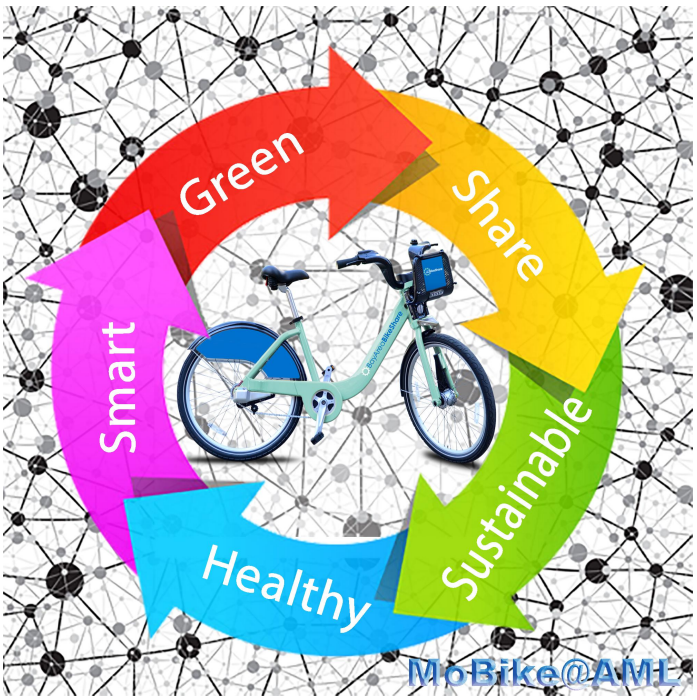
INFO 251 Project Report

MoBike

Bay Area Bike Share Lifecycle Prediction



Anna Cho, Fred Fan, Leon Li, Jolly Liang

Dec. 2017

Abstract: Bike share programs provide an environmentally friendly and healthy alternative to commuting within and exploring a city. However, Ford GoBike currently expends extra energy in order to consistently check on the condition of their bikes. To determine whether a bike is in need of repair, they depend on customer feedback or must send employees on frequent rounds to check up on bikes at each station. In this paper, we applied machine learning models to help For GoBike predict the total length of the lifetime for bikes within the San Francisco Bay Area. We defined the lifecycle of a bike to be the total duration for each of its trips before the bike was repaired or recycled. We determined this definition of lifecycle by using K-Means and Probability Distribution methods. We found the best intervals of inactivity between lifecycles (7 days and 4 days when parking station is changed) to group trips into distinct lifecycles. Data analysis was then conducted to study the relationship between lifecycle and potential features, including trip count, subscriber ratio, temperature, education level, etc. After that, we applied and compared Linear Regression with Lasso Regularization, Decision Tree, and Random Forest Models to predict the lifecycle of bikes based on features of trips, weather, income, education, and crime data. Through error analysis using the error percentage (defined in section 7) of the Linear Regression Model, we found and excluded outliers within the data. With the new trimmed dataset, we trained Decision Tree and Random Forest Models, and reached a better error percentage of 20%.

Keywords: Bike Share, Lifecycle, Machine Learning, Linear Regression, Decision Tree, Random Forest

Table of Content

1 Problem Statement

1.1 Background

Bike sharing systems are getting more and more popular in urban life, especially in big cities. Their automated features and convenient access have made them an important supplement to public transportation. However, the damaged bicycles which are still parked in stations may hinder the efficiency of the whole bike sharing system. Currently, it is technically hard for the bike sharing companies to determine when they should rotate their bicycles out of the system for maintenance. The only practical way to find out is to check regularly. Thus a forecast of bike lifetime would be of great help to this issue and would enable the bike sharing companies to plan their inventory budget accordingly. On top of that, it would also help ensure the hardware quality and thus reduce hardware risks to bike users.

1.2 Related work

There have been a number of studies conducted using the San Francisco bike share data to predict bike deficits at stations, trip duration, and other information about individual bike trips. The most popular kernel for the dataset on Kaggle is a model that predicts how many bike trips will occur on any given day using only known data from the morning of that particular day, such as weather, number of available bikes, and whether or not it is a business day[1]. Wang et. al. conducted a different study using Minnesota bike share data to understand how nearby businesses and access to jobs correlate with bike share usage[5]. Trip duration is another popular target variable using this dataset. One popular kernel on Kaggle uses correlations between available features, such as day of the week, year, season, and city (San Francisco vs. Palo Alto), to determine the duration of a bike trip[2]. This is accomplished using a linear regression model. Another group predicted the bike station destinations and arrival times of bike trips[4]. In order to predict these target variables, they used features about the users (subscriber status, gender), features about the department time (time, day), and features about the stations (location). To determine potential destinations, they used regression trees as their model. For trip duration, they used a lasso regression model. The goal of studies that explore individual trips like the ones above is often to help the bike share company predict surpluses and deficits at bike stations so that they may better optimize the stocking of their bikes.

Other studies are interested in determining how bike shares affect other activities in their cities. For example, the UCI Machine Learning Repository[3] proposes that bike sharing data can be treated like a virtual sensor network that can be used for understanding mobility within a city. This system of bike sharing can be used to detect events within the city. Another study explored how bike sharing affects car usage[6]. Rather than predicting a target variable, these studies look at the relationship between bike usage and other events within the city.

Instead of studying other events within the city or individual bike trips, our team will predict each bike's lifetime. We will see which features are most important in predicting the total number of

minutes a bike is used before it is taken out of rotation from the bike share program. In our research thus far, we haven't found a study that has implemented a bike lifetime prediction model. The feature importance that we obtain from this model will be new information that will help bike share companies better understand how to maintain their bikes so that they may optimize each individual bike's usage. It may also become another feature that will help make the models in these other studies more accurate.

1.3 Motivations

In order to ensure each bike is being used to its maximum potential, the bike share company needs to quickly detect when a bike is in need of repair so that it is not just sitting unused at a station. Currently, Ford GoBike does not have an efficient system for checking up on its bikes. After speaking with a GoBike employee, we learned that there is a button on each bike for customers to notify the company that a bike is broken. In addition to this button, the employees will frequently go on rounds to each station to check on the statuses of the bikes. This takes a lot of time and unnecessary energy. We aim to predict the actual times that a bike becomes in need of repair so that the employees can check stations at targeted times.

2. Objectives

2.1 Research questions

- How do external factors, namely, weather, crime, education, and income, affect the lifetime of a bike?
- Can we use trip, weather, crime, education, and income data to predict the length of a bike's active duration (i.e. lifecycle)?

2.2 Expected outcomes

We expected to see some visible patterns in the relationship between bike lifetimes and weather and crime data since bikes might fare worse in poor weather or suffer from vandalism in areas with high crime. Similarly, we thought that education and income might affect bike usage and therefore bike lifecycles. Our ultimate goal was to use all of our features to predict the time when a bike would need to be checked on for maintenance.

3 Dataset

The main dataset we used is provided by Kaggle, which is in fact a transformed version of published data by Bay Area Bike Share[7]. The data contains four parts, including station, status, trip, and weather data. The station data is for bike stations at which users can pickup or return

their bikes. The status section is the number of bikes and docks available for a given station and minute. The trip section represents an individual bike trip. And weather represents the weather for a specific day and zip code.

We also incorporated external weather, income, education and crime data. The weather data is provided by Weather Underground[8]. Compared with the weather data by Kaggle, it provides more detailed hourly weather data with which we can use to correlate with each trip to get more precise relationship. Besides that, we also added income, education, crime data for each bike stations to explore how those data can influence the trip activity.

The crime data is provided by the San Francisco Police Department[9], Santa Clara County Police Department[10], Redwood City Police Department[11], and Palo Alto Police Department[12]. We used this data to count how many crimes occurred in the same zip code locations as the bike stations. The education[12] and income[14] data is provided by United States Census Bureau, which has a detailed education and income reports.

4 Assumptions

Our project is based on the following assumptions.

- The lifecycle of a bike is the sum of all trip duration minutes before the bike is taken in for maintenance or recycled.
- For each trip, the duration is the time spent traveling and doesn't include the time bikes are parked at stations.
- Once a bike has been repaired or recycled, it enters a new lifecycle.
- Users always make a direct trip from an economical perspective, rather than stopping by other stations before their end station.

5 Data Preprocessing

5.1 Defining a valid trip

After reviewing the trip information, we found that it contains several unreasonable trips, e.g., some trips are over 24 hours. Thus, we needed to perform some initial data cleaning.

Since we don't have an objective standard to judge which trip durations are reasonable or not, we used a statistical method to remove some of the strange trips.

We first checked the distribution of the trip duration (seconds) as shown below.

