**Citi Bike Trend Forecasting**

Chou I Cheong (CIN - 305176968)

Zi Xiao Fu (CIN - 305851759)

Ying Ying Lai (CIN - 305176916)

Department of Information System

California State University, Los Angeles

**Abstract:** This project aims to help Citi Bike build machine learning prediction models to make predictions on several matters that will contribute to the profit of the company. Citi Bike wants to know if their prospective customer tend to use Citi Bike more on weekdays or weekends, because this valuable information will direct them to make better strategy plans in aiming their marketing target and charging on their membership dues. Our team will clean the dataset which collected by the company, and first perform the algorithms in Azure ML. Then we will perform the algorithms again using Spark ML and evaluate which algorithm outperforms the other. Eventually, we will suggest the best algorithm to Citi Bike for the use in predicting their interested matter and improving the company’s profit.

**I. Introduction**

Citi Bike is a privately owned company which operates a public bicycle sharing system. It is currently the largest bike sharing program in the United States with 10,000 bicycles and 600 bike stations. Although Citi Bike is only available in New York City at the moment, this concept of public bicycle sharing system is widely spread among the world as it brings positive impact to the society and environment. In big cities where traffic is highly congested, riding a bike will let commuters travel in a much faster speed, hence saving time. The cost of riding a bike is also lower than taking the subway or hailing a cab, hence another pros to it is saving money. Furthermore, bike riding has zero contribution when it comes to polluting the environment, and at the same time, encourages people to exercise more. For the sake of the benefits of a bike sharing system, our team believe that it is vital that this concept can be further expanded in other big cities. Therefore, our purpose is to help Citi Bike make predictions in matters that are important for them to make better company strategies and gain necessary profit.

**II. Related Work**

Due to the increasing popularity of bike sharing system, many researches and analysis were done on bike sharing system to find insights and relationships that relates to bike sharing. There is one similar work done by Todd W. Schneider in analyzing the New York City bike sharing system. Although similar, the main difference between our work and Todd’s work is that ours is Big Data using Spark on cloud computing, which is extremely valuable in the current world. If big data, cloud computing and business has to meet somewhere, Spark is likely that place.

**III. Background/existing work**

Todd has done an outstanding job in analyzing Citi Bike’s company’s public data of 22.2 million rides from July 2013 through November 2015. After a news which states Citi Bike’s system topped 10 million rides at that time and thus making Citi Bike’s system the world’s largest bike sharing system, Todd felt that it was an opportune time to investigate their publicly available data. Citi Bike’s huge success and deep impact, as well as Todd’s passion and curiosity on finding valuable insights from the public dataset, have urged us to also dig into the Citi Bike’s dataset and make contributions.

**IV. Our Work**

Our team start our work from searching for a suitable dataset from the Citi Bike company’s own dataset collection. Among all the company’s public datasets that are available, we decided that the completed dataset of March 2017 will be the one that is most relevant and suitable to our work. It is a big dataset that is composed of over 100MB of data, and it is also the most current data that is available up to date. After selecting a suitable dataset, we imported the dataset into Microsoft Azure ML platform to clean the dataset. We decide to use linear regression in both Azure and Spark ML; two-class boosted decision tree algorithm in Azure ML and GBT regression in Spark ML.

**Part 1: Regression**

**Perform algorithm in Azure ML**

In Azure ML, we first import our dataset citibike.csv, then we used trip duration, start date, start ampm, end ampm, user type, and gender as features for our prediction in bike riding duration. Trip duration is remained in seconds, but we converted daytime to 1 and nighttime to 2 for start ampm and end ampm; subscriber to 1 and customer to 2 for user type; gender is remained as is which male is shown as 1, female is shown as 2. Following, we cleaned missing data by replacing missing values with zero and normalized the data using z score. We split the data to 70% for train and 30% for test, and used it for both linear regression and two-class boosted decision tree algorithm. After that, we dragged in train model, score model, and permutation feature importance module for each algorithm, and finally performed an evaluation for both models. The results of Root mean square error came out as shown below: linear regression has rmse of 4887.59 and boosted decision tree has rmse of 4854.66. the difference in rmse between the two is almost negligible. While both algorithms performed fairly good, but if compared to each other, boosted decision tree is doing slightly better in Azure ML.

|  |  |  |
| --- | --- | --- |
|  | Linear Regression | Boosted Decision Tree |
| RMSE | 4887.59 | 4854.66 |

Table 1.

