CS 470 Final Project: Citi Bike

Samantha Lin, Yuritzy Ramos, Matthew Joeseop, Ruilin Chen

April 24, 2023

1 Collaboration Statement

I did not collaborate with anyone, receive any assistance nor use any external sources not cited below.

2 Problem Description

Citi Bike is a bikeshare program with 750+ stations across NYC. Users can temporarily rent a bike from one station and return to any other station for a single ride (3o mins) or for a longer period with a 24-hour day pass or annual membership. We aimed to use link analysis and cluster analysis on the Citi Bike dataset in order to make informed business decision about where to focus resources to improve the user experience.

2.1 Link Analysis: Upgrade and Expand or Retire?

Our team hopes to classify the relative importance/popularity of different bike stations given in the data based on how frequently they appear as start and end destinations for different trips. There were two methods to approach this. First, we could construct a MultiDigraph from the data which would allow for multiple edges between nodes (i.e., five people took a trip from station 380 to 250). This would give a more realistic outcome about a station's usage based on user transactions/trips. However, including these edges may be redundant since they represent the same path/route and will result in the page rank scores of each station being lower. The second method is to use a DiGraph which ignores multiple edges between nodes. This means that it only adds an edge between nodes once, even if the edge appears multiple times in the data. This would result in a page rank outcome based on the stations' interconnectivity rather than how many customers frequent a specific route. This would allow us to observe which stations have the most in-coming links/trips from other stations and how this impacts their importance in the network. We expect there to be a group of stations with higher levels of importance than others and hope to use this information to determine which stations should be expanded/upgraded to increase customer usage. This information could also be used to determine which stations are less frequented and should be removed from the network (i.e., demolishing or selling the station).

2.2 Clustering Analysis: What Kind of Trips Do People Take?

Our team hopes to classify the various kinds of trips that Citi Bike users take using the service. That way, we can identify opportunities for Citi Bike to provide specific offers and services to those riders. In order to answer this question, we take advantage of one of the clustering methods which we learnt in class, which is k-means clustering. We expect to be able to figure out whether there are trips of a certain length, being taken by certain kinds of users. That way, we hope to be able to produce information that could be acted on (e.g. are there ways to make access even easier for users who do these kinds of trips?).

3 Dataset

We got the dataset from Kaggle called Citi Bike 2013-2017. This dataset is a randomly sampled 0.01 of the data on Citi Bike representative of the 2013-2017 period. We only used the trip dataset which has 16 features and 473,557 trips:

- Trip Duration (seconds)
- Start/Stop Time and Date (NYC local time)
- Start/End Station Name and ID
- Start/End Station Lat/Long
- Bike ID
- User Type (Customer = single ride or day pass user; Subscriber = Annual Member)
- Year of Birth of user
- Gender (0=unknown; 1=male; 2=female)

4 Preprocessing

4.1 Visualizing the Data

We made two graphs to visualize the data. Figure 1 shows the number of Citi Bike trips by months of the year. We observed that winter months tended to be the least popular time for trips. That makes sense due to temperature concerns. Figure 2 refers to the length of trips taken by Citi Bike users, and we find that users typically take short trips of 0 to 15 minutes. Aside from that, most users generally have a trip that does not last longer than 30 minutes. This makes sense considering how users may not be so interested in paying for a bike for so long.

