Mini Project 3 : Shortest Path with NetworkX

Harry Setiawan Hamjaya

***Abstract*— This project focuses on optimizing the Helsinki City Bikes system, introduced in 2016 with 46 stations and expanded to 350 by 2019, providing 3,510 bikes in 2020. The system addresses the "last mile" problem, offering affordable passes for day, week, or seasonal access, allowing unlimited 30-minute rides. Additional charges apply for extended use, but effective use of pick-up and drop-off stations can reset usage, potentially resulting in free rides with subscribed passes. The approach involves data download, graph initiation, exploratory network analysis, with outcomes including Exploratory Data Analysis (EDA) insights, network metrics, and a user-friendly interface for calculating and visualizing the shortest path between locations based on input coordinates.**

Keywords: Last Mile, graph initiation, exploratory network analysis, EDA Insights, network metrics, shortest path

I. Introduction

The "Helsinki City Bikes" initiative, established in May 2016, is a public bicycle system strategically designed to integrate with the broader public transport network in Helsinki and Espoo. Operated as a collaborative effort between key entities such as the Helsinki Regional Transport Authority (HSL), Helsinki City Transport (HKL), Espoo Technical and Environment Services, Moventia, and Smoove, this system aims to address the pervasive "last mile problem" in urban transportation.

Comprising 3,510 shared bicycles across 241 stations in Helsinki and 110 in Espoo as of 2020, the Helsinki City Bikes offer accessible and sustainable transportation options. Users can purchase passes for a day, a week, or the entire cycling season, providing unlimited 30-minute rides. The bikes are conveniently returned to designated stations after use, reinforcing a seamless and eco-friendly commuting experience.

Recognizing the significance of optimizing routes for cost reduction, our study aims to explore the operational dynamics of the Helsinki City Bikes system. The overarching goal is to identify the most efficient pathways for traversing Helsinki and Espoo, mitigating the risk of increased usage duration and associated charges. Furthermore, our research underscores the strategic utilization of pick-up and drop-off stations, presenting an avenue for users to seamlessly exchange bicycles, reset usage, and potentially access free rides with subscribed passes. This paper delves into the critical aspects of the bike-sharing system, aiming to contribute valuable insights for urban mobility planning and the ongoing discourse on sustainable transportation solutions.

II. Preliminary Data processing

A. Dataset

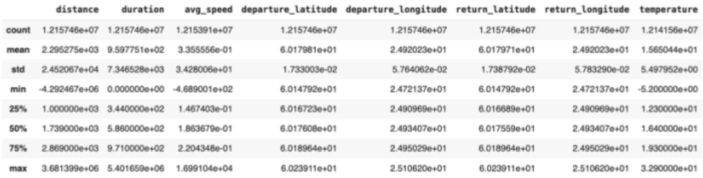
The dataset size is 1.99GB and downloaded from Kaggle website, specifically on this link: https://www.kaggle.com/datasets/geometrein/helsinki-city-bikes . The information in the dataset covers the dataset contains more than 10 million trips made by citizens of Helsinki between 2016-2020. It includes several types of data such as date time, integer, float, and string type dataset. The list of features that is available in the dataset are listed as follow:

1. 'departure': Date Time type of departure,
2. 'return': Date Time type of return,
3. 'departure_id': ID of departure,
4. 'departure_name': Name of the departure station,
5. 'return_id': ID of return,
6. 'return name': Name of the departure station,
7. 'distance (m)': Distance of the trip,
8. 'duration (sec.)': Time duration on the trip,
9. 'avg_speed (km/h)': Average speed in Km/h,
10. 'departure_latitude': Latitude of the departure station,
11. 'departure_longitude': Longitude of the departure station,
12. 'return_latitude': Latitude of the return station,
13. 'return_longitude': Longitude of the departure station,
14. 'Air temperature (degC)': Air temperature during the trip

B. Data Analysis and Preprocessing

First, we have to analyze the whole dataset before initiating the graph. We use describe function to print out the Integer and Float type data statistic information such as count, mean, std, min, max and the quartile (25%, 50%(median), 75%).

