Jiamin Xuan
Applied Data Science
Group member: Danni Wang, Tong Jian, Jiamin Xuan
December 9, 2014

Data analysis on Citibike

Clustering, Markov Matrix, Random Forest

Abstract:

In this project we explored the Citibike system data and did a spatial cluster of the station location and we build two Markov Matrixes and a Random Forest to predict the bike usage every hour. We find that public bike usage is greatly influenced by climate and weather, morning/evening traffic peak and also weekdays. In addition, there are more subscribers than random customers and there are more male subscribers. Also, people use Citibike mostly for short distance trip. People in Brooklyn ride longer maybe the transportation is inconvenient. You can also find the movement from residential area to business areas.

In this team, my job is mainly doing clustering and build random forest. I also help with preprocessing data using Hadoop (8 million records) and some exploratory data analysis.

Overview of Citibike:

Citibike in New York City is a great success. Since the launch, the cumulative trips has increased to nearly 4,000,000 and annual membership has increased to over 80,000. Up to October 2014, there are over 328 stations activated and there are more than 5,000 active bikes on the street on average. Just in October, CitiBike riders took 924,178 trips and traveled 1,496,213 miles, which offset 778,030.76 pounds of carbon.

Despite the success it has in solving the "last mile" problem and promoting green transportation option to reduce the carbon emission and improving health condition. It still has many problems especially in bike redistribution. we think we can do some analysis on the data we acquired on Citibike.

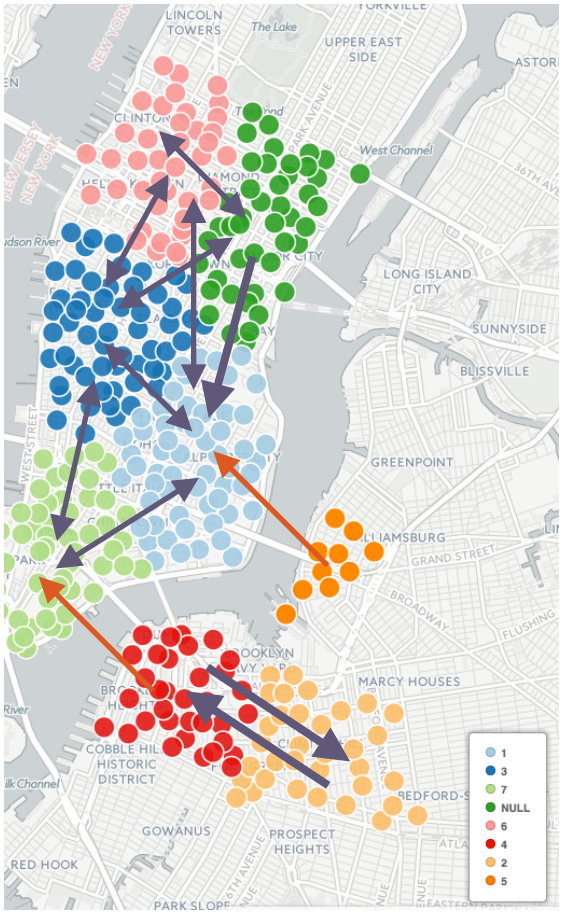
Data acquiring and Preprocessing:

The system data of Citibike can be obtained from *citibikenyc.com*. It is in CSV format and we use python scripts to download and combine the data. We select an entire year, from Sep 2013 to Sep 2014, to export the pattern of citibike usage, e.g., seasonal effects, hour effects, gender percentage.

The entire data set has 8,562,172 records and the processing will be extremely slow. We use Hadoop Streaming to preprocessing the data and get the hour, month, weekday and count(see attached code).

We also get the real time data feeds about station in JSON format, we transform the JSON format and stored in CSV along with clustering result.

Spatial clustering:

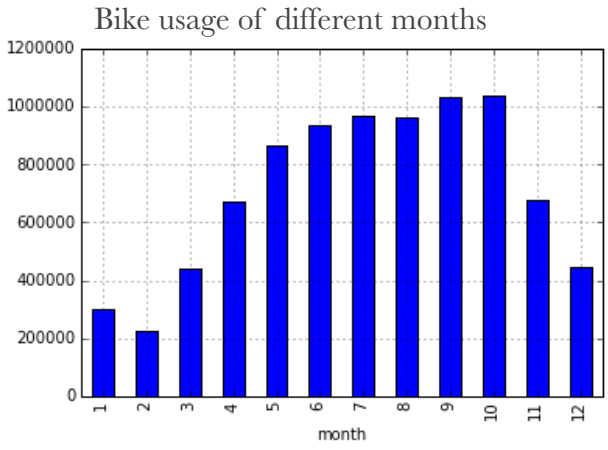


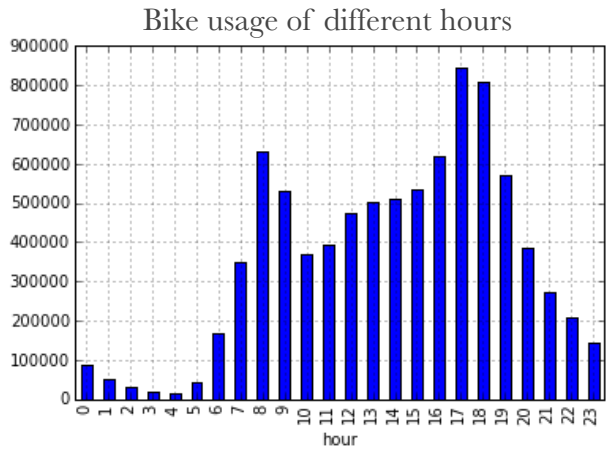
The major challenge for Citibike is the bike redistribution. we consider both long distance and short distance movement and use Markov matrix to indicate the movement. For long distance movement, we can't build a matrix of over 300 stations, so we should cluster the station into several cluster. We tried multiple ways to do this job, such as DBSCAN, K-means, dp-means with different parameters. We tried to get a result more similar to our common sense since people make decisions based on common sense. Each cluster is about 20 blocks so it is usually what 30 minutes people can go. Finally we decide to use K-means with 8 clusters and it returns this result very satisfying. Clearly, you can name the clusters as downtown, lower west, lower west, upper east, upper west, navy yard, downtown Brooklyn and east Brooklyn.

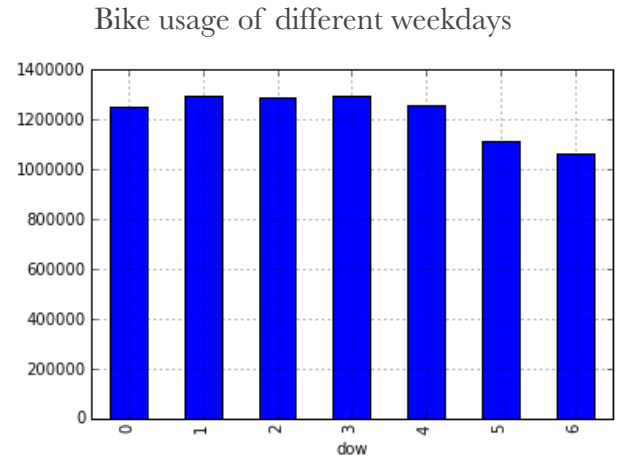
But if you use DBSCAN on entire year records, you will find some connections between different stations. For example, the station near Brooklyn bridge in Brooklyn actually belong downtown Manhattan cluster, it make sense since tourist like ride a bike to across the bridge and view over the bridge.

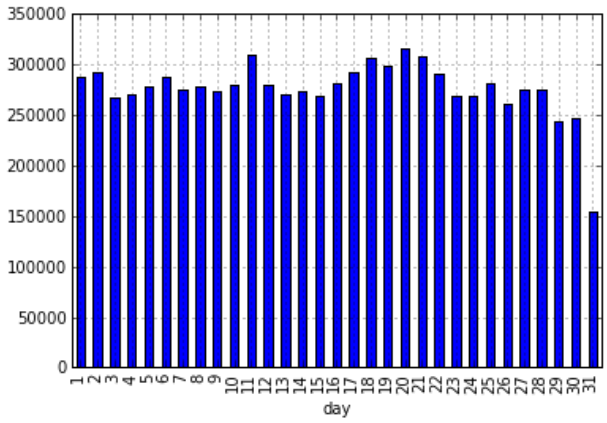
Exploratory data analysis:

After preprocessing the data, we ran some basic slicing, indexing and sorting code to explore the data. To our surprise, we found that almost 90% of the bike users have monthly/annual membership and there are more than 76% of the subscribers are male. I'm interested in the pattern of the bike usage over time:









From the data we can tell that bike use is greatly reduced by cold weather, and we can find mooring and evening peak in every day and slightly drop in weekends. Tong find the age difference and Dannie find the top 10 stations.

Markov Matrix :

We made 2 different Markov matrix on top 10 stations and also clusters.

The naïve Markov Matrix of transition probabilities among the 10 stations is computed as follows:

ID	151	285	293	327	402	426	435	497	519	521
151	19.22%	13.06%	28.73%	3.58%	4.89%	3.69%	6.17%	14.33%	4.27%	2.07%
285	17.48%	17.27%	25.38%	2.07%	5.45%	2.29%	13.51%	3.81%	4.90%	7.84%
293	11.21%	16.42%	12.84%	2.64%	8.54%	3.78%	11.49%	19.86%	5.02%	8.20%
327	5.60%	0.80%	2.96%	45.34%	2.96%	21.12%	3.53%	1.93%	3.67%	12.09%
402	6.98%	10.19%	10.06%	3.11%	12.38%	3.21%	12.28%	20.02%	10.72%	11.05%
426	5.01%	1.09%	3.38%	20.37%	3.92%	52.87%	3.23%	1.27%	1.89%	6.97%
435	2.39%	5.26%	7.23%	4.65%	10.28%	6.76%	15.02%	12.01%	12.90%	23.51%
497	10.70%	4.92%	14.87%	2.30%	8.40%	3.55%	13.00%	16.98%	13.93%	11.35%
519	4.49%	7.86%	4.06%	6.87%	15.64%	4.71%	9.20%	18.66%	19.40%	9.11%
521	2.05%	5.47%	7.25%	8.70%	11.09%	9.87%	13.03%	13.10%	14.36%	15.08%

After taking trip duration, user type, birth year and gender into consideration, the transition probability within Station 426 has slightly dropped to 49.19%, while that within Station 327 has also slightly dropped to 40.67%.

