d(\tau)$$

Due to the GLM model letting us specify the link function $g(\mu_i)$, which relates the expected value μ to the so called linear predictor η_i:

$$g(\mu_i) = \eta_i = x_i^T \beta$$

where β is the regression parameter. Often the link function is chosen such that $A^{'} = g^{-1}$. In which case

$$\theta_i = A^{'}(\mu_i) = g(g^{-1}(\eta_i)) = \eta_i$$

Then the GLM model finds the regression parameter β to maximize th Likelihood function:

$$\max_{\beta} \mathbf{L}(\theta|y,X) = \prod_{i=1}^{N} h(y_i,\tau)\exp(\frac{b(\theta_i)^T y_i - A(\theta_i)}{d(\tau)})$$

where the parameter θ_i is related to the regression parameter β as :

$$\theta_i = A^{'-1}(g^{-1}(x_i^T \beta))$$

Now the functions in the Poisson distribution formula are as follows:

$$h(x) = \frac{1}{x!}$$

$$\eta = \log\lambda$$

$$A(\theta) = \lambda$$

$$\theta = \lambda$$

The second model we tried was the Gaussian Distribution , the functions in which are as follows:

$$\theta = \mu$$

$$h(x) = \frac{\exp(\frac{-x^2}{2\sigma})}{\sigma\sqrt{2\pi}}$$

$$\eta = \frac{\mu}{\sigma}$$

$$A(\theta) = \frac{\mu^2}{2\sigma^2}$$

Here $h(x)$ is the base measure, which is related to the distribution we choose for our model.

2.3 PageRank

We employed this model in hope of finding upcoming stations for our real-estate prices part, which we were not able to do. We simply arranged the graph $G = (V, E, W)$, where V is the vertex set, E is the edge set and W is the weight matrix, as follows:

1. V is the set of all bike stations
2. If there is a bike trip between 2 stations, then there exits an edge between the two vertex.
3. W_{ij} is the number of trips from station i to station j

So almost like the actual deployment of the PageRank algorithm where each vertex is a hyper-link, two vertex are connected if there is a link from one page to another, and the weight matrix is the probability of clicking the link from one page to another, our model also solves a similar problem with hyper-links replaced by Stations, the rest is the same. The resulting PageRank vector r satisfies the following Eigenvalue Problem

$$\mathbf{P}r = r$$

where $\mathbf{P}$ is the transition matrix for a random walk on this graph and r is thus an eigenvector with an eigenvalue of 1, hence it is a stationary point for the Transition Matrix.

In our case, the resulting rank vector r can be interpreted in a similar way as the search procedure for Google, that is the stations are ranked in accordance with their demand.

2.4 Clustering

We also employed K-means clustering to cluster pairs of stations based on the demand between them. To build these models we first extracted three features for each pair of stations. For every station pair S_i, S_j, we have the weekday (1 if weekday, 0 otherwise), week-number(1-52) as the first two features and daily demand, which is the number of daily trips between these two stations as the third feature for the clustering.

Clustering is an unsupervised learning problem whereby we aim to group subsets of entities with one another based on some notion of similarity. Spark MLlib uses a parallelized version of the K-means++ clustering method. The algorithm is as follows:

1. Choose a center uniformly randomly from the data points
2. Calculate distance between each data point and the center nearest to it and call it D(x)
3. Choose a new data point randomly as center, with weighted probability distribution proportionate to $D(x)^2$ for an element x
4. Repeat 2 and 3 till K centers are chosen
5. Now proceed with K-means clustering

This seeding method greatly improves the final error rates of K-means.

3 Experimental Results

3.1 Clustering

Our best error rate with K- means comes with the parameters k=5, maxiterations=10 and runs=10. As we can see in figure 1 there are 5 clusters with different average daily demand between the pairs of stations that fall in each of these clusters respectively. Cluster 0 consists of pairs of stations that have high demand between them while Cluster 3 can be thought of as the cluster that has pair of stations with low daily demand.

On exploring these clusters further we find that one of the highest demand is between the Yankee Ferry Station in the Governor's Island and itself. This could be because it only one of the

3 Citibike clusters on the whole island. So many of the trips are made in the summer months from this station to itself. This station falls in Cluster 0.

One of the pair of stations in Cluster 3 with lowest demand is the pair of 11th and W52nd, 3rd and E12th stations. We can why this would be true, as this ride is more than 4 miles long and would have taken 40 minutes on average to complete.