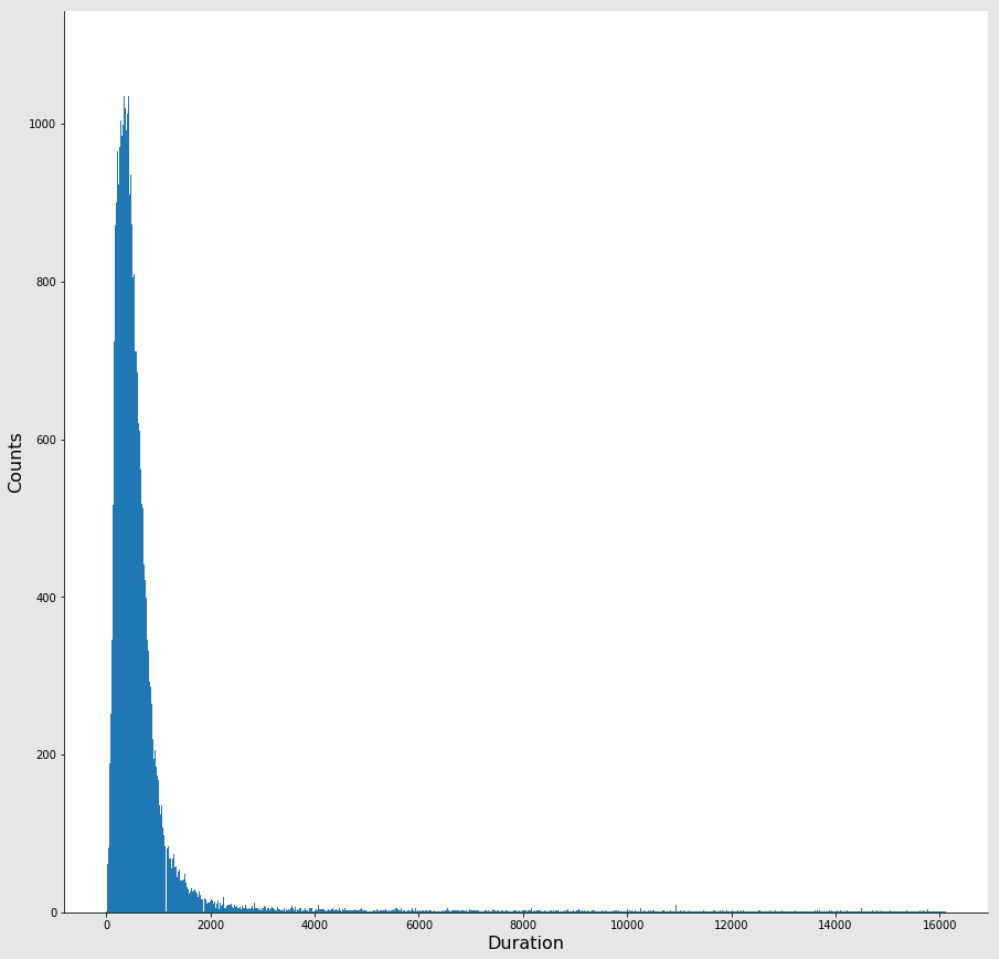


Fig. 5.1-1. Distribution of the trip duration

Since the distribution is positively skewed, we then transformed this data by log 10 and got the following distribution:

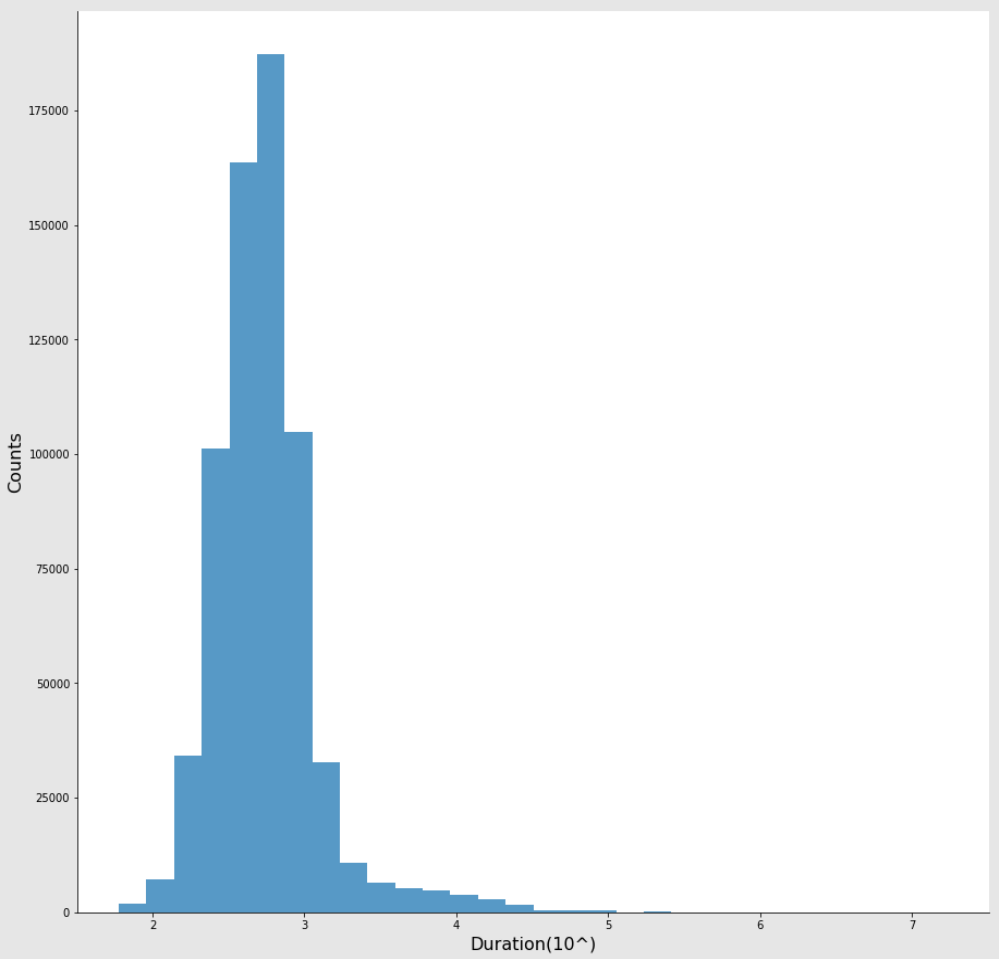


Fig. 5.1-2. Distribution of the trip duration (log10 transformed)

We then calculated the mean and the standard deviation. To keep as much data as possible, we found the approximate upper threshold of trip durations to be the mean + 2* the standard deviation which would just remove approximately 2.5% data. As a result, we got the upper threshold, 4 hours. Meanwhile, we also chose the approximate lower threshold to be the mean - 2* the standard deviation which would just remove another 2.5% of the data. Based on this, the lower threshold for trip durations is 2 mins. Thus, we removed 5% of the trips with extreme values for trip duration and kept the remaining 95% data as valid trips.

In order to trim the dataset, we defined the range of valid trip durations to be from 2 mins to 4 hours.

5.2 Defining the interval between life cycles.

After reviewing the dataset, we found each bike has frequent trips for some period of time but then have an interval of inactivity for several days. Thus to define the lifecycle (the period of time a bike is used before it is brought in for repair or recycling), we needed to define a proper

interval to split the multiple lifecycles of each bike. These intervals would represent the time that the bike is being taken in for maintenance.

Since we failed to get an official or direct definition of a bike's lifecycle from the bike sharing company, we first tried K-means to cluster the trips of one bike based on its trip start & end times and used silhouette metrics to find the best K for each bike. We hoped that K-means could cluster the trips that happened continuously one after another and split trips by distinct intervals of inactivity. This method worked for several bikes. For example, we found the best K is 8 for the bike in Figure 5.2-1. The clusters (the small dots represents trips and the big dots represent cluster centroids) seem to show distinct groups of trips based on these intervals of inactivity (see plot below).