Figure 1: Trips by Month of Year

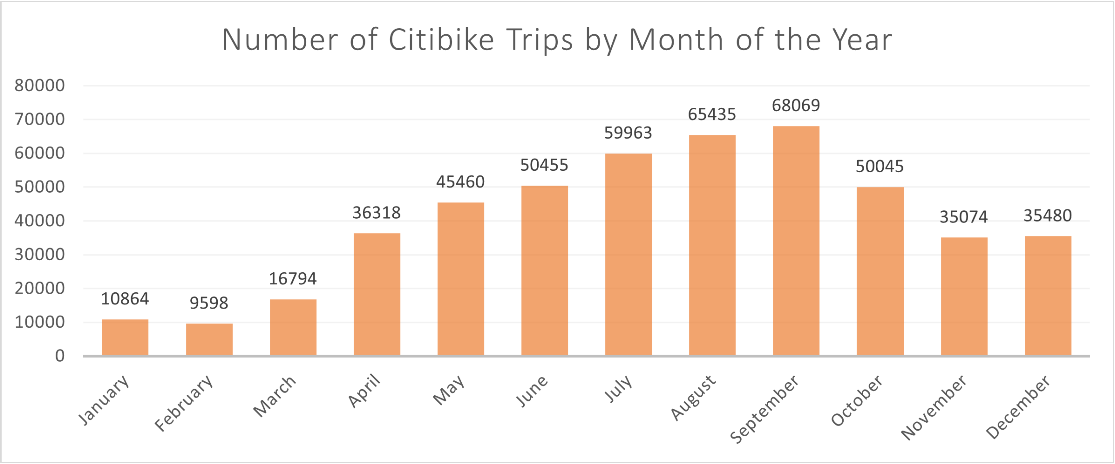
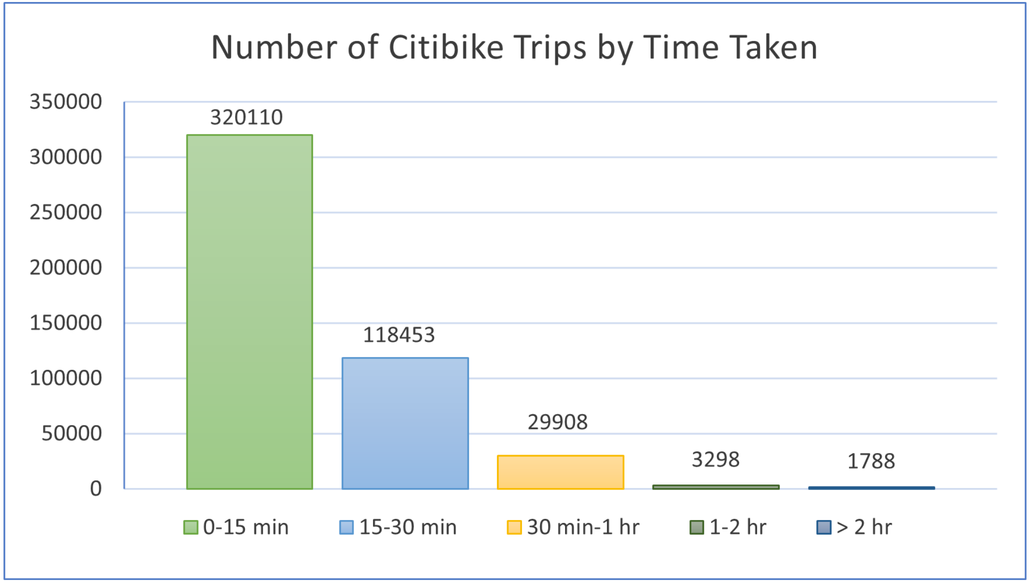


Figure 2: Length of Citi Bike Trips



4.2 Preprocessing for Clustering

For clustering the trips data, we primarily looked at numerical attributes for the clustering algorithm. We did this to attempt to answer the question: what are the most frequent kinds of trips being taken, and by whom? This information could then be used to help all kinds of business functions on the part of Citi Bike.
We did the following preprocessing steps:

- From starttime and stoptime, we extracted an ordinal attribute called "hours_start" which corresponds to how much later in the day did that trip start.
- For the year of birth data, we opted to fill in the missing values by assuming that users with missing birth year data had the birth year of the average user who we did have data with. This has the weakness of skewing the data, but this also allowed us to investigate this specific group of users.
- For the subscriber status and gender data, we used one-hot encoding to turn those variables into dummy variables that would help for clustering. For subscriber status, 1 would mean that the person is a subscriber, and 0 if not. For the gender data, 1 refers to a user being female, and 0 if not.
- Finally, in addition to the above steps, we also performed MinMax scaling such that numerical values were between 0 and 1 to reduce the effect of large numbers/different scales between the attributes.

