ANALYSIS ON TRAVEL PATTERN OF CITI BIKE USERS IN JERSEY CITY

CSE 163

Kairui Huang

Runbo Wang

Danhiel Vu

Table of Contents

Executive Summary

Q: How specific locations correlates with the frequency of Citi Bike ridership?
A: Most of the bike trips are located inside Newport, Historic Downtown, and The Waterfront. Furthermore, most of the bike trips are near transit stations.

Q: How are the trends of Citi Bike ridership in Jersey City affected by time, age, gender, and user type?
A: Most people bike less than 10 minutes and age group of Citi Bike users is around 30~40. Total trip duration difference between male and female is not big, but the total duration of users who don't want to specify their gender is much bigger than users with specified gender, also unsubscribed users are much more than the subscribers. It seems fewer people would want to ride a bike in winter, and not many like biking in midnight.

Q: Is it possible to predict the users' destination based on information acquired when they unlock the bikes? If it is possible, this would give the Citi Bike company information to better strategically place their stations based on user's trip data. Use a machine learning model to explore this possibility.
A: Yes, it is possible to predict the user's destination by using either a decision tree model or multilayer perceptron model.

Q: What are the correlations between the dependent variable and all independent variables based on model build in Q3? How to improve bike share performance?
A: In our trip dataset, the current month, season, hours, and start station id features played the most significant role in training the machine learning models. While not as prominent, the gender, peak, and user type independent variables contributed to the dependent variable as well. Through this data, we could improve bike share performance by marketing and clustering Citi Bikes in specific areas based on the current time of year.

Background

Nowadays, the United States is promoting public transit, cycling and electric vehicles all over the country because of the industrial development of sustainable energy. Docked bikes play an important role and are considered the future of this. We are motivated to further propagate the number of docked bike users to further advocate for energy sustainability, however, in order to do this, we first need to thoroughly understand the common characteristic of the user groups, the factors that enable people to use docked bikes, and the trends of the docked bike development. The result of the first two questions should present a simple and clear demonstration of the performance and trends of docked bikes in Jersey City. Combining with the result of the last two questions, it can also help the docked bike agency for further improvements, such as allocation of bikes based on season, transit station, or prediction.

Data

The bike-share data is acquired from the Jersey City Citi Bike website. The data contains information about the Citi bike trip histories from 2015 to 2019. The attributes of the data contain trip duration, start/stop time, start/end station, station ID, station latitude/longitude, bike ID, user type, gender, and year of birth.

Also, there is another dataset acquired from Jersey City Public Transit Open Data. There are multiple types of file are available such as either csv file or Json file. The only information in this dataset is the location and name of each public transit station in Jersey Cities.

The download link for Citi Bike trip data: https://s3.amazonaws.com/tripdata/index.html

The download link for Jersey City Public Transit Stations: https://data.jerseycitynj.gov/explore/dataset/jersey-city-public-transit/export/?location=11,40.66554,-73.87379&basemap=jawg.light

Methodology

Since more details about the specific approaches to address each of the research question, the following paragraphs will instead, briefly describe the general idea for the aforementioned research questions, respectively.

1. First, create a buffer for each transit station represent the maximum accessible distance. Second, spatial join all bike docks with the buffer and only keep station inside buffer. The other stations will be classified as not near any transit station. It's also expected that one bike dock might fall into multiple transit stations' buffer. Third, use spatial relation to determine how to allocate the trip counts of one bike dock to multiple transit station. As the result, a table with concluded trip count and trip duration corresponding to each transit station should be computed.
   For visualization, the distribution of all bike docks and popularity of each transit station can be plotted. Then, plot a pie chart displaying the percentage of trip counts in different transit station buffers, include trip count for bike docks that are not near any transit station.
2. Re-process the dataset and add the columns needed to show on plots, then draw different plots using different functions to show the connections between factors
3. The machine learning will apply both decision tree classifier and multilayer perceptron classifier. The dependent variable will be the end station name while the independent variables will be season, month, peak, gender, period, birth year, user type, end station name, and start station id. To begin, using the Citi Bike trip dataset in Jersey City, we were able to accumulate just over a million rows of data. With this split the data in two: 80% being the training data and the remaining 20% being the test data.