Table 1: The distribution of grades toward All Scores



	distance	duration	avg_speed	departure_latitude	departure_longitude	return_latitude	return_longitude	temperature
count	1.215746e+07	1.215746e+07	1.215391e+07	1.215746e+07	1.215746e+07	1.215746e+07	1.215746e+07	1.214156e+07
mean	2.295275e+03	9.597751e+02	3.355556e-01	6.017981e+01	2.492023e+01	6.017971e+01	2.492023e+01	1.565044e+01
std	2.452067e+04	7.346528e+03	3.428006e+01	1.733003e-02	5.764062e-02	1.738792e-02	5.783290e-02	5.497952e+00
min	-4.292467e+06	0.000000e+00	-4.689001e+02	6.014792e+01	2.472137e+01	6.014792e+01	2.472137e+01	-5.200000e+00
25%	1.000000e+03	3.440000e+02	1.467403e-01	6.016723e+01	2.490969e+01	6.016689e+01	2.490969e+01	1.230000e+01
50%	1.739000e+03	5.860000e+02	1.863679e-01	6.017608e+01	2.493407e+01	6.017559e+01	2.493407e+01	1.640000e+01
75%	2.869000e+03	9.710000e+02	2.204348e-01	6.018964e+01	2.495029e+01	6.018964e+01	2.495029e+01	1.930000e+01
max	3.681399e+06	5.401659e+06	1.699104e+04	6.023911e+01	2.510620e+01	6.023911e+01	2.510620e+01	3.290000e+01

As evident from the table 1, there exist certain irregularities in the data, such as negative and excessively large distances. Addressing these anomalies requires contextual understanding of the data.

1. Regarding Distance:

   In the Euclidean space, distances cannot be negative. However, merely filtering for positive values is insufficient. To eliminate unusual cases, trips covering less than 50 meters are considered irregularities within the context of this Exploratory Data Analysis (EDA), as stations are always positioned more than 50 meters apart.

2. Concerning Duration:

Setting appropriate upper and lower limits depends on the specific case. The maximum allowed rental time for a bike is 5 hours, with users exceeding this duration incurring an 80€ penalty. Therefore, the 5-hour mark (18000 seconds) serves as the upper limit for trip duration. The lower limit is straightforward, determined by the previously defined limits: a 50m distance limit at an average speed of 25km/h.

3. Regarding Temperature:

Examining the temperature data, the only anomaly observed is a recorded temperature of 32 degrees in Helsinki. This anomaly is likely attributed to global warming rather than an error in our dataset, as indicated by the table above.

Based on these criteria, the dataset can be refined by excluding records that fall into the anomaly categories. This filtering process involves removing instances with negative or extremely large distances, trips covering less than 50 meters, and durations exceeding the 5-hour limit. Additionally, anomalies in temperature, such as extreme values like 32 degrees, can be excluded to enhance the dataset's accuracy and reliability.

III. Data analysis

For the visualization analysis for this analysis:

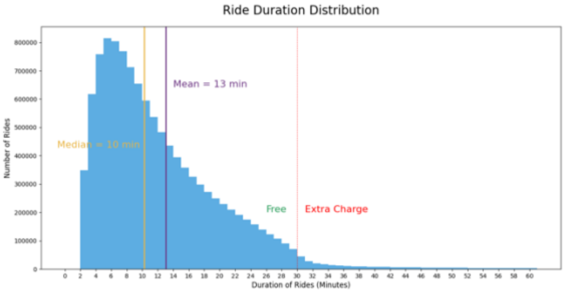


Figure 1: The distribution of duration toward the rides

As depicted in the figure 1, most rides are under 30 minutes. Nevertheless, a small percentage, specifically 3.176% of users, surpassed this limit. Those exceeding the 30-minute but not the 60-minute threshold have cumulatively paid €261,715 since the introduction of city bikes in 2016.