3.2 Demand prediction

We define the variable of interest Y_i as the daily demand per station, that is the number of bikes taken out from one station in one day. We then worked with mainly two feature sets:

1. Week-number(1-52), weekend(1 if weekday , 0 otherwise)
2. Week-number(1-52), weekend(1 if weekday , 0 otherwise), Average Rain, and Average Temperature

Quite surprisingly the first two features yielded strong performance, the mean absolute error was about 20 bikes for arbitrarily selected stations, however by adding the second feature set yielded better performance.

The most crucial feature is the weekend/weekday feature as we saw a drastic drop in the number of bikes taken out during weekends and this drop was further exacerbated by colder weather. As you can notice in the plots in the appendix, there is a periodic drop in demand, both actual and predicted, this drop can be primarily attributed to the weekend/weekday feature.

Figure 2 and Figure 3 are plots of predicted vs actual demand for arbitrarily selected stations. Both were modeled using a Gaussian Distribution in the General Linear Model. We were expecting to use the Poisson Distribution for our model, however due to memory overflow we are unable to do so.

Note that we have not exploited the graphical structure of our problem, as we assumed each station to be a stand-alone unit, free from any effects by other stations, although we did get good performance, it could possibly lead to a substantial improvement in the performance. These kinds of approaches, where you disregard the geometry of the problem, are called mean-field methods. These methods are commonly used in Statistical Mechanics (Ising Model) and in Machine Learning, where any analytic approach quickly becoming unfeasible. The reason we used this approach was to reduce the complexity of the problem, as learning the evolving structure of a graph over time is not only time consuming but also difficult to accomplish given present techniques.

We calculated the Pearson correlation between actual demand and prediction to show which one has the better prediction. The average correlation for each station without using the weather feature is 0.376, while the average correlation using the weather feature is 0.641. This shows that the prediction using weather features is much better.

4 Architecture

For the implementation of these ideas we used the big data technologies provided by HPC at NYU through the DUMBO cluster (Fig 9). All the data, including all 36 million trips from the Citibike dataset and the Weather variables from the Weather dataset were first transferred to the cluster. They were then transfered to HDFS where it was stored in two different Apache HIVE tables. We then accessed the HIVE table using SPARKSQL and also the Spark Core. This was done for building the linear regression models and evaluating them by using the Spark MLLib library. For the Clustering, the Citibike data in HDFS was accessed directly using the Spark Core, using a Pyspark Core and the Cluster.

5 Future Work

As mentioned earlier, we would like to exploit the graphical structure of our problem further to predict daily demand per station. Possibly deploy a Representation Learning Technique which can automatically learn the features it needs to predict demand.

Another set of features that we did not use in our project were Gender and Age, which when combined with the start location and end location of the user's trip, can give us a reasonable approximation of the user's social status and preferences. The idea that we are getting at is essentially building a profile for a typical Citibike User. This can possibly be used as a targeting mechanism by Citibike and other services which have overlapping user bases, such as Uber, Airbnb, etc. So if there is an increase in the neighborhood population of Citibike users, then Citibike could start a station there or if there is already a station there, then Citibike can provide more bikes to that station. This is actually a problem that Citibike already faces, they had a station is Brooklyn with very low traffic but then over a period of time it started getting a lot of traffic, a phenomenon which can most likely be ascribed to demographic changes.

Bike-Sharing programs have been gaining prevalence around the world so another possible suggestion for any future work would be use more data that is available from other bike-sharing programs, especially European and Chinese Bike-Sharing programs, which are on a bigger scale and have been around for much longer. This can then be used to contrast the effect on the environment different programs have had in the cities. Possible effects we could observe would be:

1. Change in Vehicular Pollution Levels
2. Changes in General well-being (Health or mortality rates) of the people using such systems.

Also one could see if an increase in Citibike users also coincides with an increase in users of similar apps, particular restaurants, or even gym membership.

References

[BGK] Cashous W Bortner, Can Gürkan, and Brian Kell. Ant colony optimization applied to the bike sharing problem.

[CJ15] Longbiao Chen and Jérémie Jakubowicz. Inferring bike trip patterns from bike sharing system open data. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 2898–2900. IEEE, 2015.

[CMP+] Longbiao Chen, Xiaojuan Ma, Gang Pan, Jérémie Jakubowicz, et al. Understanding bike trip patterns leveraging bike sharing system open data. *Frontiers of Computer Science*, pages 1–11.

[CR14] Edoardo Croci and Davide Rossi. Optimizing the position of bike sharing stations. the milan case. 2014.

[HBFR13] Ronan Hamon, Pierre Borgnat, Patrick Flandrin, and Céline Robardet. Tracking of a dynamic graph using a signal theory approach: application to the study of a bike sharing system. In *ECCS'13*, page 101, 2013.