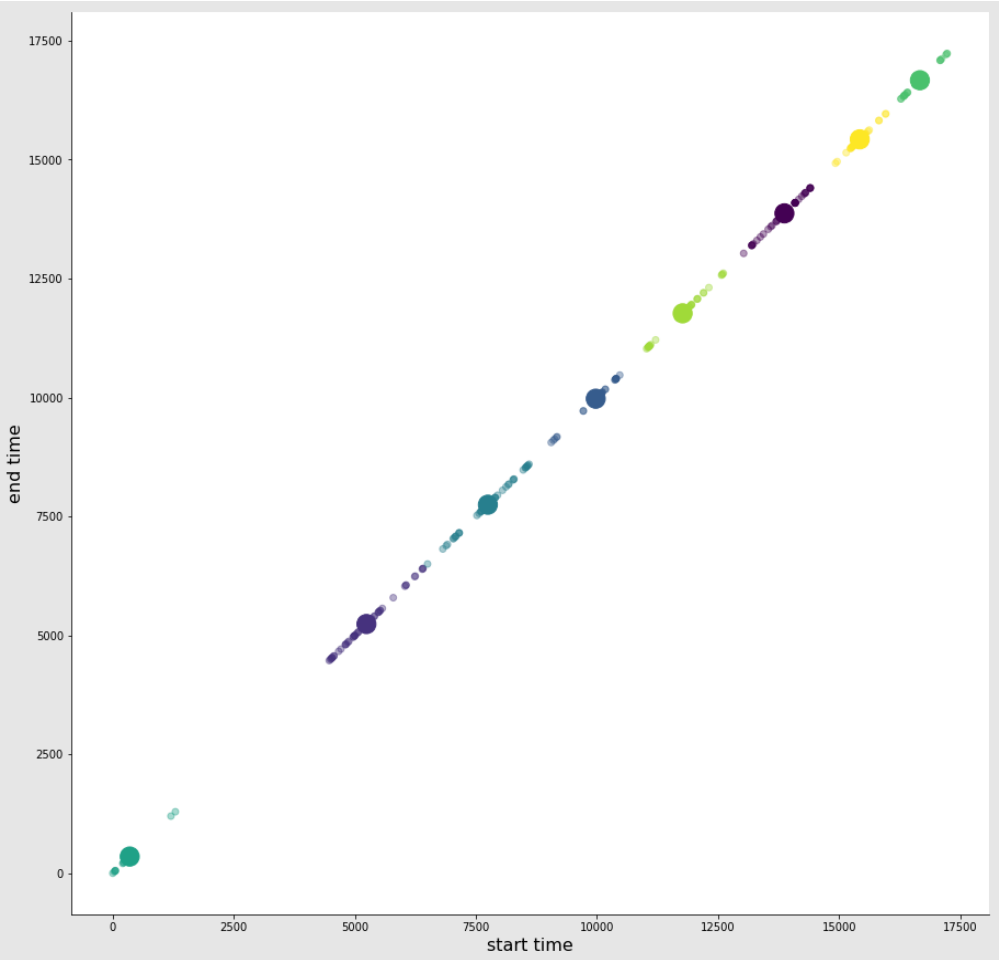


Fig. 5.2-1. Success example for KMeans

However, we also found K-means is not suitable for all bikes.

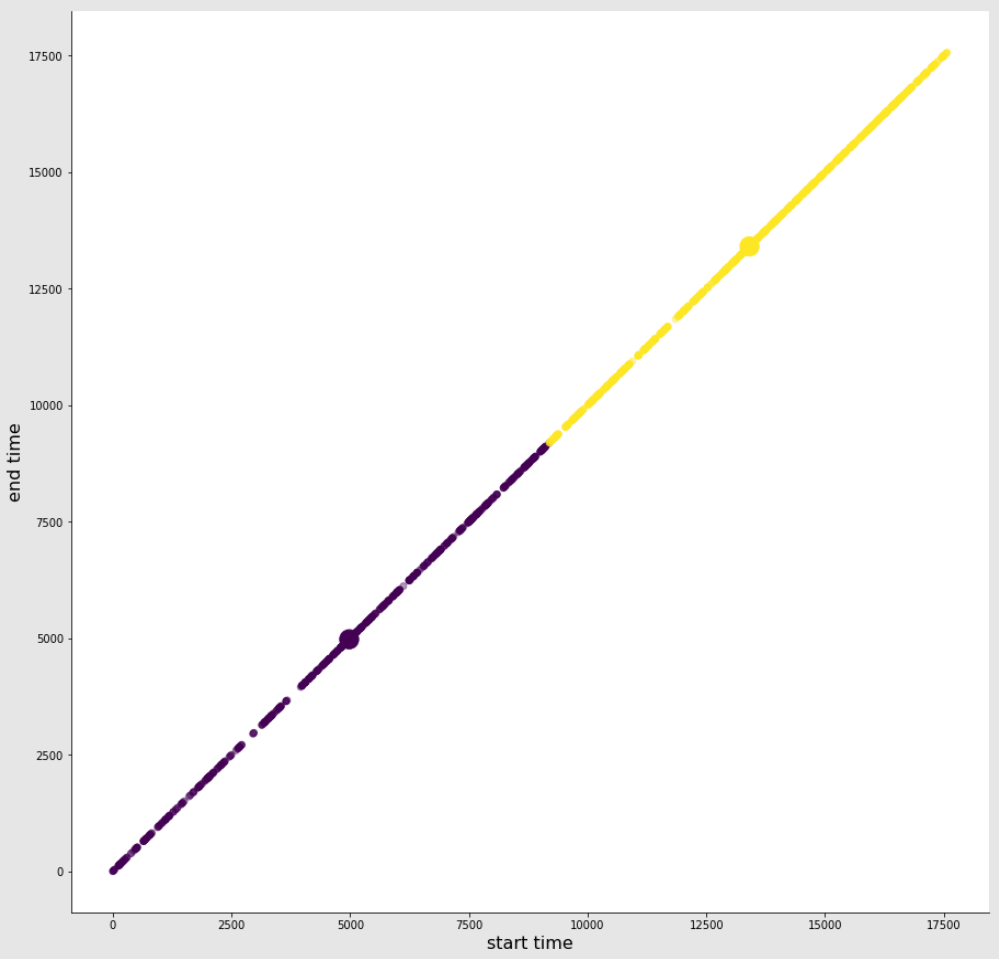


Fig. 5.2-2. Failure example for KMeans

In the example above, the best K is 2. However we failed to find a clear cut-off between the two lifecycles.

Thus, we began to use statistical methods to find a proper interval. We first plotted the histogram of the frequency of the intervals between trips as shown below. Since the distribution of the original intervals is positively skewed, pointy, and heavy-tailed, we then transformed the original data using the natural log and got a normal distribution as shown below:

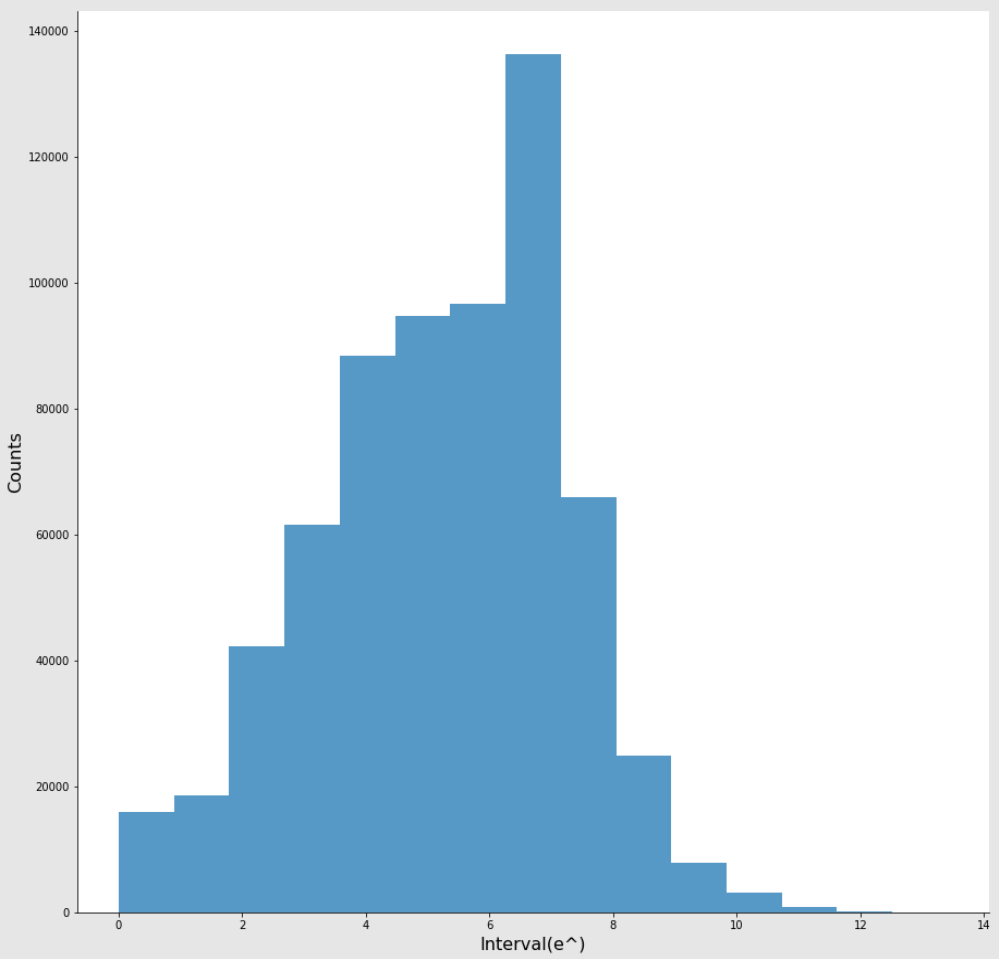


Fig. 5.2-3. Distribution of intervals (log transformed)

Based on the log transformed data, we got the mean and the standard deviation. We then got the “significantly different” interval (possibility < .05) by calculating the mean + 2*the standard deviation. The result was 7.3 days. Thus, we decided that if an interval is longer than 7 days, a new life cycle for that bike begins.