The next step was using this data to train a decision tree classifier model. We then used the model to plot a line graph representing training and testing accuracy over max depths in order to acclimate for overfitted models. Next, use the model to calculate the accuracy score. In order to be completely sure of our answer, we decided to use two machine learning models. So, the next model we will use is multilayer perceptron. First, we use the data from before and tune the hyperparameters to find the best learning rate for the neural network. In the end, we found that in such a large-scale dataset, the hidden layer size would be (150, 150, 150). Train the newly created MLPClassifier and calculate training, testing, and accuracy score.

4. Find the coefficient of correlation to each variable and discuss possible reasons why the coefficients are positive or negative. With these inferences, we then recommend feasible suggestion to the bike-share operators. Besides, the recommendation will not include determinant factors such as gender, even though we will still consider them as the factors while building the model

Procedure

Get Data

1. From https://s3.amazonaws.com/tripdata/index.html, we download all the needed CSV files. Since there are more than 50 files, download them by hand can be a pain, so we use a *for* loop to generate a list of urls we need, then make a request to those urls and download each zip file from the index using *urlretrieve* imported from *urllib.request*. Then use *zipfile* module to extract all the zip files to CSV files and save them in one folder. Next, we clean the zip files to avoid unnecessary memory usage.
2. The geojson file and shp file are manually downloaded from the Citi Bike Open Data and Jersey City Open Data. The links are attached as the following:
3. Public Transit station: https://data.jerseycitynj.gov/explore/dataset/jersey-city-parcels/export/
4. Basemap of Jersey City: https://data.jerseycitynj.gov/explore/dataset/jersey-city-public-transit/map/?location=12,40.71851,-73.96494&basemap=jawg.light
5. After previous steps, *for* loop and *os.listdir* are applied to search each CSV file. Then *Pandas* is used to read and combine them into one. Before appending these dataframes, we need to lowercase and eliminate all whitespace for all column names because sometimes they are not uniform format. Lastly, we extract this big dataframe to CSV for future use.

Clean Data

In order to analyze travel pattern of Citi Bike users in Jersey City, it's necessary to filter outliers and infer hidden factors from given information. The following demonstrates all data processing have been done:

1. Any row contains NaN or with a destination station outside Jersey City are filtered.

2. Age is calculated by first changing the start time column from string to year using *pd.to_datatime().dt.year*, then subtract this number by given birth year. Besides, some misinput lead the age to be huge, so, age greater than 90 are filtered, assuming they won't ride bike since it would be ridiculous if someone with 100 years old riding share-bike.
3. Hour is also transformed from the start time column using *pd.to_datatime().dt.hour*. With the hour information, time period of each day is classified into 4 categories: midnight, morning, afternoon, and evening based on 00:00~04:59, 05:00~11:59, 12:00~16:59, 17:00~11:59, respectively. Peak hour is determined to be 6:00~09:59 and 16:00~19:59. Off-Peak hours are the remaining hours.
4. Lastly, month is transformed in the same way and compute the season: winter, spring, summer, and autumn. The range of them are equally distributed to be Dec~Feb, Mar~May, Jun~Jul, and Aug~Nov, respectively.

Spatial Analysis for Q1

1. Transform the filtered bike data to geopandas dataframe using *gpd.GeoDataFrame()* and sum the values by start station id using *dissolve()*.
2. Take the transit station data; filter type other than light rail; and create buffer for each transit station using *GeoSeries.buffer()*. The parameter uses 0.01 in our project because of location. When New Jersey's Coordinate is 40.0583° N, 74.4057° W, it means 1 decimal degree change correspond to about 80 km change on Earth. So, a 0.5 mile = 0.8 km buffer is about 0.01. Besides, the reason why we set 0.5 mile as the threshold is according to the definition from TCQSM[1].
3. Join the bike data to transit station data using *gpd.sjoin()*. We also ensure the consistency of coordinate reference system for both dataframes using *GeoSeries.crs*. In this step, we generate a modified bike dataframe that only contains stations within the transit station buffer. We also have another new dataframe for transit station that only contain stations holds at least one bike station.
4. We merged the above two modified dataframes again, so we have filtered all unnecessary data, at least for now. Then we use *GeoSeries.distance(self, other)*. Because it requires two dataframe, we split the bike related columns and transit related columns into two dataframes for distance calculation. And we will add this calculated distance to a previous dataframe that merged bike and transit station.
5. We then apply *groupby()* to sum the distance by bike station. Add this total distance column to the same previous dataframe that merged bike and transit station. Now, in this merged dataframe, each bike station has information about distance to each transit station and the total distance from this bike station to all transit stations. Then, we can calculate the distance ratio and distribute the bike counts and trip duration of each bike station to multiple transit stations proportionally based on distance. Then we again apply *groupby()* to sum all bike counts and trip duration by bike stations and form them as the conclusion dataframe.
6. And now, it's time to reconsider those filtered bike stations, which are not within transit station buffers. We first find all these stations, then sum all their trip counts and trip duration