[LLQ+15] Junming Liu, Qiao Li, Meng Qu, Weiwei Chen, Jingyuan Yang, Hui Xiong, Hao Zhong, and Yanjie Fu. Station site optimization in bike sharing systems. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 883–888. IEEE, 2015.

[LSCX16] Junming Liu, Leilei Sun, Weiwei Chen, and Hui Xiong. Rebalancing bike sharing systems: A multi-source data smart optimization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1005–1014. ACM, 2016.

[OS15] Eoin O'Mahony and David B Shmoys. Data analysis and optimization for (citi) bike sharing. In *AAAI*, pages 687–694, 2015.

[Vog16] Patrick Vogel. Service network design of bike sharing systems. In *Service Network Design of Bike Sharing Systems*, pages 113–135. Springer, 2016.

[BGK] [CJ15] [CMP+] [CR14] [HBFR13] [LSCX16] [LLQ+15] [OS15] [Vog16]

6 Appendix

Cluster	Daily Demand
0	25.710392969774283
1	5.019148586697327
2	3.1435010605346645
3	2.2340122961265534
4	9.92133366417079

Figure 1: Cluster number and average daily demand

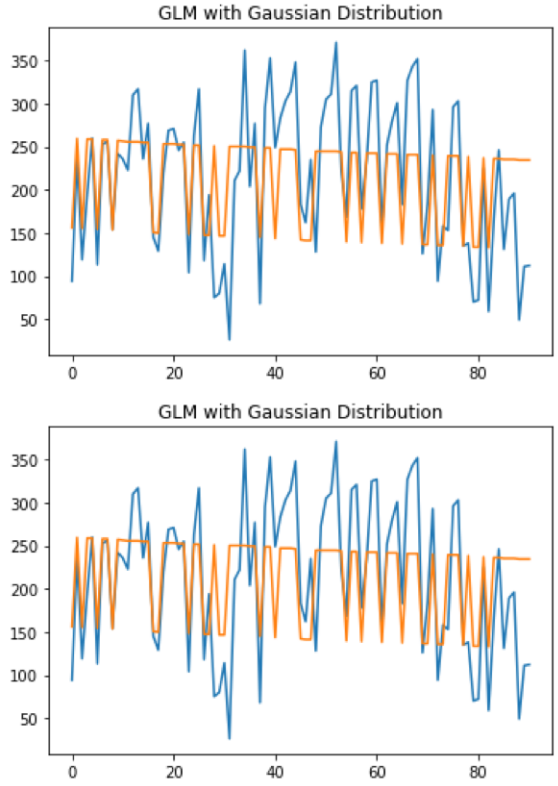

Figure 2: Actual demand (blue) vs. predicted demand (orange) for bike stations 1 and 2 without weather data

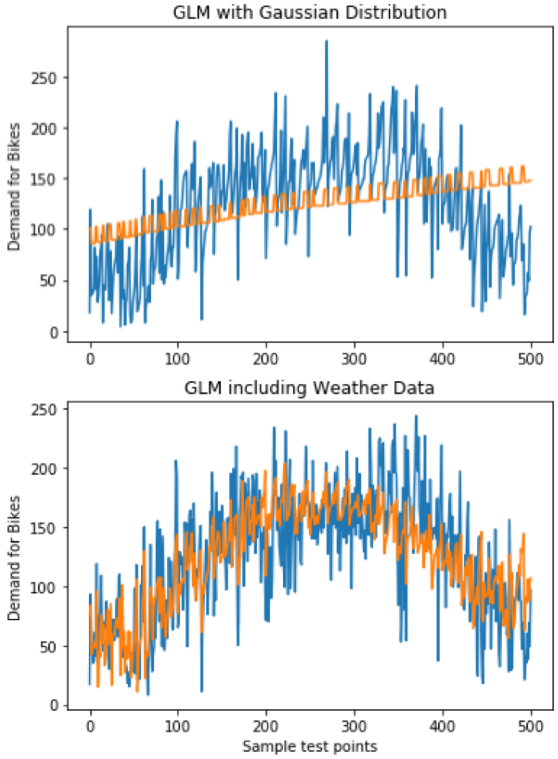

Figure 3: Actual demand (blue) vs. predicted demand (orange) for bike station3 without and with weather data