4.3 Preprocessing for Page Rank

In order to observe any differences between versions of the Citi Bike network demonstrating interconnectivity (single edges) versus trip frequency (duplicate edges), we preprocessed the trip data as follows:

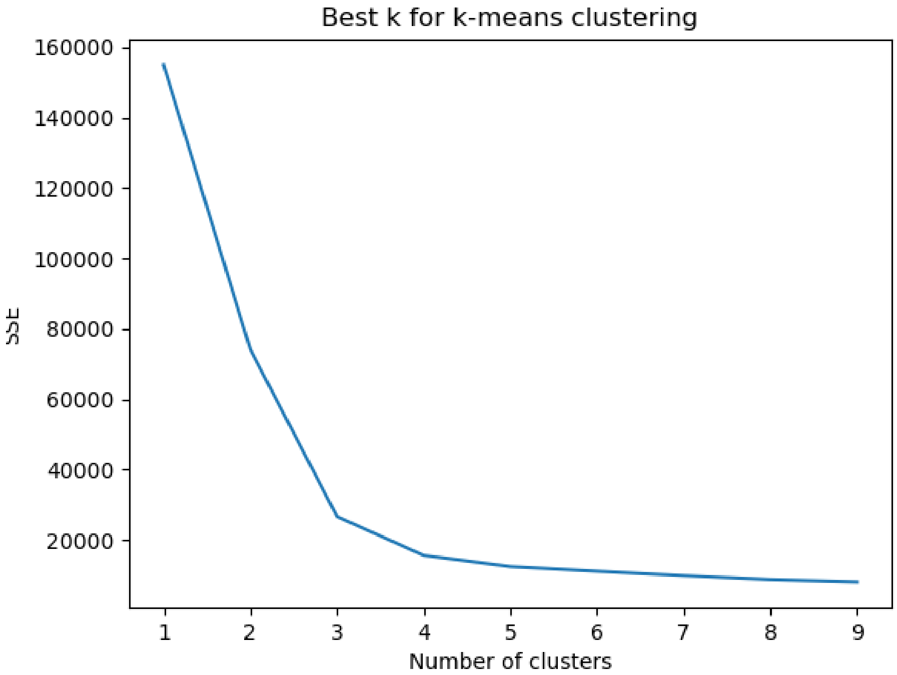
- We created .dot files with and without duplicate edges to be used in the *Mining of Massive Datsets* implementation of PageRank.
- We created digraphs and multi-digraphs which could later be passed to the PageRank function within NetworkX. In this case digraphs do not contain multiple edges while multi-digraphs do

5 Methods

5.1 K-Means Clustering Using scikit-learn

To implement k-means clustering, we used scikit-learn's included KMeans package to run a k-means clustering model on the preprocessed data. The parameters we used specifically for this experiment were a maximum iteration of 10, and a variable k. We chose to cap the amount of iterations at 10 because of computing limitations given the size of the dataset. However, we absolutely recognize that the clustering could be improved if we were willing to have a higher amount of maximum iterations. With that said, the first step in our clustering approach was to identify the "best" amount of clusters. In this case, we ran k-means clustering on k from 1 to 10, and visualized the SSE for the clusters made in the following figure.

Figure 3: Increasing k clusters plotted against SSE of clusters.



This figure shows that based on the elbow method, there are diminishing returns for increasing k or the number clusters beginning after 4 clusters. Hence, we choose to run k-means clustering with 3 clusters. Hopefully, this should result in three relatively distinct clusters.

5.2 Page Rank: NetworkX vs. MDD Implementation

We used PageRank to rank the stations by level of importance in the network. Page Rank is frequently used to determine the importance of web pages based on the number and weight of incoming-links from other web pages. In our case, stations serve as web pages and trips serve as links between them. Stations with more incoming-links are expected to have a higher ranking; however, links from stations that have high rankings will boost the page rank score of the stations they point to.