In addition, considering the reallocation mechanism of bike sharing to treat bike station surpluses and deficits, we used the “slightly different” interval (possibility < .1) to create another rule. If the interval is longer than 4 days (we got this by calculating the mean + 1.64*the standard deviation) and the end station of the last trip before this interval is different than the start station of the first trip after the interval, it is a new life cycle for that bike.

5.3 External Data

Since the original dataset only contains trip information, we gathered new data from external sources and engineered many new features, e.g., weather, crime, education. Please see appendix and references for details.

6 Data Analysis

6.1 Available Features

6.1.1 Lifecycle V.S. Subscription Ratio

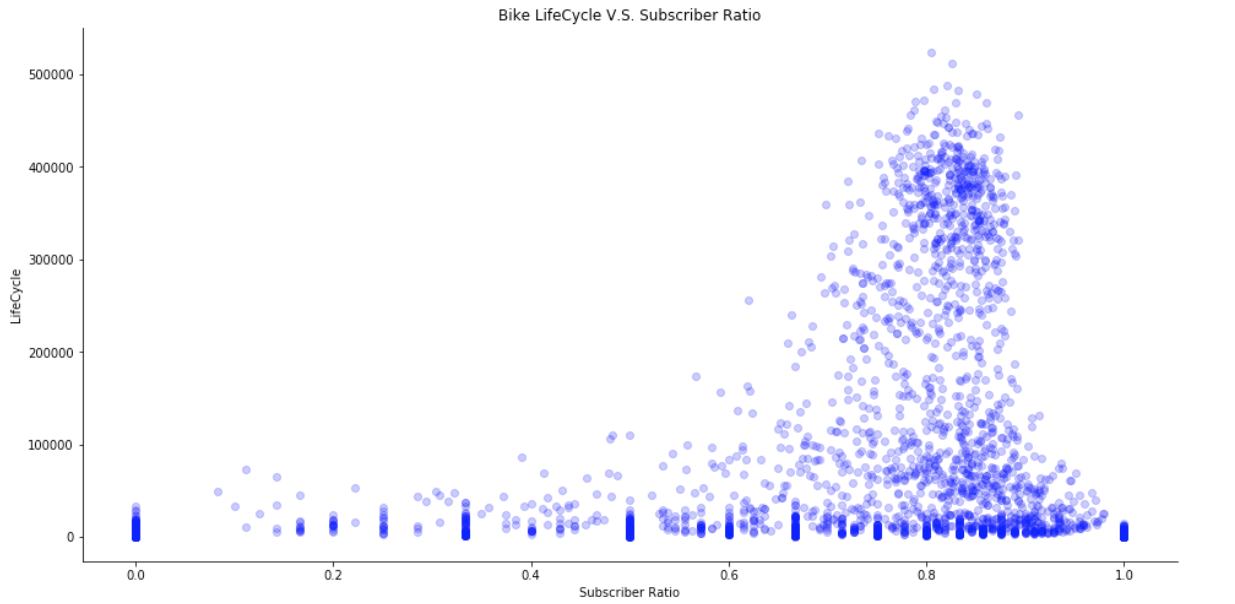


Fig. 6.1.1-1. Bike Lifecycle VS. Subscriber Ratio

We began our analysis starting from the relationship between lifecycle and subscription ratio. From the above figure, we notice that most of the data is more than 0.5 on the x-axis, which means that users use the bikes in subscription mode most. Basically, the higher the subscription ratio, the longer the lifecycle is. The relationship between those two variables seems to be exponential. And there is a drop off when the ratio is near 1.0.

6.1.2 Lifecycle V.S. Trips Number

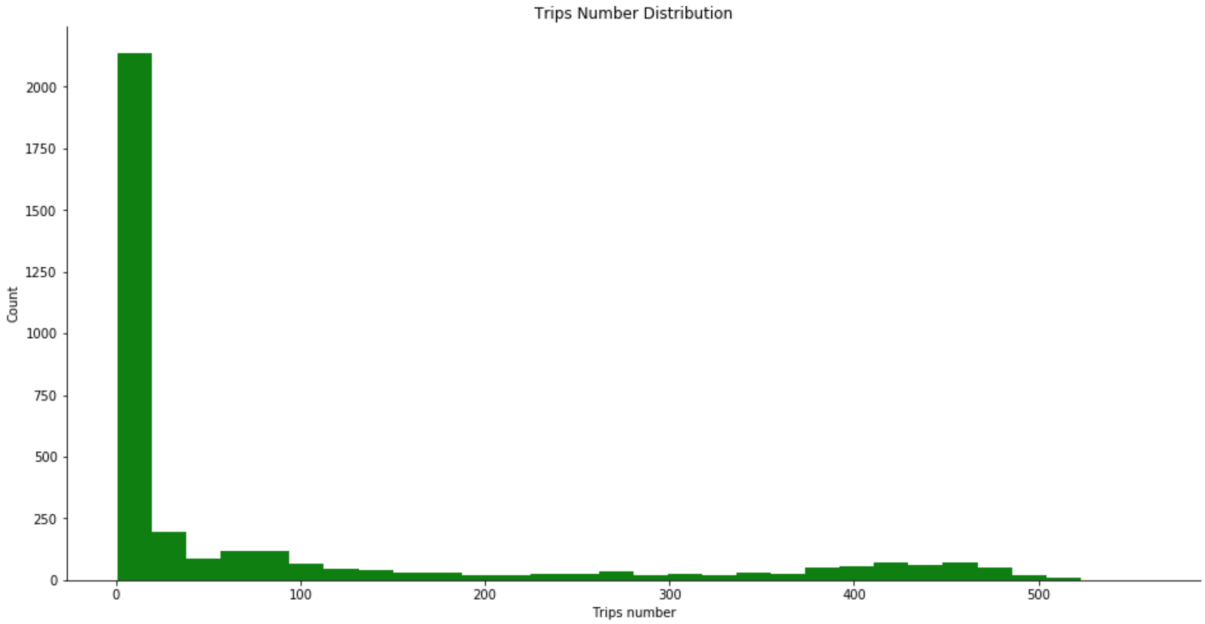


Fig. 6.1.2-1. Trips Count Distribution

The histogram above shows the distribution of the number of trips in all lifecycles. Most trip counts are below 100. As the trip count increases, the lifecycles decreases.

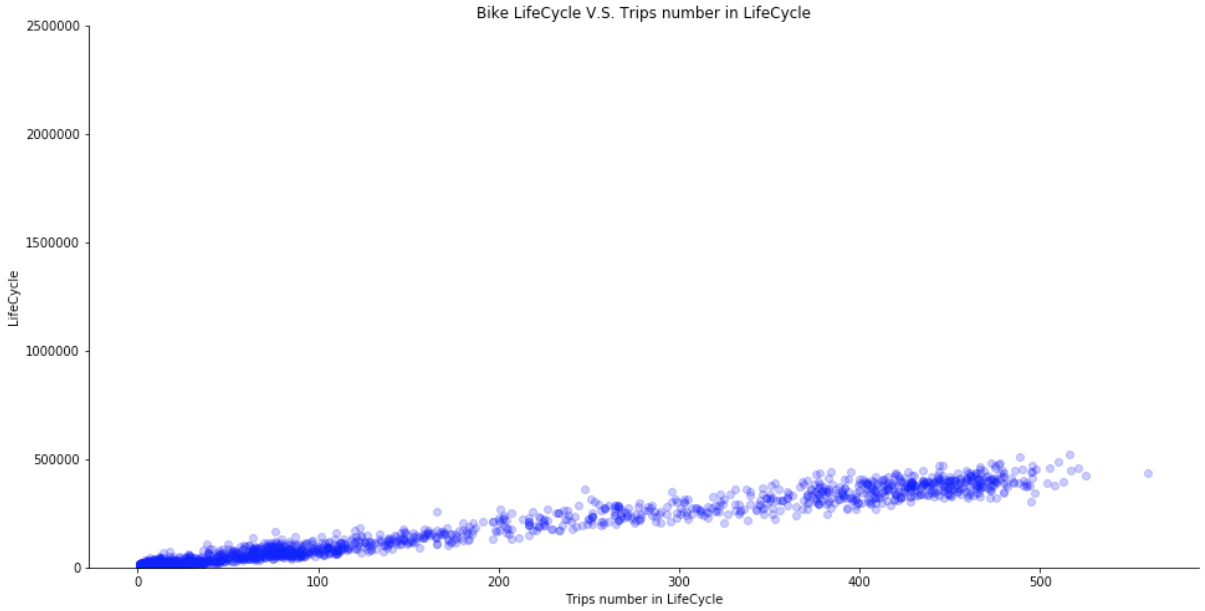


Fig. 6.1.2-2. Bike Lifecycle VS. Trip Count in Lifecycle

The figure above shows the relationship between the number of trips in a lifecycle and the total life cycle duration. The relationship is obvious: the more trips, the longer the life cycle is. The

