Austin Bikeshare Analysis

- Asad Husain -

MGMT 590, Computational Complexity  
Purdue University, Department of Management, 403 W. State Street, West Lafayette, IN 47907

[husain10@purdue.edu](mailto:husain10@purdue.edu)

# Abstract

I am performing analytics to solve business problems for the public bike share system Metro Bike in Austin, Texas. I have access to the bike-rental data in Austin through this [link](https://data.austintexas.gov/Transportation-and-Mobility/Austin-MetroBike-Trips/tyfh-5r8s). I will use concepts from graph theory, linear optimisation, and machine learning to solve the following problems.

* Travelling Salesman Problem
* Predicting trip duration
* Clustering kiosks
* Optimizing which kiosks to close
* Minimum vertex cover

# Data

I have 2014’s data regarding the bikeshare activities and the local weather data in Austin Texas for each day.

The dataset that I imported had 130943 rows and 20 columns (after initial cleaning and merging multiple tables into one). A short description of the columns follows:

1. Trip\_ID – Unique code for each customer interaction with the business
2. Membership\_Type – There are 30 unique categories of customer who rented a bike including 7-Day Membership, Annual Membership, 24-hour Kiosk, etc.
3. Bicycle\_ID – Unique code to identify the bike rented from 375 unique options all over
4. Checkout\_Date – Date when bike was rented
5. Month – Month from the Checkout Date; 1-12
6. TempAvgF – Average temperature of the city in F; 29-92F
7. DewPointAvgF – Average Dew Point of the city in F; 8-74F
8. HumidityAvgPercent – Average percentage of humidity in the air in the city; 27-97%
9. VisibilityAvgMiles – Miles visible to the naked eye, measurement of fog; 2-10miles
10. WindAvgMPH – Average speed of wind on that day; 1-11MPH
11. PrecipitationSumInches - Inches of rain or snow or hail on that day; 0-10inches
12. Checkout\_Kiosk\_ID – Unique ID for the kiosk from where the bike was rented
13. Checkout\_Kiosk – Name of the Kiosk from where the bike was rented; 44 unique kiosks
14. Checkout\_Kiosk\_latitude – Latitude of Checkout Kiosk
15. Checkout\_Kiosk\_longitude – Longitude of Checkout Kiosk
16. Return\_Kiosk\_ID – Unique ID for the kiosk to where the bike was returned
17. Return\_Kiosk – Name of the Kiosk to where the bike was returned; 44 unique kiosks
18. Return\_Kiosk\_latitude – Latitude of Return Kiosk
19. Return\_Kiosk\_longitude – Longitude of Return Kiosk
20. Trip\_Duration\_Minutes – Time in minutes for when the bike was rented by the customer; 2mins – 5days

My dataset does not have any missing values. I changed the datatype for “Checkout\_Date” to datetime and disregarded trips with “Trip\_Duration\_minutes” less than 2.

I created a new column called “Distance” where I stored the distance between the checkout and return kiosks, calculated with the help of their coordinates.

I also created a new column called “Revenue” where I calculated revenue as 1$ to unlock then just $0.23 per minute after that. For sake of simplification, I have used same cost rule for all types of memberships.

# Business Problems and Insights

### Traveling Salesman Problem

Every evening, a company employee visits every kiosk to check for any damage or problematic situation. Currently, that employee visited each kiosk randomly every time and travelled ~43 km overall. There was scope of optimisation here as his route could be fixed based on his starting node.

I used the **travelling salesman algorithm approximation present in the network library (\*1)** to get a better route which would reduce the distance travelled by the employee.

Now his route makes him travel ~37 km, almost 6kms less. Assuming the company pays $10 per km travel concessions to him, this algorithm now has saved $60 per day for the company.

### Trip Duration Prediction

There is an opportunity cost to when a bike is rented as we will have less bikes to offer for the next customer. If we could estimate the unavailability of bikes from a kiosk, we could feed that data in our inventory rebalancing system and we could maintain optimal level of inventory every time.

Based on the bicycle ID, month, various weather conditions of the day, checkout kiosk and its location, and the membership type of the transaction, I predicted the trip duration in minutes. I used **XGBoost algorithm from the sklearn library to predict (\*2)** my target with an accuracy of 8.48%.

Given more computation time or faster processor I could tune it better for a more accurate prediction. (Since XGBoost uses a sequential process to minimize the error of the previous prediction, I could not use parallel computing to reduce the time). Based on this model, now we can predict how long a particular bike will be rented out and thus plan our inventory accordingly.

### Clustering Kiosks

Our R&D team has come up with a new solution for one of our kiosks and want to test it out on different kiosks with similar popularity. As this a test stage we cannot risk it on a popular kiosk as it might affect customer experience. Thus, we need to cluster our kiosks based on the number of trips originating or finishing at them.

I made a graph with all 44 kiosks as nodes and No. of trips between them as the edge strength. I used the **cluster coefficient function of the network library (\*3)** to get the fraction of pairs of the node’s neighbours that are adjacent to each other. Then I used the **Kmeans algorithm from the sklearn library (\*4)** to cluster the nodes based on their cluster coefficients. With the help of the elbow and silhouette plot I could cluster the nodes in 2 groups, high and low popularity.

Now we can make business decisions for the whole cluster and do our A/B testing accordingly among them.

### Downsizing

Due to COVID, the demand of bikes has declined, and the company is considering closing some of the kiosks. The management has asked to shut down the 5 to 10 kiosks which would significantly reduce their operational costs. Choosing the least popular kiosks won’t be correct as the operations cost won’t reduce accordingly. Hence, we need to optimize while keeping high revenue. A special request by the company is that at least 2 of the 4 kiosks around their office should not close.

I used **Linear programming from the pulp library (\*5)** to optimize revenue constraining to more than 34 kiosks total and less than 39 kiosks and the special request. I calculated average revenue for a unique trip and multiplied that by the variable in the objective function. For simplification I assumed that eliminating one kiosk would only reduce the revenue by half.

After solving I got a list of 10 kiosks to shut down which would reduce enough operational costs and not reduce our revenue too much.

### Minimum vertex Cover

Now the company wants to know what the minimum number of kiosks is from where a person has cycled to all other kiosks in one go. The company wanted to mark those kiosks specially.

I used **linear programming from the pulp library (\*6)** to get the minimum vertex cover. I minimised the total no of kiosks needed constrained to each vertex of an edge being covered. To my surprise 41 kiosks were needed to cover all other kiosks in 1 ride.