[1] Capacity, T. (2014). Quality of Service Manual (TCQSM), (2013). Part–7, 2nd ed. Kittleson and Associates.

and add this new row of data called "Stations not near transit" to conclusion data using *append()*. Here is all the ridership information (trip counts and trip duration) we have computed.

7. For visualization, first, the distribution of stations plot is created. The bottom layer is the basemap using shape file from Jersey City parcels data. The second layer requires the buffer dataframe to merge with conclusion dataframe which contains % trip for coloring purpose using *plot(column=?)*. The third layer will be the unfiltered bike stations dataframe. The top layer will be filtered bike stations dataframe with different color.
8. The last spatial analysis is a pie chart. We first filter all rows of data with % trip lower than 1% otherwise they will all collide together and look messy. So we sum them by *groupby()* and add it back to conclusion dataframe. To draw a pie chart, we use *plt.pie()*.

Graphs for Q2

1. A scatter is drawn using *seaborn.relplot()*. Before plotting, we first select the columns needed, include age, trip duration, period (morning/etc), and month. Then, we average the age and trip duration by period and month. Then, *DataFrame.reset_index()* is applied to retrieve the columns of period and month from index. The y-axis and x-axis are Average Age and Month, respectively. The color of each dot represents Season and the size represents the Average Trip Duration.
2. Since we want to show the connection between gender/user type and year using *seaborn.lineplot()*, we need a column representing year, but we only have a date column indicating the start time and end time, so we use *pandas.to_datetime()* to transform the starttime column and create a new year column. Then we *pandas.groupby()* the original data using gender/user type and year, then *seaborn.lineplot()* the connection between gender and year, as well as the connection between user type and year, and put the two plots into one figure using *matplotlib.pyplot.subplots()*.
3. We want to show the trip number in different seasons and time periods using *seaborn.barplot()*. First of all we need the trip counts, so we *pandas.groupby()* all the trips by seasons/periods, one row counts as one trip and *sum()* all the trips together. Then we plot two bar plots: seasons versus trip count and time periods versus trip count, and put them in one figure using *matplotlib.pyplot.subplots()*.
4. Last, we want to plot the monthly trip numbers in different years. So we count all trips like the last figure then *pandas.groupby()* the data by month and plot the data of every year using *seaborn.lineplot()*.

Modeling for Q3 & Q4

1. To reproduce our results with our machine learning models, we must first filter the data we have obtained from the previous instruction in *get_data*. Pass in that data in *filter_data* in ml_model.py and that should return a filtered_data variable that will be passed in for the following next steps.
2. First plot the Decision Tree Classifier model to discover the test and training accuracy score over the max depth. This is to prevent and search for an overfitted/underfitted model. To do

this, run the *plot_dtc_accuracy* method in ml_model.py and pass in our newly filtered_data variable. This should save two plots in your directory named train_dtc.png and test_dtc.png. With this data, we determined that the best max_depth is 20

3. Next, to get the score accuracy of the DecisionTreeClassifier, call the *decision_tree_classifier(filtered_data, max_depth)* method and pass in the filtered_data.
4. The next step is to train the neural networking model, however before doing this we created a method that helps us get the best hyperparameters. You can call this method by *tune_hyperparameters(filtered_data)*. This will print out the different learning rates and hidden layer sizes with the training and test set score. Through this data, we have already determined that the best hyperparameter is 150, 150, 150 with a default learning rate.
5. Finally, to get the training, testing, and accuracy score of the MLPClassifier model, call *mlp_classifier(filtered_data)*. This should print out all the scores.