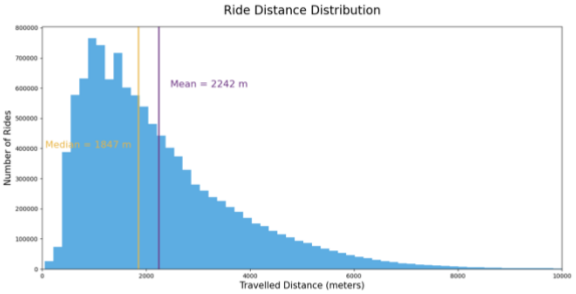


Figure 2: The distribution of distance toward the rides

Based on figure 2, The city bike system has experienced substantial growth since 2016; nevertheless, the usage patterns of city bikes have remained relatively consistent. Analyzing individual trips over the past five years reveals an average ride duration of approximately 13 minutes and an average traveled distance of around 2242 meters (1.4 miles). Due to the right-skewed distribution of the data, the averages are slightly influenced, with the majority of trips lasting between 4 to 8 minutes and covering a distance of approximately 1700 meters.

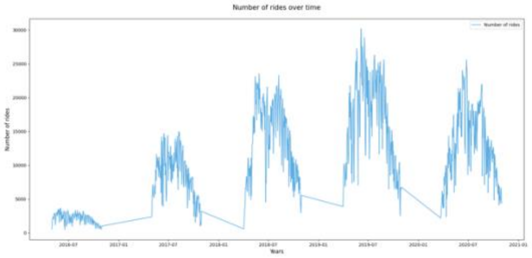


Figure 3: The number of rides regard to time

Displayed in the figure 3 is the count of daily bicycle trips since the inception of the city bike system. The data illustrates a significant correlation between expanding the network coverage and an increase in the number of trips taken by residents. Notably, 2020 marked the initial decline in bike usage. Various factors could account for this decrease, including the impact of the COVID-19 pandemic or the possibility that the city bike network has reached the conclusion of its growth phase.

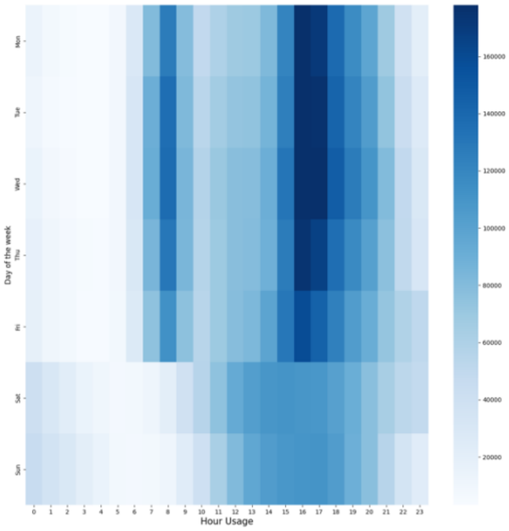


Figure 4: The heatmap of hourly usage

Based on figure 4, during weekends, usage patterns diverge. It appears that Helsinki residents opt for a later start to their weekends, with peak activity observed between 3:00 PM and 5:00 PM. Notably, city bike usage peaks around midnight on weekends, suggesting a potential substitution for other public transportation options that might be unavailable during those hours.

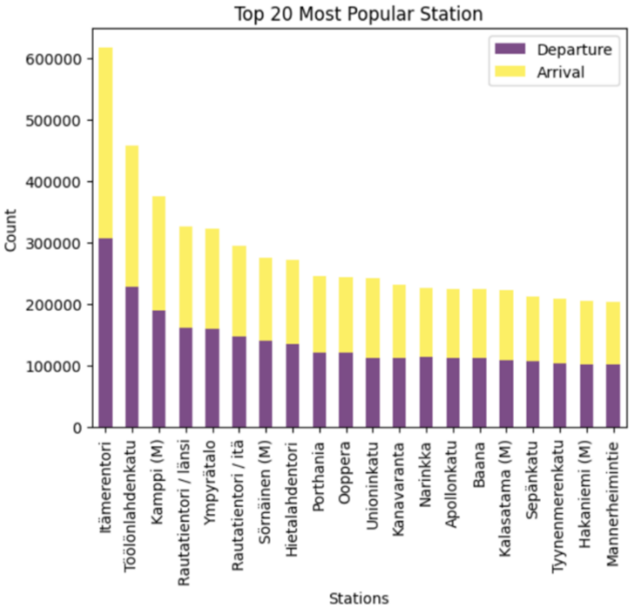


Figure 5: The number of rides regard to time

Not all bike stations are equally popular. In 2016, Kamppi was the top station, but since 2017, Itämerentori has taken the lead. The surprising popularity of Itämerentori and Töölönlahdenkatu is due to their strategic positions in the expanding bike network, even though they're not in the center of Helsinki. Their importance increased as the network grew northward, shifting the central hub. Bike availability plays a vital role, and Itämerentori and Töölönlahdenkatu are popular for both starting and ending trips, ensuring bikes are consistently available and boosting overall station use.