**Perform algorithm in Spark ML**

In Spark ML, we want to predict bike trip duration using some features such as trip duration, startdate, startampm, endampm, usertype, gender, the user-defined tree depth set as 5 (default) to prevent overfitting. Then we generate the training and testing data as follow: we randomly partition the data into training set ( say, 70%, 508,623 samples) and testing set (the remaining 30 %, 219,042 samples), then we use cross-validation to prune the tree for GBT regression. whereas linear regression, we use evaluation to measure the performance of model. GBT training takes about 19.98 minutes while the training time for linear regression would be 1 minutes. Although GBT training time is longer than linear regression, but we do not recommend using linear regression model for Citi bike data set since the root mean square error of GBT is lower than linear regression which is 3969.09 seconds(around an hour). Lower RMSE values mean the model is more accurate in making predictions. And also the data has a range of 61 (1 minute) to 1,089,983 (302 hours) then RMSE value of 3969.09 seconds is small, therefore, the result is acceptable. Overall, Gradient Boosting Decision Trees is preferable with higher accuracy result in predicting Citi bike Trip duration.

|  |  |  |
| --- | --- | --- |
|  | Linear Regression | GBT Regression |
| RMSE | 4928.84 | 3969.09 |

Table 2.

**Algorithm Comparison**

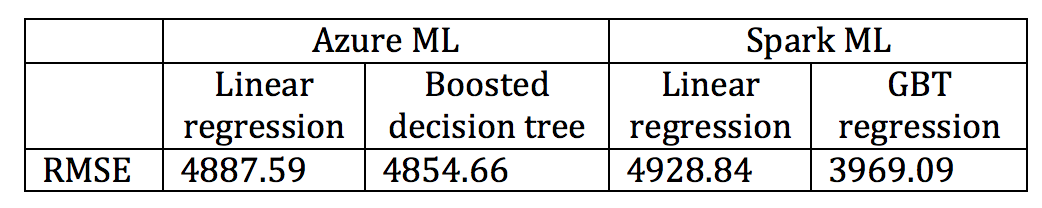


Table 3.

Different platform provides different functions and algorithms. Table 3 interpreted that Two-class boosted decision tree model in Azure ML and GBT tree model in Spark ML have better performance than linear regression. For beginner users, Boosted Decision Tree in Azure ML has a workflow and visual editor that they can easily build their first machine learning project, but for intermediate or advanced users, GBT regression in more desireable, since Spark ML provides faster machine learning compatibility and supports sophisticated pipeline. In addition, our experiment results suggest that GBT regression model is the optimal model for trip duration prediction with regards to Citi bike rental data set.

**Part 2: Clustering**

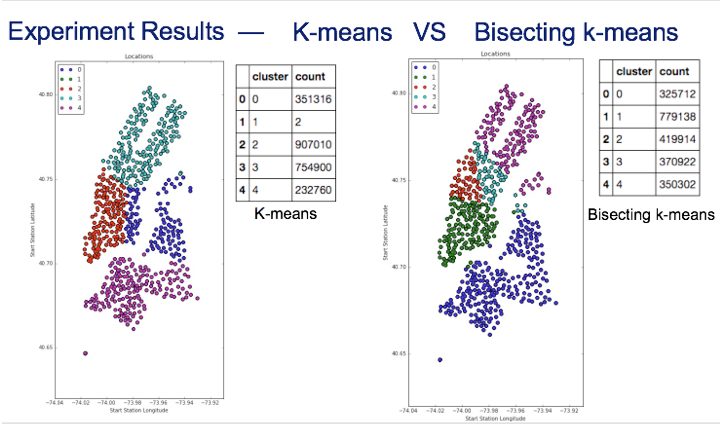
The objectives for this part are using two different clustering algorithms to cluster the location data of the bike stations, segment the data in a geographic-visualized chart. The result can help the New York Citi Bike Company to analysis the popularity of the bike station, predict the traffic stream and adjust the distribution amount of the bike for different stations.

I chose two algorithms which are K-means and Bisecting k-means.

1. First, I want to introduce the K-means clustering. K-means is one of the most commonly used clustering algorithms which can be implemented as an estimator. It can cluster the data points into a predefined number of clusters, and the initial cluster centers will be randomly chosen. The benefit of this algorithm is easy to achieve. But at the same time, random choice of initial cluster centers may cause local minimum cluster. And the processing speed is slow when clustering large scale of data.