Result

Spatial Analysis for Q1

According to Figure 1 and Figure 2, we can see the bike trip are densely located near the downtown area and are near transit stations. It's reasonable that bike-share is a great tool for the first or last mile travel transportation mode for commuters. However, even though I want to conclude that bike trips are significantly correlated with transit stations, it's still necessary to consider the multicollinearity between transit station and other factors such as neighborhoods. For example, transit stations are built in popular neighborhoods, so the reason lead to high bike frequency is neighborhoods instead of transit stations. Therefore, further investigation on other factor is required in the future study.

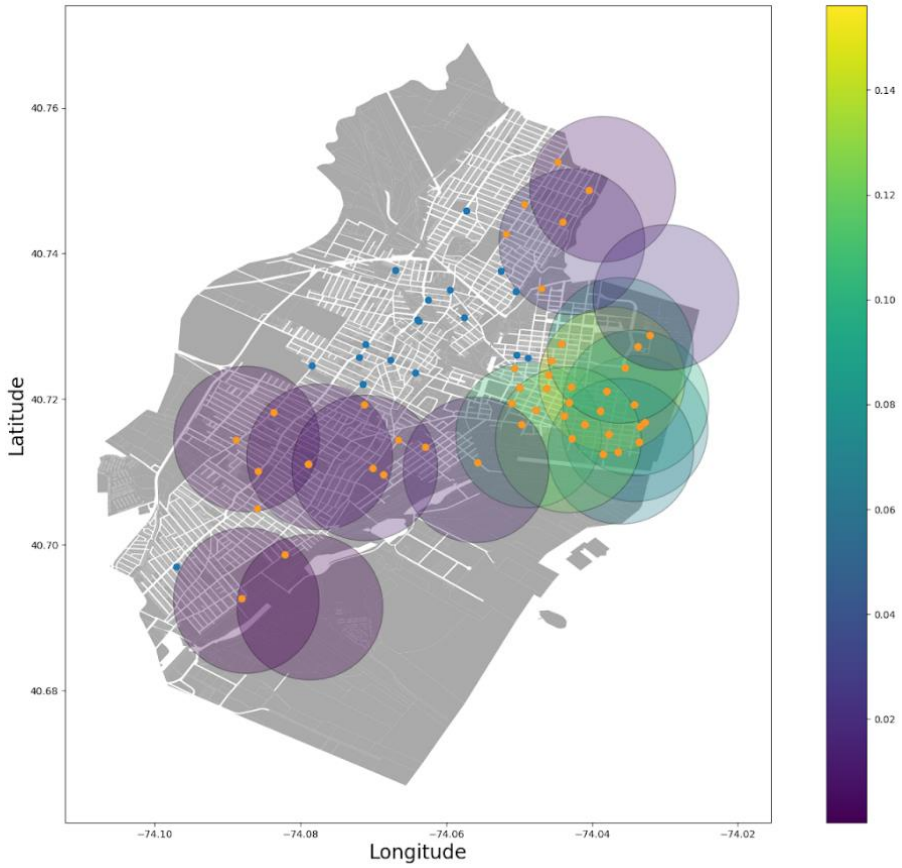

Figure 1

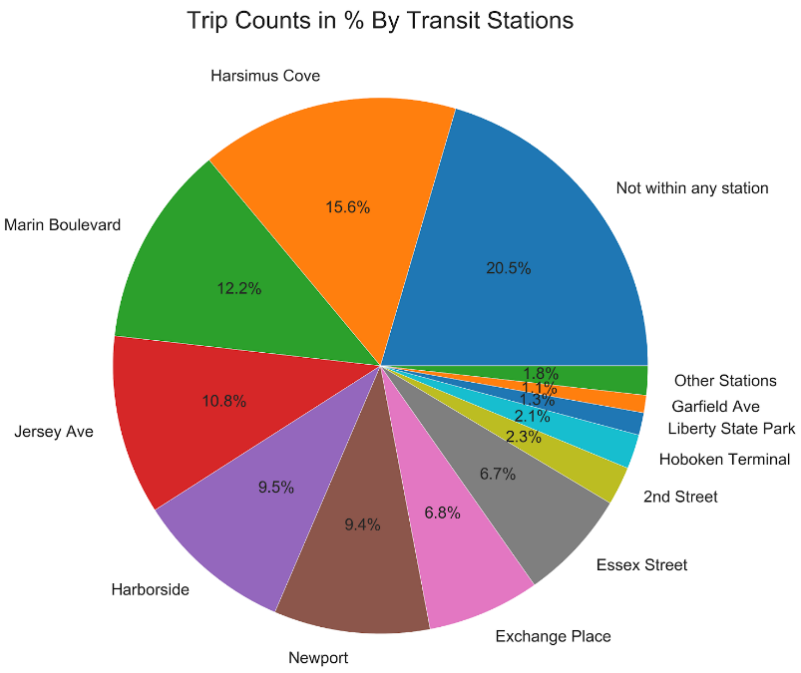

Figure 2

Graphs for Q2

By visualizing the bar plots shown in Figure 3, we can tell the trends indicate the Citi Bike usage were gradually increasing. There are a lot more people prefer to use Citi Bike during summer and autumn. This can infer the influence of temperature and seasonal weather conditions. Similarly, more people prefer to ride during morning and evening, inferring these are the time periods that people have most frequent transportation.