There are a total of 11,278,850 data points; however, it's crucial to note that not all recorded transportation is free, as some trips exceed the 30-minute free pass. In addressing this, we aim to exclude all trips equal to or exceeding 30 minutes since our goal is to identify the shortest path within this time frame. By constructing the graph using data points with durations less than 30 minutes, we ensure that completing the trip between two points is feasible within this timeframe. After this exclusion, we are left with 10,920,525 data points.

IV. NetworkX

Since we are building a graph network with NetworkX module, we try to fit the datapoints from the dataset into the graph, with each station both departure and arrival(return) are fitted as the corresponding nodes(vertexes), and the edge would be the distance between the stations/nodes. The result could be plotted, with blue circle as the node and the edge would be dotted gray line. The bigger the node circle then the more datapoints that it has compared to the other as depicted in the figure 6.

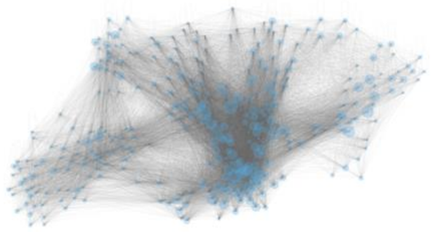


Figure 6: NetworkX graph for Helsinki City Bike

The important stats from the NetworkX are listed as below:

1. Number of nodes 347
2. Number of edges 26913
3. Average degree 155.11815561959654

In NetworkX, Degree centrality is a simple way to figure out how popular or connected a point is in a network. In our city bike scenario, it tells us how many other bike stations people have ridden to from a particular station. If we look at the graph, we can see that stations in the middle of Helsinki are more popular and have more connections. As you move away from the center, the stations have fewer connections.

Top 10 nodes (stations) by degree:

1. ('Haukilahdenkatu', 303)
2. ('Paciuksenkaari', 257)
3. ('Huopalahdentie', 256)
4. ('Munkkiniemen aukio', 255)
5. ('Tilkanvierto', 253)
6. ('Laajalahden aukio', 248)
7. ('Paciuksenkatu', 246)
8. ('Pasilan asema', 243)
9. ('Töölöntulli', 242)
10. ('Saunalahdentie', 242)

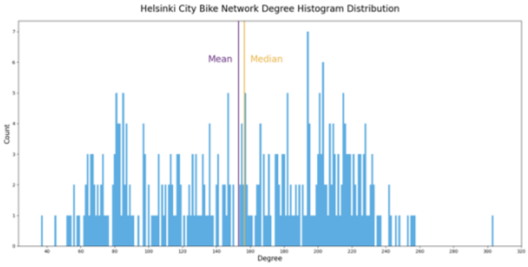


Figure 7: Centrality Distribution

"Haukilahdenkatu" is an outlier since from the histogram plot the degree centrality, it stands at the right side by itself. For a better plot toward the centrality on the map, we would remove the "Haukilahdenkatu" datapoints from the Graph. The centrality plot map is depicted on figure 8.

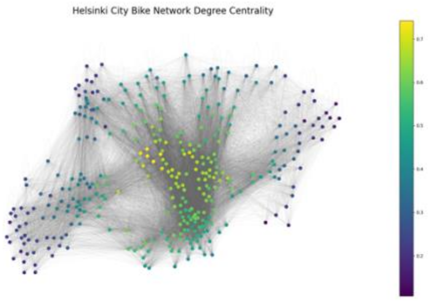


Figure 8: The centrality plot graph

In NetworkX, Betweenness centrality is like a traffic cop in the network. It looks at how often a point acts as a bridge between other points. It counts how many times the shortest routes between different points pass through it. In the graph, you can see the city bike stations ranked by their betweenness centrality, showing which ones are like key bridges in the network.