We decided to use the page rank function from NetworkX since it can take both DiGraphs and Multi-DiGraphs as inputs. By default, the page rank algorithm in NetworkX uses a dampening factor of 0.85 and a maximum of 1000 iterations. For comparison, we also used the page rank implementation from the Mining of Massive Datasets (MMD) textbook to obtain page rank scores for each station. Similar to the NetworkX page rank implementation, we used a dampening factor of 0.85 and a maximum of 1000 iterations.

6 Results

6.1 K-Means Clustering results

The clustering algorithm did result in three relatively different clusters, but this is not immediately obvious from the cluster means of the preprocessed dataset. Indeed, it is not clear whether the differences in attributes like the duration of a trip or the hour that a trip started is very minor. Hence, after obtaining this cluster means and the assigned labels for every row in the data, we attached those labels to the original data and then computed the averages for each of those clusters.

Table 1: Cluster Means for $k = 3$

	tripduration	birth_year	hour_start	is_subscriber	is_female
0	0.000359	0.792675	0.603907	1.000000	$7.052692e^{-14}$
1	0.000423	0.806381	0.604518	1.000000	1.000000
2	0.000992	0.799334	0.630002	$8.115730e^{-14}$	$2.280769e^{-02}$

The following table therefore shows the averaged out statistics of the rows that fall into each of the three clusters:

Table 2: Statistics for each cluster

	tripduration	birth_year	hour_start	is_subscriber	is_female	N
cluster_0	767.810578	1976.950293	13.889871	1.0	0.000000	316901
cluster_1	892.278741	1978.540183	13.903916	1.0	1.000000	99569
cluster_2	2014.201678	1977.722717	14.490050	0.0	0.022808	57086

What the statistics show that there is one huge cluster and two other smaller but still relatively large clusters. The first cluster, which is described by the first row, seems to indicate that the vast majority of trips last around 767 seconds (12 minutes), and occur around 2PM in the afternoon. Furthermore, these are rides by male subscribers. Given how huge this cluster is, there is potential to consider digging into this group of riders/trips to see if there was more to dissect. Nevertheless, it is clear that most trips seem to be relatively short and ridden by subscribes, perhaps pointing to how Citi Bike is used to make small journeys faster.

The second cluster shows that there is a significant group of trips made by female subscribers which start at around the same time but last slightly longer (892 seconds or about 15 minutes). This suggests that perhaps, on average, female subscribers may take a little longer to complete their average ride. It may also suggest that work could be done to encourage more women to subscribe to the Citi Bike service as opposed to men.

The final cluster shows that there are much longer trips of around 2014 seconds (about 30 minutes) which are another significant cluster. In this case, these trips tend to be performed by mostly male riders who are not subscribers. That means that these longer rides were done by users who purchased day or hourly passes to use Citi Bike. Despite this cluster being relatively smaller, the fact that it is still 57086 trips big suggests that there is potential to cater to this demographic of users.

Overall, even from this simple use of k-means clustering, it is evident that there are a few key groups of trips (and users who make those kinds of trips). There are promising avenus of further exploration, and were we to ever have the ability to work with Citi Bike, we could attempt to do further study on how to best serve these specific users. These conclusions were limited by the sparsity of attributes (eg. it would be interesting to know more about the income/other demographics of the user, as well as other details about the trips). Nonetheless, the conclusions are still valid and interesting. A study that extends these conclusions could consider doing a deepdive on the greater Citi Bike dataset (considering this data is a subset of all the data that is available).