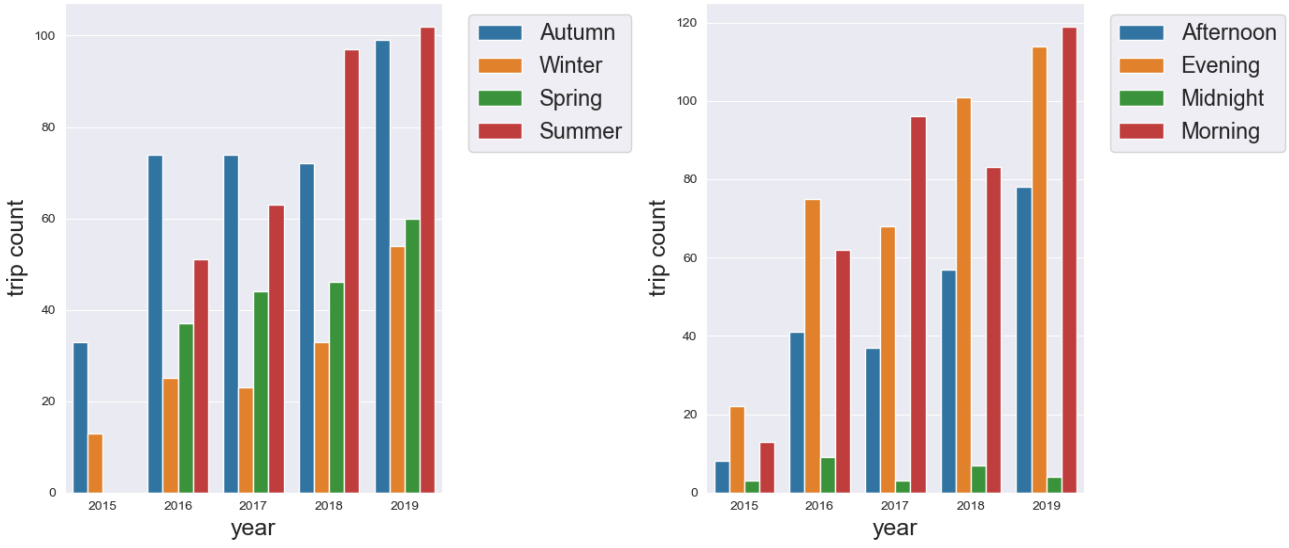

Figure 3

According to Figure 4, the trip duration for customers or unknown gender users are much longer than the other users. It's reasonable because of the difference of attitude. From the perspective of a subscriber, they will never want to exceed the threshold and pay the extra money for the extra time. Instead, they can re-login the bike every 45 mins so they can have unlimited times of 45 mins trips without extra fee. In contrast, from the perspective of a customer, they might be visitors that don't care about this money; might be unfamiliar will be charge standard; or they just need this tool for the day, so they don't care about the payment.

Moreover, customer and unknown usertype can be self-correlated, meaning that most of customers are not required to fill up as much information as the subscribers. As we can see in these two plots, both unknown and customer lines start from 2017, inferring Citi Bike might begin to promote their bikes to customers at that time.