2. The second algorithm I picked is Bisecting k-means. This is a kind of hierarchical clustering which can also implemented as an estimator, it is using a divisive approach. All observations start in one cluster, and splits are performed recursively as one moves down the hierarchy. This algorithm often be much faster than regular K-means, but it will generally produce a different clustering. Local minimum cluster will not happen in this algorithm either.

Now let me explain some main steps of doing spark ML using Bisecting k-means. Import the k-means function and loading data first, then I create the k-means model using the code I find in our class lab. Use the features in the geography data to create a Kn-Means model with a k value of 5. This will be used to generate 5 clusters. Then I get the cluster centers, Next, I predict cluster, now that I have trained the model, segment the geography data into 5 clusters and show each data with their allocated cluster. Then I just show up the cluster information and view to ensure the information of those data are complete. At the end, I use two steps of python code to visualize the clustering results into a scatter diagram. Import matplotlib function, import panda function, edit the information of x axis for longitude and y axis for latitude. And the final results are showed as below.

 Figure 1

As we can see, the left one is the result for K-means and the right one is the result for Bisecting k-means. It’s easily to find out that there’s one cluster center in K-means model is local minimum point which contains only two data. Although I create 5 cluster, but only 4 are effective cluster. For the Bisecting k-means model, the results show much better, the 5 cluster has been split well-distributed. So, I picked the Bisecting k-means model as my end result. As we can see the chart, the cluster 1 have largest amount of station data. So, in the local map, this part of place is the place with highest popularity, the company may distribute more bikes to these places to ensure that the market requirements had been satisfied.

The reason why K-means clustering algorithm may cause the local minimum cluster is that the clustering centers were chosen randomly at the very first beginning. Although run a lot of loops may help the spark ML got a better result, but as a algorithm, the Bisecting k-means algorithm can efficiently avoid this problem appeared, the Bisecting k-means can be treated as an better edition for solving the disadvantages from the K-means. It’s more accuracy and save more time. So the Bisecting k-means algorithm is better for this project.

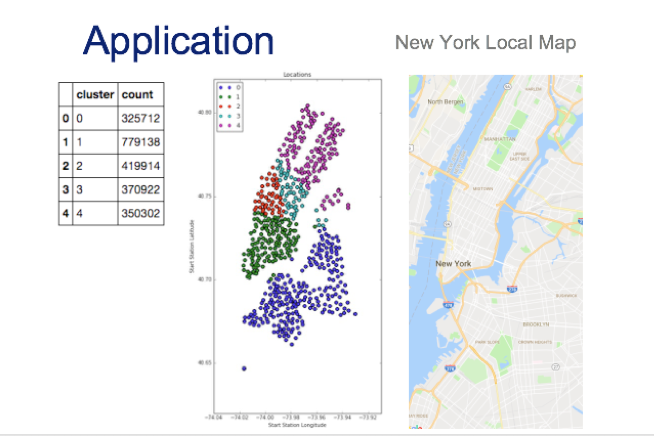


Figure 2

These are my summary conclusion for this clustering experiment.

1. Those two algorithms produce two models which have totally different clusters.

2. One local minimum cluster appears in the K-means algorithm which is not representative and grossly misunderstood.

3. Bisecting k-means is more suitable for this project.

**V. Conclusion**In conclusion, our project objective is to help Citi Bike make predictions in bike riding duration for them to gain necessary profit. This project is valuable in the method it used in making predictions, which is using big data in Azure ML and Spark ML. After performing algorithms such as linear regression, two-class boosted decision tree, and GBT regression in both platforms, we found the prediction results, although different, still very close to each other. One interesting finding from our project is that we figured that boosted decision tree perform better in Azure yet GBT regression performed better in Spark. This tells us that the best algorithm we should use in different platforms could be different. According to our experiment, GBT regression in Spark ML had stood out among others in this particular case.

**VI. References**[1]Todd W Schneider’s Analysis  
<http://toddwschneider.com/posts/a-tale-of-twenty-two-million-citi-bikes-analyzing-the-nyc-bike-share-system/>  
  
[2]Algorithm code learning  
<http://spark.apache.org/docs/latest/ml-clustering.html#bisecting-k-means>  
  
[3]Professor Jongwook,Woo course slides,  
Chapter 13 - Spark ML Unsupervised

[4]Professor Jongwook,Woo course slides, Linear Regression and GBT Regression