6.2 PageRank Results

For the NetworkX implementation of page rank, the start and end station IDs for each trip were obtained from the data set. These were stored in a tuple of form (start, end) and appended to a list of edges for later use. Once all trips were stored, the edges were added to either a NetworkX

Digraph or MultiDigraph. The main difference between the two is that whole MultiDiGraph allows for multiple edges between nodes, DiGraph only counts the appearance of an edge once. The resulting graph was then passed to the NetworkX page rank function with dampening parameter of 0.85 and max iterations 1000. The page rank scores were sorted in descending order and the station IDs were sorted in ascending order. The top 10 results for the graphs with and without duplicate edges can be found in the tables below:

Table 3: NetworkX Page Rank Results for Stations (Duplicate Edges)

Station Name	Station ID	Page Rank
Pershing Square N	519	0.00629
E 17 St & Broadway	497	0.00563
West St & Chambers St	426	0.00503
W 21 St & 6 Ave	435	0.00489
Lafayette St & E 8 St	293	0.00475
Centre St & Chambers St	387	0.00468
Broadway & E 22 St	402	0.004619
Cleveland Pl & Spring St	151	0.004617
Broadway & W 60 St	499	0.00430
Broadway & E 14 St	285	0.00425

Table 4: NetworkX Page Rank Results for Stations (No Duplicate Edges)

Station Name	Station ID	Page Rank
Cleveland Pl & Spring St	151	0.00298
Pershing Square N	519	0.00291
E 17 St & Broadway	497	0.00289
S 5 Pl & S 4 St	532	0.002762
Broadway & W 60 St	499	0.00269581
Centre St & Chambers St	387	0.002623
Mott St & Prince St	251	0.002603
Broadway & E 22 St	402	0.0025953
Lawrence St & Willoughby St	323	0.0025923
Lafayette St & E 8 St	293	0.002586

For the MMD implementation of page rank, two separate dot files were created as input. One is a dot file with all 473,556 edges between the 840 nodes and the other contains no duplicate edges, reducing the number of edges to 107,649. The page rank scores (with dampening factor 0.85 and max iteration 1000) were sorted in descending order and the station IDs were sorted in ascending order. The top 10 results for the graphs with and without duplicate edges can be found in the tables below:

Table 5: Page Rank Results for Stations (Duplicate Edges)

Station Name	Station ID	Page Rank
Cleveland Pl & Spring St	151	0.002991
Pershing Square N	519	0.002913
E 17 St & Broadway	497	0.00288951
S 5 Pl & S 4 St	532	0.002761
Broadway & W 60 St	499	0.002670
Centre St & Chambers St	387	0.002626
Mott St & Prince St	251	0.002610
Broadway & E 22 St	402	0.0025997
Lawrence St & Willoughby St	323	0.002593
Lafayette St & E 8 St	293	0.002592

Table 6: Page Rank Results for Stations (No Duplicate Edges)

Station Name	Station ID	Page Rank
Cleveland Pl & Spring St	151	0.002991
Pershing Square N	519	0.002912
E 17 St & Broadway	497	0.0028985
S 5 Pl & S 4 St	532	0.002761
Broadway & W 60 St	499	0.002698
Centre St & Chambers St	387	0.002625
Mott St & Prince St	251	0.002610
Broadway & E 22 St	402	0.0025997
Lawrence St & Willoughby St	323	0.002593
Lafayette St & E 8 St	293	0.002592

For the top ten results of the trips graphs with duplicate edges, there was a 70% match in the stations between the two methods we implemented - stations 151 (Cleveland Pl & Spring St), 519 (E 42 St & Vanderbilt Ave), 497 (E 17 St & Broadway), 499 (Broadway & W 60 St), 387 (Centre St & Chambers St), 402 (Broadway & E 22 St), and 293 (Lafayette St & E 8 St). The three dissimilarities that occurred were attributed to the random walk aspect of the two algorithms and the influence of the multiple edges on page rank. Although the ranking position for the stations were not always the same, we found it sufficient to approximate the top ten stations for our purposes and so the order does not affect our decisions as far as which stations are the most frequented by customers. It was interesting to note, however, that the bottom ten stations (the least frequented stations) had the same IDs and rankings for the two methods. These were stations 3468 (NYCBS Depot - STY - Garage 4), 3017 (NYCBS Depot - FAR), 3240 (NYCBS Depot BAL - DYR), 3014 (E.T. Bike-In Movie Valet Station), 3450 (Penn Station Valet - Valet Scan), 3485 (NYCBS Depot - RIS), 3506 (Lexington Ave & E 120 St), 3557 (40 Ave & 9 St), 3607 (31 Ave & 14 St), and 3636 (Expansion Warehouse 333 Johnson Ave)