By looking at figure 5, several interesting things are discovered. The most interesting part is the highest average trip durations always appear at midnight. Although the bike counts at midnight are little, it's true that there are multiple people bike over an hour currently period. Maybe they are the Citi bike fanatics. On the other hand, older people tend to ride in afternoon and morning and least likely to be at evening. It seems like the older you are, the more likely you are going to have a healthy habit, waking up early and sleeping early, even though this might a stereotype. Moreover, the age of most Citi Bike users is between 30 ~ 40 years old, indicating most of the bike-share users are adult and likely to be commuters.

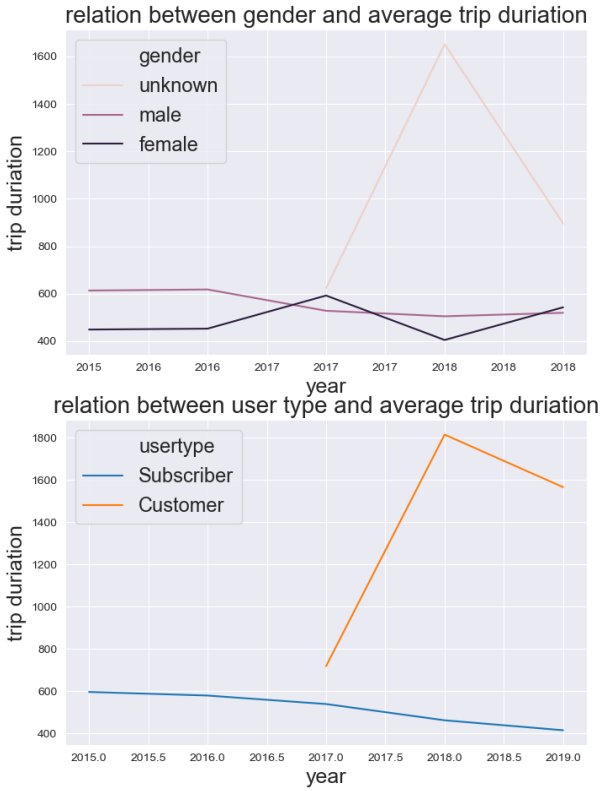

Figure 4

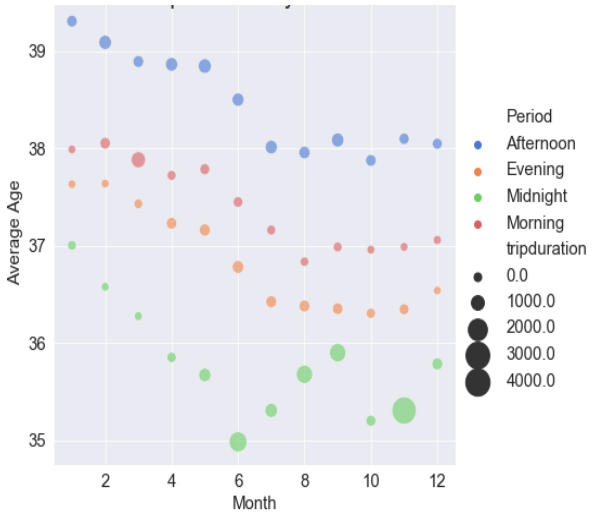

Figure 5

Modeling for Q3 & Q4

After training the decision tree classifier model and calculating the best max depth, the score accuracy came out to be surprisingly 37.3%. In our next model, the MLP had a scoring accuracy of 31.6%, a training set score of 31.8% and a test set score of 31.6%. Surprisingly these scoring results are hovering around 35% - a lower number than what we were expecting. We manipulated column features, hyperparameters, increasing the number of data we had to try and increase scoring accuracy, however, in the end, these changes made no significant change to our number.

Taking our scores and plot into consideration, there may be a couple of factors as to why this number is low. One being that these features don't provide enough information for the model - meaning that there could be other factors that play a significant role in determining user destination, factors that are not included in our features. Another factor as to why our score is low could be inherently from the problem itself. Our TA discussed a possible problem that predicting the specific behavior of an individual is too complex for such a model. The next time we tackle and develop this problem I would make sure to up the number of features and significant features to the model.