increase of life cycle duration is linear with the increase of trip counts. As evident in the two figures above, most lifecycles have trips less than 100, and their durations are very short.

6.1.3 Station V.S. Lifecycle

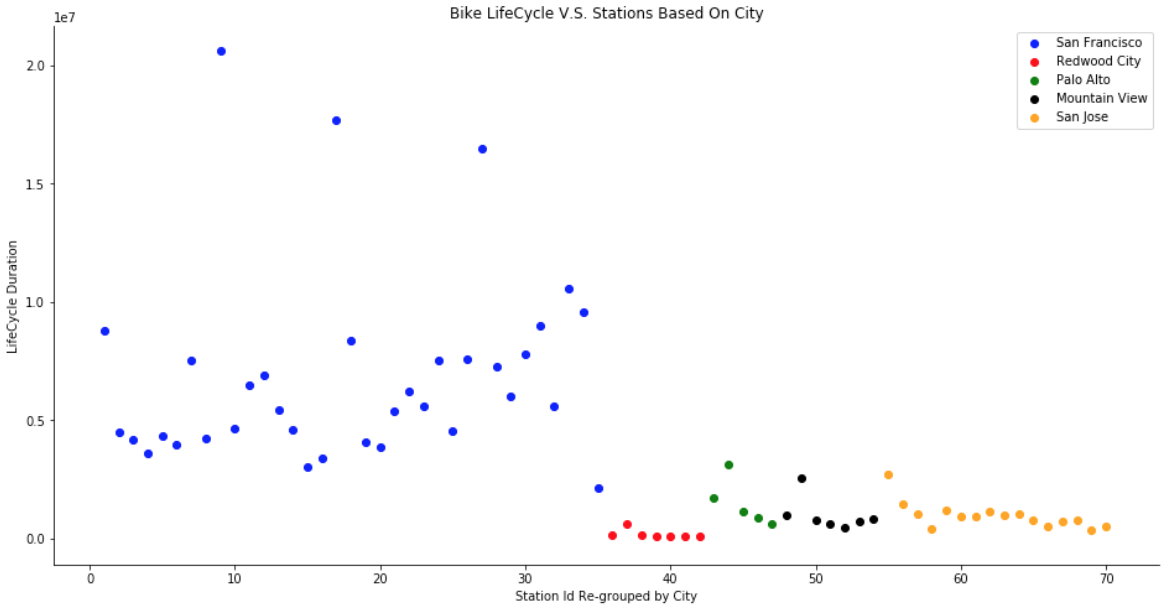
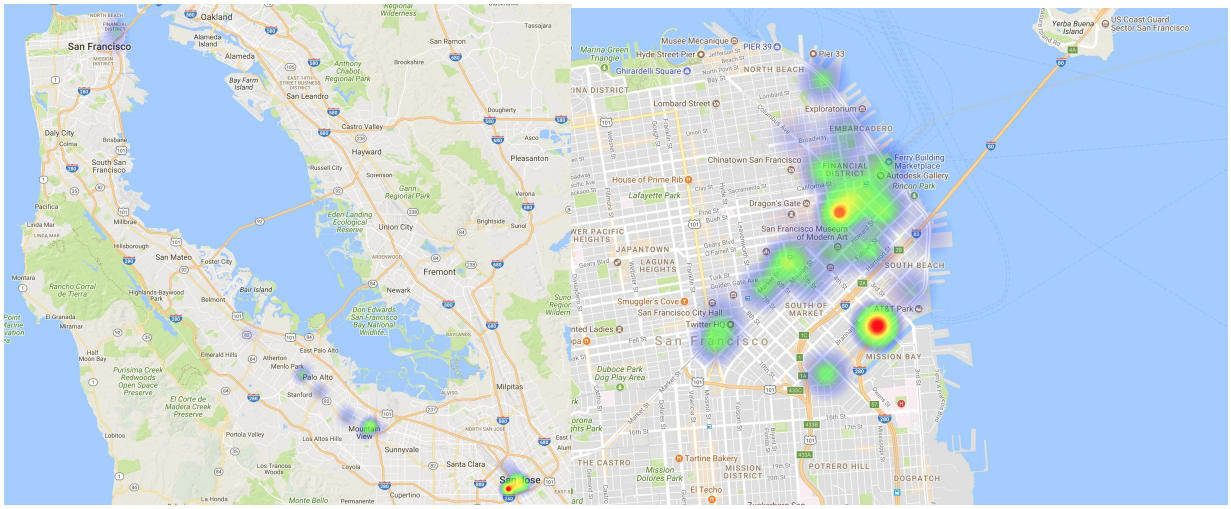


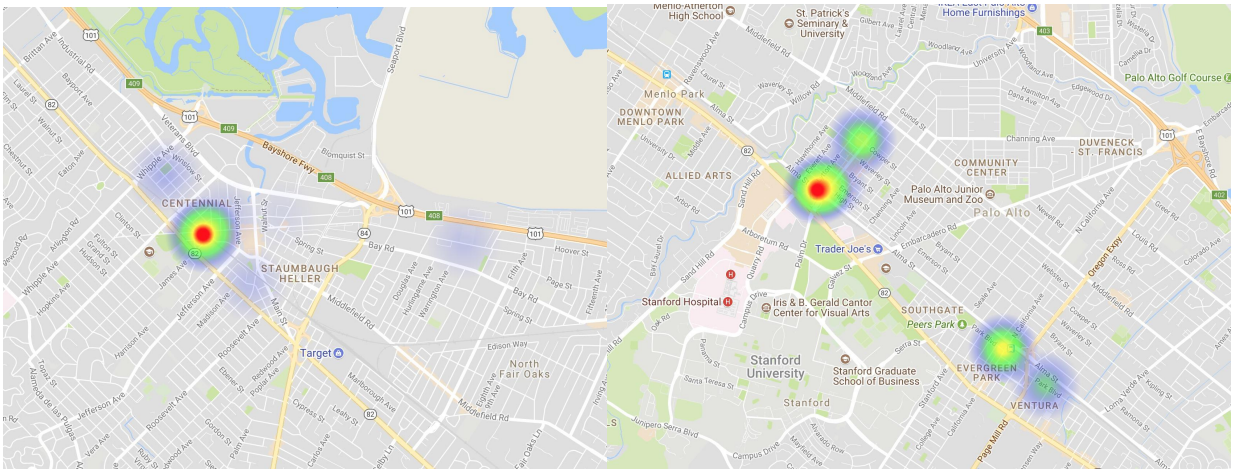
Fig. 6.1.3-1. Bike Lifecycle VS. Stations Based on City

We then looked at how stations can affect the duration of each bike's lifecycle. The figure above shows the relationship between each trip's start station and the lifecycle. We can see that San Francisco has the half of the total stations (35 out of 70). And it seems that bike usage is more active in San Francisco because the overall trips duration is larger than that of the rest of the cities.

Duration HeatMap

The heatmap below also shows the active status of each station. We can see that San Francisco is more green than other cities. And also San Francisco has two centers (two red points) compared with the rest of the cities. One is near AT&T Park, and the other is near the Museum of Modern Art. One possible reason for this is that the usage of bikes is related with the events held in those two places.