When it came to the graphs without duplicate edges, our results were a 100% match in both rankings and station IDs. The top ten stations were 151 (Cleveland Pl & Spring St), 519 (E 42 St & Vanderbilt Ave), 497 (E 17 St & Broadway), 532 (S 5 Pl & S 4 St), 499 (Broadway & W 60 St), 387 (Centre St & Chambers St), 251, 402 (Broadway & E 22 St), 323 (Lawrence St & Willoughby St), and 293(Lafayette St & E 8 St). The bottom ten stations were the same as in the previous example, stations 3468 (NYCBS Depot - STY - Garage 4), 3017 (NYCBS Depot - FAR), 3240 (NYCBS Depot BAL - DYR), 3014 (E.T. Bike-In Movie Valet Station), 3450 (Penn Station Valet - Valet Scan), 3485 (NYCBS Depot - RIS), 3506 (Lexington Ave & E 120 St), 3557 (40 Ave & 9 St), 3607 (31 Ave & 14 St), and 3636 (Expansion Warehouse 333 Johnson Ave). The similarity between the two page rank methods used for this graph case reveals a potential downside to using the original data with duplicate edges as it produces less consistent rankings depending on the method used compared to singly linked graphs.

Based on our results, it seems that stations 151 (Cleveland Pl & Spring St), 519 (E 42 St & Vanderbilt Ave), 497 (E 17 St & Broadway), 499 (Broadway & W 60 St), 387 (Centre St & Chambers St), 402 (Broadway & E 22 St), and 293 (Lafayette St & E 8 St) appear in the results for both the multidigraph and digraph cases when using NetworkX's page rank function as well as the MDD implementation of page rank. For the sake of completeness, we can also assume that station 253 (W 13 St & 5 Ave), 532 (S 5 Pl & S 4 St), and 323 (Lawrence St & Willoughby St) form a part of this grouping based on the digraph case which produced the same results for both methods. Given their high level of importance in the CitiBikes network, these ten stations are prime candidates for expansion and upgrades.

Finally, stations 3468 (NYCBS Depot - STY - Garage 4), 3017 (NYCBS Depot - FAR), 3240 (NYCBS Depot BAL - DYR), 3014 (E.T. Bike-In Movie Valet Station), 3450 (Penn Station Valet - Valet Scan), 3485 (NYCBS Depot - RIS), 3506 (Lexington Ave & E 120 St), 3557 (40 Ave & 9 St), 3607 (31 Ave & 14 St), and 3636 (Expansion Warehouse 333 Johnson Ave) repeatedly appeared as the least frequented destinations for customers. This suggests that CitiBikes could benefit from retiring these stations to reduce maintenance costs and allocate resources to more frequented stations.

We were only able to use a small portion of Citi Bike's trip history given that the data contained trips from 2013 to 2017. It's important to note that the trips were all made in the NYC area,this means our conclusions are only applicable to bike stations for this specific location and not generalizable to any other Citi Bike locations. We were also unable to visualize the network of stations due to node limitations in the *Graphviz* package. The maximum number of nodes that *Graphviz* allows is 200 and the maximum number of edges is 400. Even when restricting the node and edge data to a single station, the number of edges always surpassed this threshold, prohibiting us from visualizing the relationships between stations in the network.

7 Future Works

The clustering and PageRank analysis of Citibike data offer several directions for future work. First, the dataset we utilized only sampled a very small part of the Citibike data with a limited time frame, so future studies could take a look at the dataset as a whole to identify certain historical trends and usage patterns. Another potential area of focus is integrating our Citibike data with other related data sources such as weather forecasts, traffic patterns, social media, etc. to see if there are any other external factors that could potentially impact bikesharing usage and behavior. Lastly, we could look at typical user demographics of each area.