Top 10 nodes (stations) by betweenness centrality:

1. ('Haukilahdenkatu', 0.06908714520533041)
2. ('Lehtisaarentie', 0.009344749491718398)
3. ('Kalasatama (M)', 0.007789453583024854)
4. ('Luoteisväylä', 0.006880588904980379)
5. ('Puistokaari', 0.00662208372040781)
6. ('Paciuksenkaari', 0.006439267278017997)
7. ('Saunalahdentie', 0.006270598565727013)
8. ('Lauttasaaren-ostoskeskus', 0.00591291440203442)
9. ('Munkkiniemen aukio', 0.005887068733614538)
10. ('Huopalahdentie', 0.005859709376718192)

In NetworkX, "diameter" is like measuring how far apart the two most distant points are in a network. It's like finding the longest distance between any two places on a map. The diameter of a network tells us how "big" or "spread out" the network is. If the diameter is small, it means most points are relatively close to each other, while a larger diameter suggests that some points are quite far apart in the network. In this case our network diameter is 3.

In NetworkX, "density" is a way to describe how packed or crowded a network is. It's like looking at how many connections or relationships exist among the different points in the network. A high density means there are lots of connections, so it's pretty crowded, like a busy social network. A low density indicates there are fewer connections, so it's more spread out, like a sparsely populated area. In our case, the value is 0.44831836884276455.

In NetworkX, "node connectivity" is a way to measure how strong or important a single point, or node, is in a network. It's like assessing the importance of a hub in a wheel; if you remove that hub, the wheel doesn't work well. Node connectivity tells us how many connections need to be broken to isolate a specific node from the rest of the network. A higher node connectivity means that the node is crucial for keeping the network connected and functioning. In our experiment, the value of node connectivity is 35

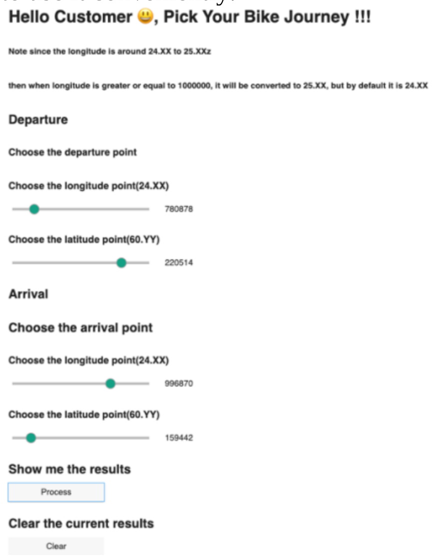
In NetworkX, "shortest path length" is like finding the quickest way to get from one point to another in a network. It's similar to discovering the shortest route between two places on a map. Shortest path length tells us how many steps or connections we need to follow to go from one point to another. It's a measure of efficiency in the network, showing how easily information or influence can travel between different points. This will be the focus of our solution for finding the minimum path.

V. SOLUTION

Since in NetworkX, there is already a feature to calculate the shortest path, then we can jump directly into how to define the nearest location based on the longitude and latitude position. The solution step:

1. Take the input from user for both departure and arrival/return longitude and latitude postions.
2. We can use the Haversine formula, a trigonometric function, to compute the distance between two points on the surface of a sphere (such as the Earth).
3. Since we have the desired points and the nearest station, then we can set the nearest station as the node for either departure or return/arrival.
4. Calculate the shortest path using the available feature from NetworkX.
5. Print out the path.
6. Dump the route into other graph, so that we can draw it as the final solution.
7. Plot the desired path with red line as it shows the path that is recommended based on NetworkX analysis.

All the steps have been implemented and tested on the google Collab, with a simple UI as shown in figure 9, users are expected to use it conveniently.



Hello Customer 😃, Pick Your Bike Journey !!!