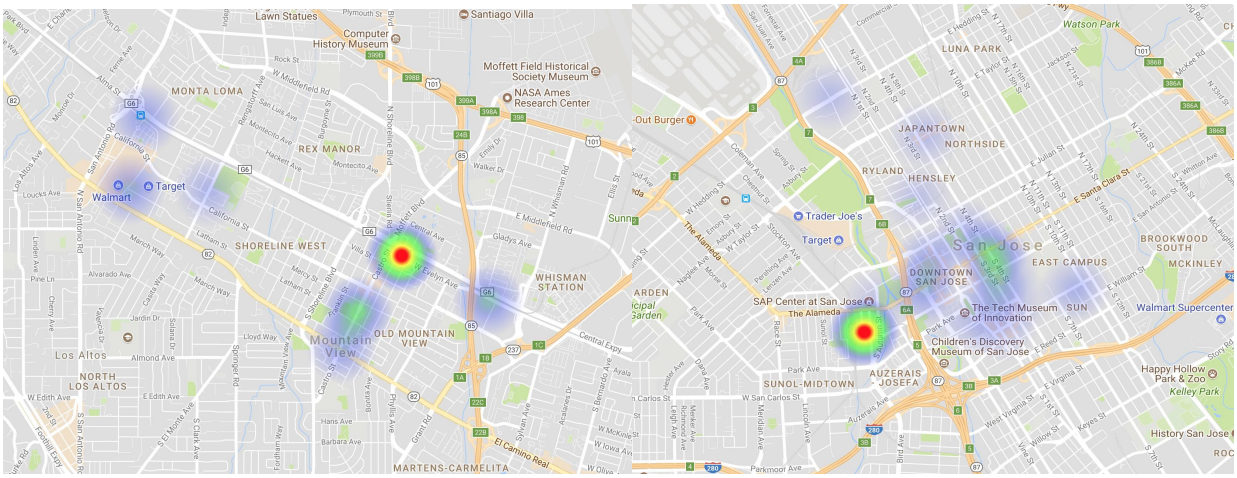


Fig. 6.1.3-2. Bike Activity Heatmap Based on Stations

6.1.4 Station V.S. Crime

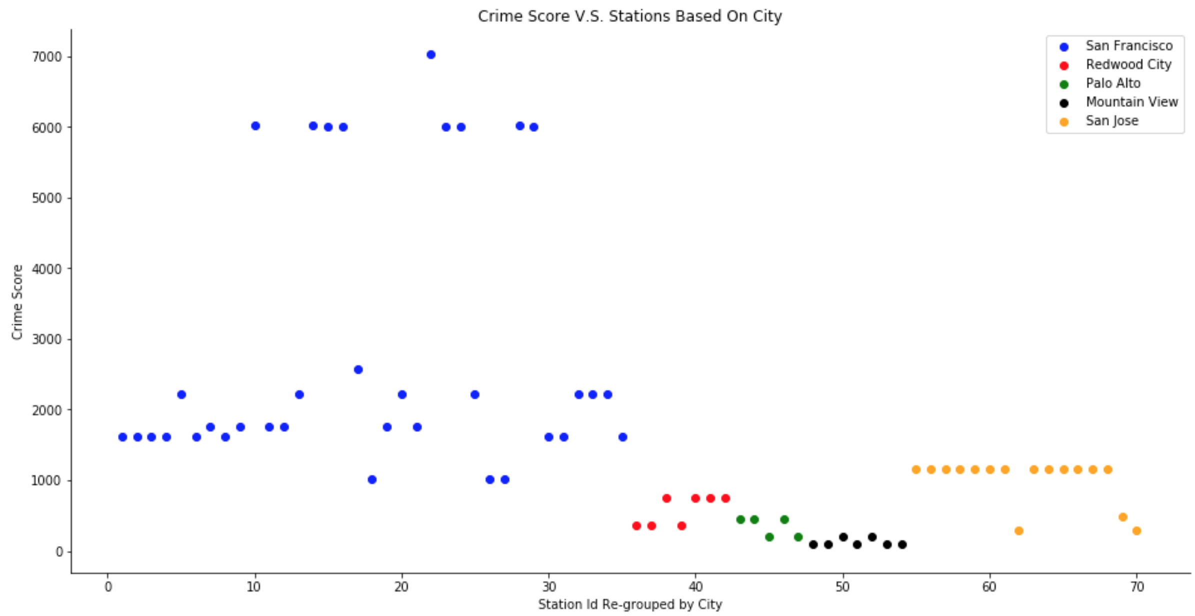


Fig. 6.1.4-1. Crime Score VS. Stations Based on City

The above figure shows the crime score for each station. The result is matches our intuition. For all the stations, those in San Francisco have the highest crime scores, followed by stations in San Jose, Redwood City, Palo Alto, and Mountain View.

Crime HeatMap

The crime heatmap below shows the crime scores for each station. If we compare the crime heatmap with the duration heatmap, we can see that the stations with longer durations have lower crime scores. For example, the most active stations in San Francisco in the duration heatmap are those near AT&T park and the Museum of Modern Art. But those places in the crime heatmap are the stations with lower crime scores compared with other stations in San Francisco.