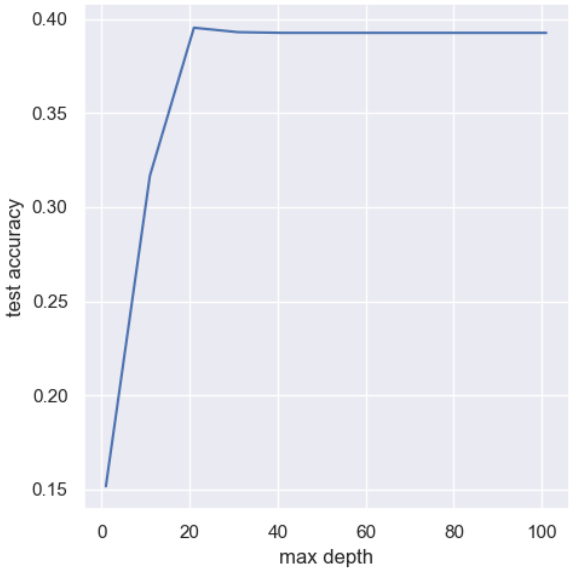

Figure 6

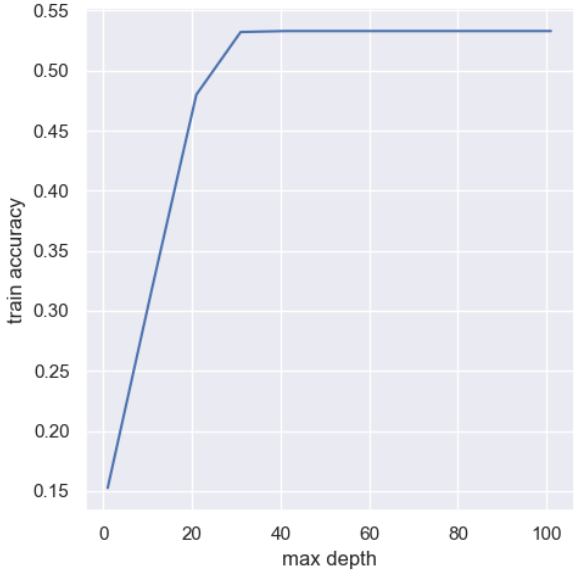

Figure 7

Work Plan Evaluation

We are following the work plan in Part 1, starting question 1 first and then question 2. Different persons are working on these two questions simultaneously, and we turn to question 3 after we finished question 1. We start working on the report 5 days before the due date and that's earlier than planned. But we didn't finish all the coding and debugging when we begin to work on the report, we are working on these two at the same time. In the sense of beginning report early, it is better than planned but question 2 took longer than expected, since customizing the plots turns out to be harder than we think.

Testing

In order to test the spatial analysis function, we first made create a sample that contains 10000 randomly selected data from the Citi Bike data. Then, we also read the other necessary files. Import the *assert_equal* function and check three things: whether the sum of bike trips in the conclusion table is equal to the length of the sample data (total bike frequency) and whether the sum of the % trip and % trip duration are equal to 1.0.

For the plotting questions, we had a 10000 sample randomly selected from the dataset, because we need to run the code for many times to debug, so reading thousands of rows each time can consume a long time, that's why we use *pandas.sample()* to get a smaller dataset for test. The samples are randomly selected each time, so the result is slightly different each time. Although every result is different, the overall tendency is the same, which proves the reliability of the random results.

For the machine learning model, we first tested using the same 1000 randomly selected samples. However, results weren't as expected and so we decided to use the entire dataset, a million rows of data from the years 2015 to 2019. For the Decision Tree Classifier model, we first plotted its accuracy in order to test our data. This would give a better representation of the model in case it becomes overfitted. The time it takes to run this an hour long if ran from 1 to 100 depth. Instead, we made the range from 1 to 100 counting by tens. Next, we tested the MLPClassifier by running a separate method that tunes its hyperparameters for us. We wanted to make the model as accurate as we could, so we made the learning rate [0.01, 0.5, 1.0] and hidden layer sizes range from 10, to 150. This part takes a while to run, however, it is completely reasonable as it is running million rows of data repeatedly. In the end, we got the best training, test, and accuracy score with the hidden layer size of 150, 150, 150.

Collaboration

Ray works on data cleaning, spatial analysis, and scatter plot. Runbo works on data collection, line plots, and bar plots. Danhiel works on machine learning. Since all the codes are posted on github, we all double check the code for each other and ensure everyone's code is running perfectly.