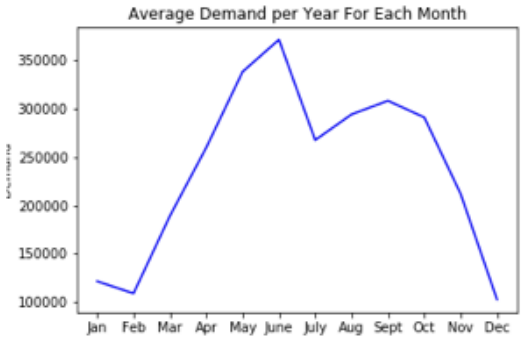

Figure 4: Aggregated average bike demand for per Month

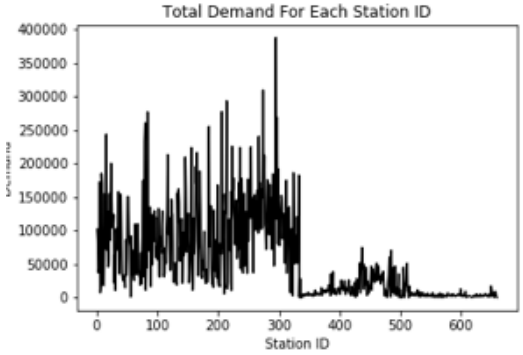

Figure 5: Aggregated bike demand for each station ID. The new bike stations (in up and coming neighborhoods) have higher ID numbers and lower demand than their Manhattan counterparts.

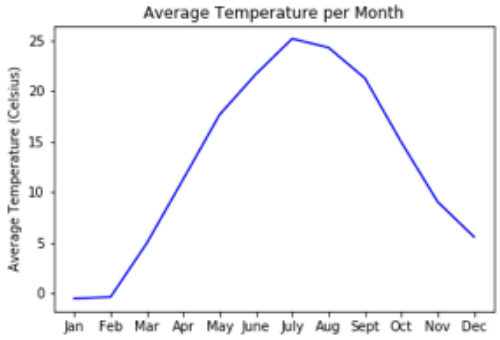

Figure 6: Using the NOAA dataset we calculate the average temperature per month

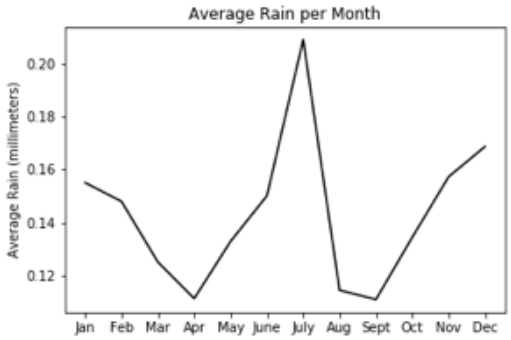

Figure 7: Using the NOAA dataset we calculate the average rain per month

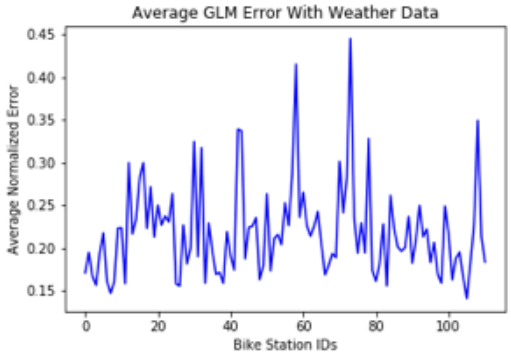

Figure 8: We calculate our GLM model with weather data's average error with weather data for each bike station ID

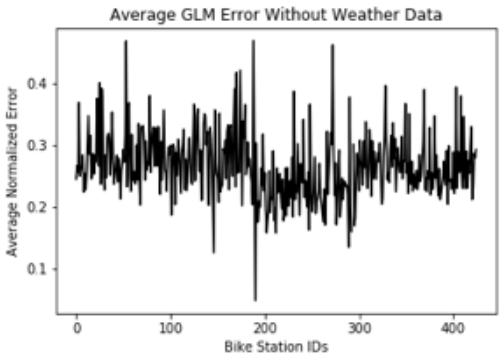

Figure 9: We calculate our GLM model without weather data's average error with weather data for each bike station ID

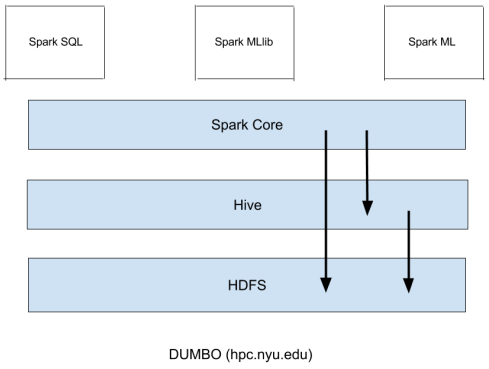

Figure 10: Architecture Diagram for the project