The naïve Markov Matrix of transition probabilities among the 8 clusters is computed as follows:

CLUSTER	0	1	2	3	4	5	6	7
0	37.57%	15.56%	0.16%	17.83%	0.35%	0.26%	23.50%	4.77%
1	11.14%	42.10%	0.64%	24.52%	1.39%	2.00%	4.01%	14.21%
2	1.47%	5.57%	43.91%	1.81%	33.36%	9.35%	0.58%	3.96%
3	9.07%	19.47%	0.20%	45.32%	0.34%	0.38%	13.49%	11.73%
4	1.48%	7.73%	20.04%	2.80%	47.05%	4.52%	0.94%	15.43%
5	2.86%	25.67%	16.31%	6.82%	10.46%	30.21%	1.17%	6.50%
6	24.62%	6.13%	0.09%	25.42%	0.19%	0.14%	37.29%	6.13%
7	3.36%	17.19%	0.64%	19.56%	3.90%	0.54%	5.54%	49.26%

Firstly, the transition on the diagonal is far higher than those in other positions. It means Citibike is mostly used for short term trip. The cluster is wide as about 20 blocks, so a trip inside a cluster is a short distance trip. Taking a subway, taking a cab or driving a car would be at higher cost and lower efficiency, but walking may be a little bit slow.

Secondly, except for adjacent clusters, there are high frequent trips crossing three bridges on East River, especially 25.67% for trips from Lower Manhattan cluster to Brooklyn Downtown cluster via Brooklyn bridge and 20.04% for trips from Lower East to Williamsburg via Williamsburg bridge. It is not only because tourists and exercising people like these classical routes but also for Citi-Bike is a better choice than ferries and subways to cross the river. So the Citi-Bike administration should increase stations and docks on both sides of the bridge in the future.

Lastly, transition probabilities between clusters are more than 20% in Brooklyn, they are higher than that in Manhattan, it may reflect on the transportation condition in Brooklyn provides less convenience than Manhattan, the public transportation network maybe less comprehensive and efficient in Brooklyn.

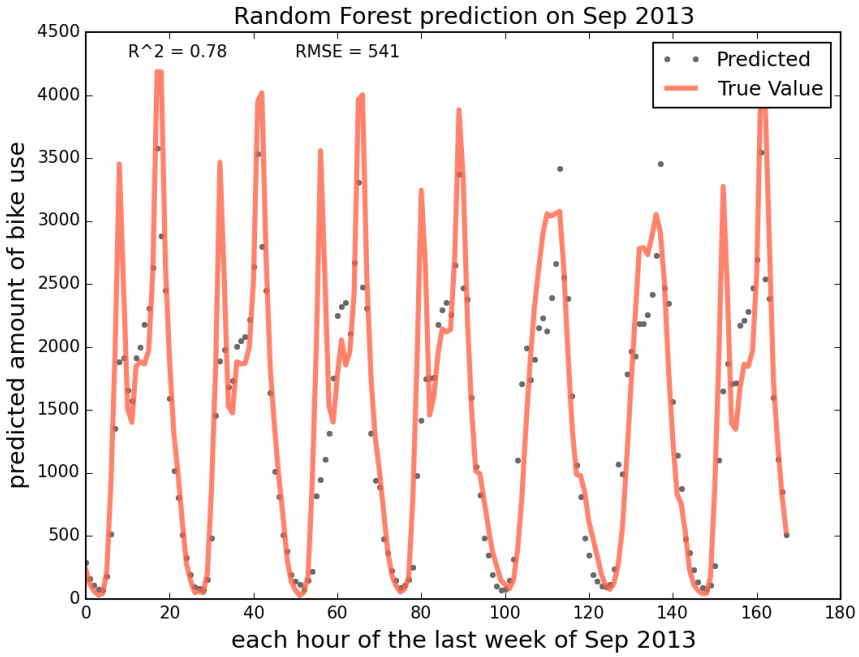
Random Forest:

In order to explore the weather and time effects on bike usage, we also build a random forest predictive model on the data. Since it is difficult to run random forest on Hadoop streaming, and also because of the seasonal effects, the pattern and relationship is different in different months, so I only select one month(Sep 2013) of the data and build the model locally.

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. It is easy to implement with good accuracy.

I use the weather data acquired from open data for each hour and I get the hours, dates to explain the weekday effects and hour peak. The weather data includes temperature, skycover, humidity, precipitation,solar_radiation, together with hours and days, I will predict the total amount of bike usage the the hour. I set the estimator (number of trees in the forest) to 100 and also use 8 CPUs to speed up the processes.

The result turns out to be this:



You can find that the model did tried to catch every pattern it occurs but the root mean square error (RMSE) is as high as 541 and R^2 is 0.78. The result is not as good as expected but with more feature engineering and more careful method of missing data as well as more training data we can do better.