Note since the longitude is around 24.XX to 25.XXz

then when longitude is greater or equal to 1000000, it will be converted to 25.XX, but by default it is 24.XX

Departure

Choose the departure point

Choose the longitude point(24.XX) 780878

Choose the latitude point(60.YY) 220514

Arrival

Choose the arrival point

Choose the longitude point(24.XX) 996870

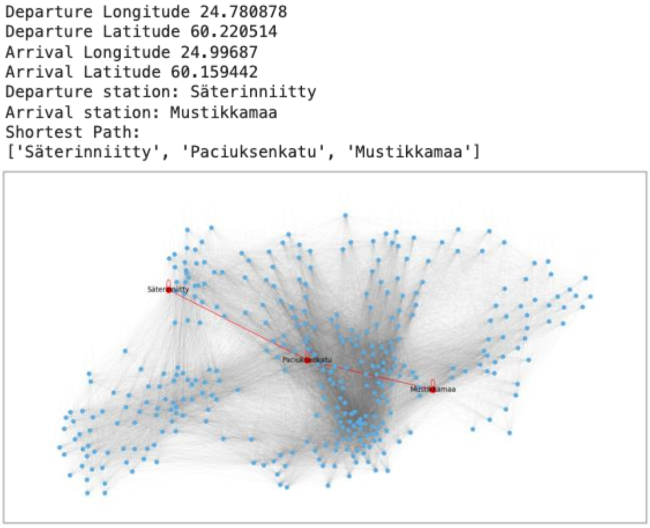
Choose the latitude point(60.YY) 159442

Show me the results

Clear the current results

Figure 9: Simple UI on Collab

The expected result would be depicted as figure 10. It shows the Departure Longitude and Latitude respectively followed by the Arrival. Consecutively, it will print the Departure and Arrival station respectively and finally the Shortest Path with the given array as how the shortest path should be taken.



```
Departure Longitude 24.780878
Departure Latitude 60.220514
Arrival Longitude 24.99687
Arrival Latitude 60.159442
Departure station: Säterinniitty
Arrival station: Mustikkamaa
Shortest Path:
['Säterinniitty', 'Paciuksenkatu', 'Mustikkamaa']
```

Figure 10: Recommendation Result on the Map

VI. Concusion

In conclusion, NetworkX is the other way to treat the data instead of fitting and training. It gives a very good experience about how we can solve the problem with the different point of view. Starting from analysing the feature data, such as removing the unrealistic and anomaly data is the first important step that should be taken by the Data Scientists. The second step is visualizing the dataset distribution, to grasp some information such as Helsinki bikes were used significantly to commute for worker, since it has a high peak on 7-8 A.M and 4-5 P.M. The other information is most of the time people use it for free after charging the seasonal passes, proven by the data that is right-skewed. Last but not least there is also a significant improvement for Helsinki bike usage based on the 2016-2020 data.

Making progress on this task becomes much easier if we find the right dataset that matches the problem we're trying to solve. Once we have that, the next steps are pretty straightforward. Creating visualizations isn't too hard either, especially since we've learned a lot from previous tasks. Plus, using NetworkX brings added benefits. It's like having a powerful toolbox that helps us solve different problems that might come up during the analysis. In a nutshell, having the right data, using visualizations, and leveraging NetworkX's features set us up well to handle the task efficiently.

This task might seem a bit challenging for those who are new to NetworkX because it takes some time to get the hang of it. However, it's not more complicated than the first task we did because this time there's only one thing to focus on, not three like in the initial project. Another thing that might be a bit tricky is figuring out the problem we want to solve. Usually, we're given problems to work on, but this time we have to come up with one ourselves. It's a cool part of the task, but it might be a bit tiring for the teacher to read and approve all our ideas. In my case, calculating the nearest two positions with the haversine formula seems really hard, but since we can research it and it was also discussed in the GPU class, understanding the trigonometry calculations shouldn't be that tough; the challenge lies in generating the idea of using the haversine formula.

Overall the result meet the project expectation. Since the problem statement was about optimizing the Helsinki bikes usage, by finding the shortest path, that could be travelled in less than 30 minutes (which is the free usage threshold for the seasonal pass). In future, easily I could recommend that instead of playing with Longitude and Latitude, easily just take the station as the input for both Departure and Arrival, but then there would be less challenging. For the mini project 3 input, I believe this task complexity and difficulty is ok compared to the first one, even though it has higher score compare to the first one, since we can define the difficulty at our limit based on our estimations. Thank you for the whole two months journey.