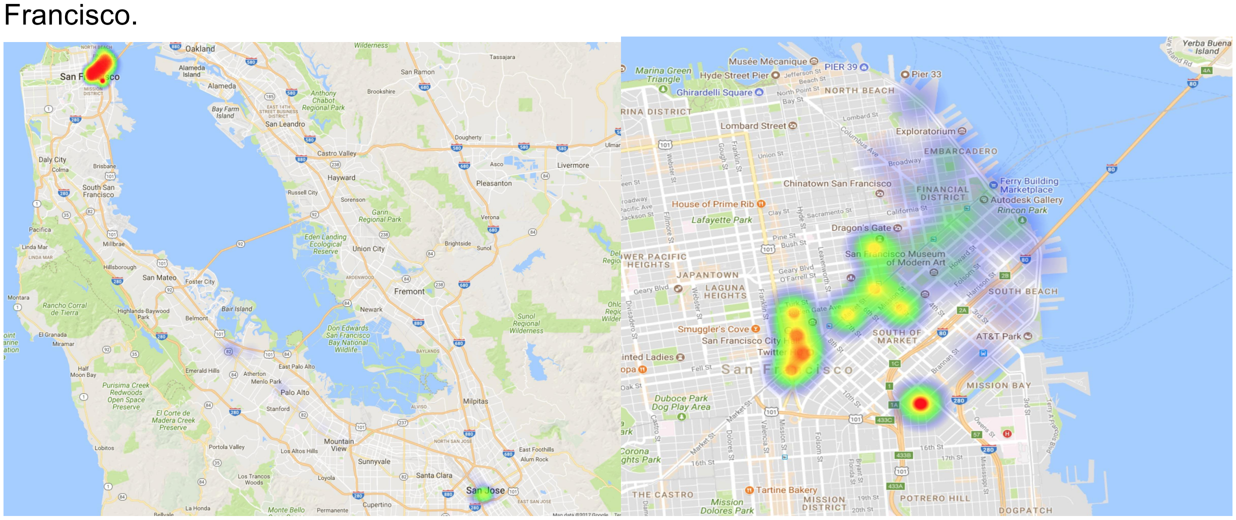


Fig. 6.1.4-2. Crime Score Heatmap Based on City

6.1.5 Lifecycle V.S. Period

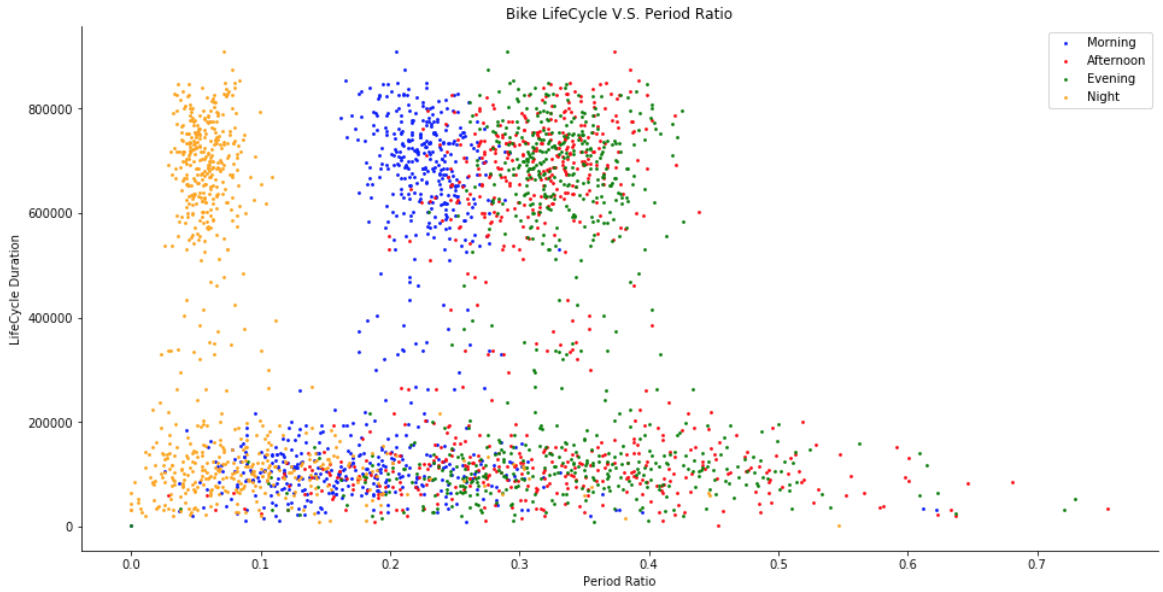


Fig. 6.1.5-1. Bike Lifecycle VS. Trip Period Ratio

The above figure reveals the relationship between the period of day a trip is taken and lifecycle. In order to do that, we cut the time of a day into four categories, including morning (8:00 AM - 12:00 PM), afternoon (12:00 PM - 16:00 PM), evening (16:00 PM - 20:00 PM), and night (20:00 PM - 8:00 AM). The period ratio is the percentage of each category among all the trips during a bike's lifecycle. From the figure above, we can see that, most trips happened in the afternoon and evening. And the basic trend is that when a specific period ratio increases, the lifecycle decreases.

6.1.6 Lifecycle vs. Parked Period

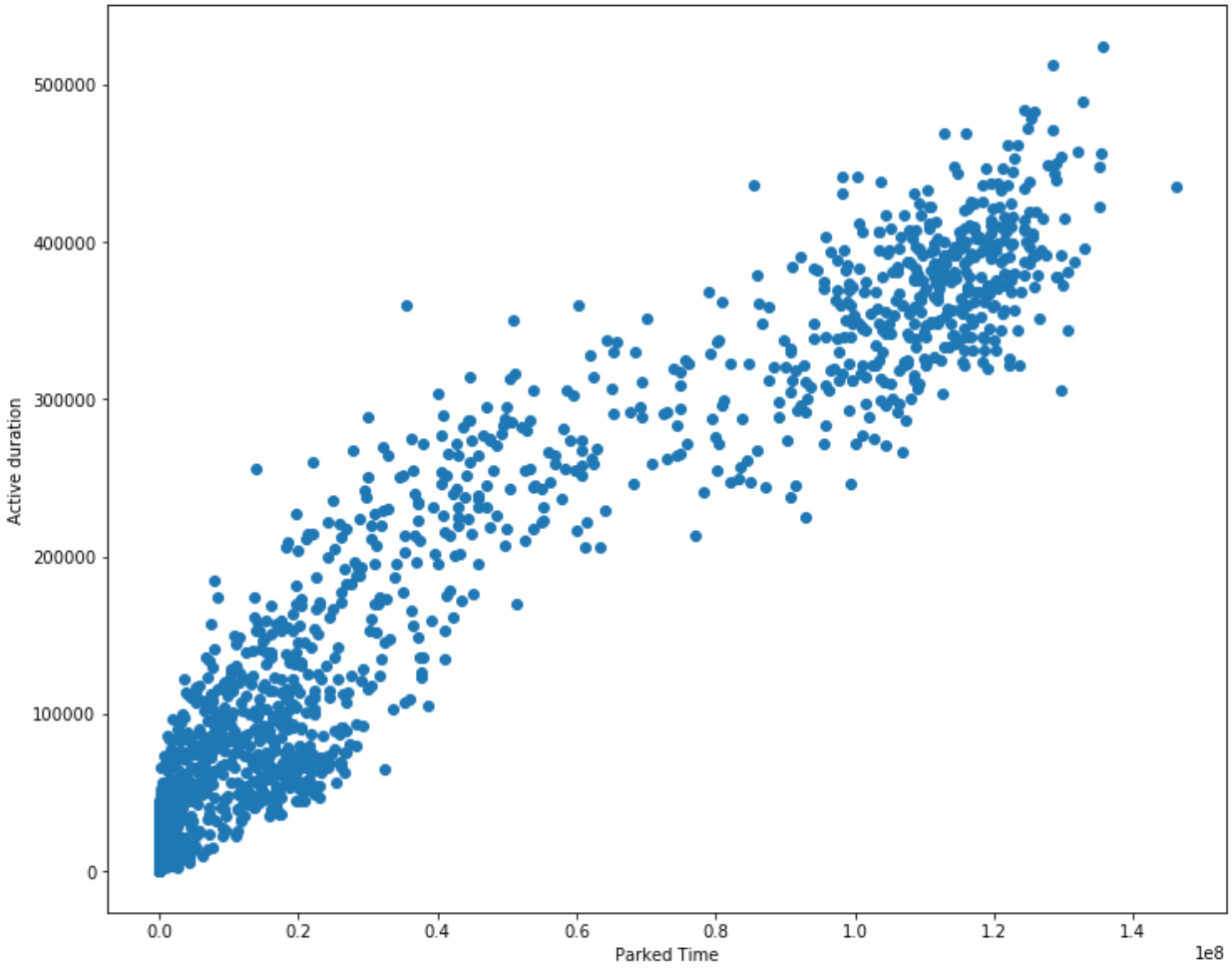


Fig. 6.1.6-1. Relationship Between Lifecycle (active duration) and Parked Time

The above figure shows the relationship between the bike's lifetime and the amount of time it was parked at a station during its lifetime. There does appear to be a linear pattern where the bike's lifetime increases with the amount of time it is parked.

6.2 Weather

To generalize the weather's impact on the life cycle, we gathered the hourly weather data from the 5 cities where our trips happened. Then, we focused on the temperature, dew point, pressure, humidity and visibility (5 weather factors).

To define the weather factors of a single trip, for each weather factor, we calculated the average of the weather factors at the start time and end time. We then multiplied the weather factors by each trip's duration and summed them up. Finally, we divided this sum by the duration of the life cycle and get the weather factors of a lifecycle.

Then we plotted the relationship between these factors and the lifecycle's duration. Two examples are shown below:

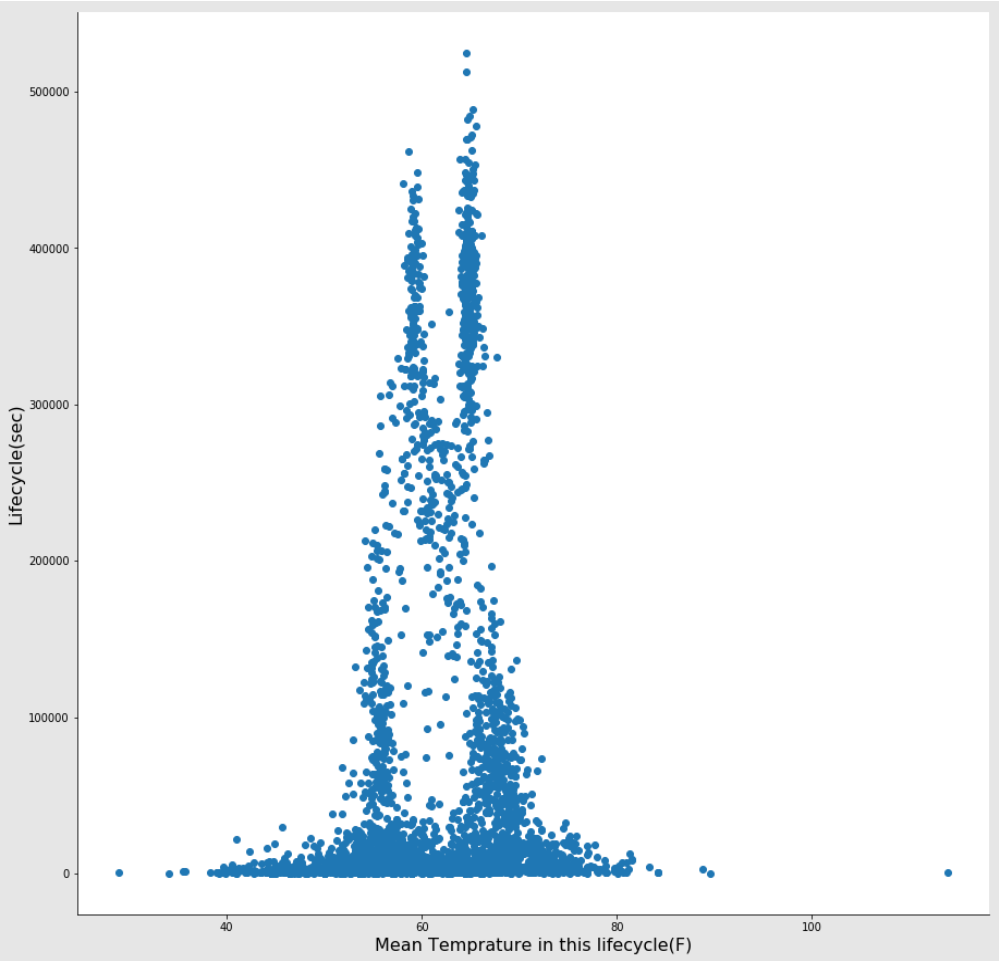


Fig. 6.2-1. Relationship between Temperature and Lifecycle

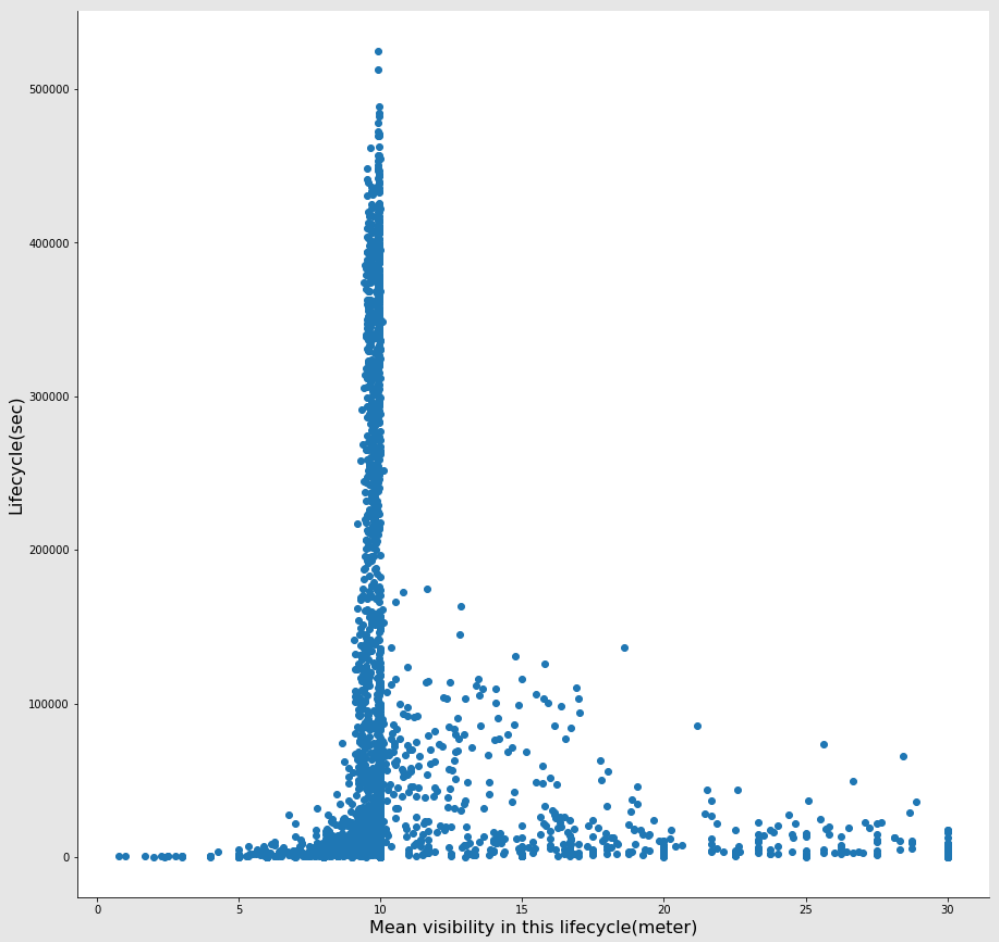


Fig. 6.2-2. Relationship between Visibility and Lifecycle

In general, besides Visibility, which has a normal distribution shape, other weather factors all have a bimodal shape (please see other plots in the ipython notebook). It is difficult to find a distinct linear relationship between a weather factor and the lifecycle. All we can determine is that long lifecycles do not occur when the weather factor is near its minimum value and its maximum value.

Furthermore, we used PCA to combine these factors excluding Visibility and reduced the dimensionality to get a “weather index” to represent the weather information. To maintain the variance, we decided to reduce the dimensionality to 2, which maintains 88% of the variance. Since we have 2 dimensions and 1 target, we generated a 3D plot to explore the relationship as below.

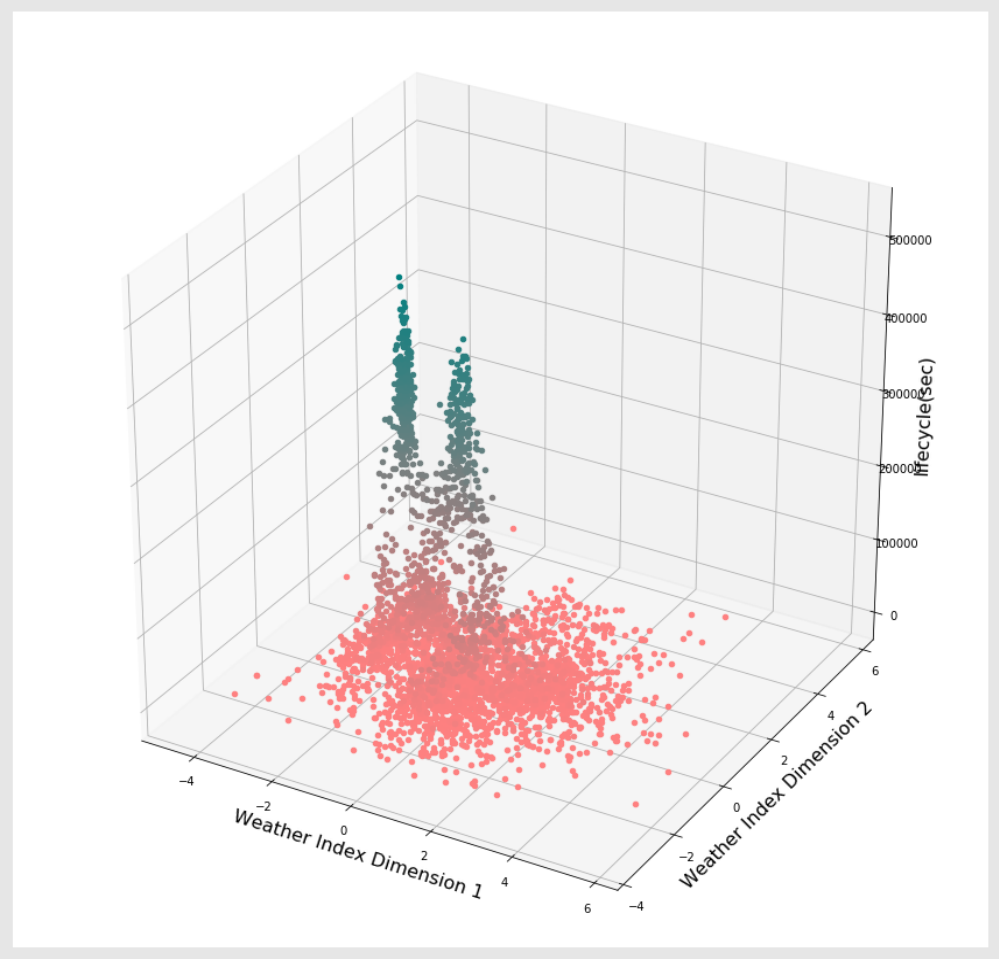


Fig. 6.2-2. Relationship between Weather Index and Lifecycle

According to the above plot, we could see long life cycles still remain at the center of the weather indexes. At the same time, we could not tell whether a short life cycle would occur based on weather conditions.

In sum, while weather factors could help determine whether it is possible to have a long life cycle, it is difficult to determine a detailed duration of the life cycle.

6.3 Economic Characteristics & Education

6.3.1 Overview of Lifecycle vs. Economic Characteristics & Education

Social and economic characteristics around the bike stations might also have a substantial effect on the bike pattern and lifecycle. After extensive research, we selected income-related data and educational attainment data **around the start station** of each trip to explore. To generalize the household income and education's impact on the life cycle, we gathered mean household income and educational attainment data from United States Census Bureau. After some comparison and analysis, we filtered the features to only include the three economic

characteristics most correlated to lifecycle and the two education attainment features most related to lifecycle.

With these 5 social and economic characteristics features, we first plotted a correlation matrix to see which of them are most related to the bike life cycle (duration).

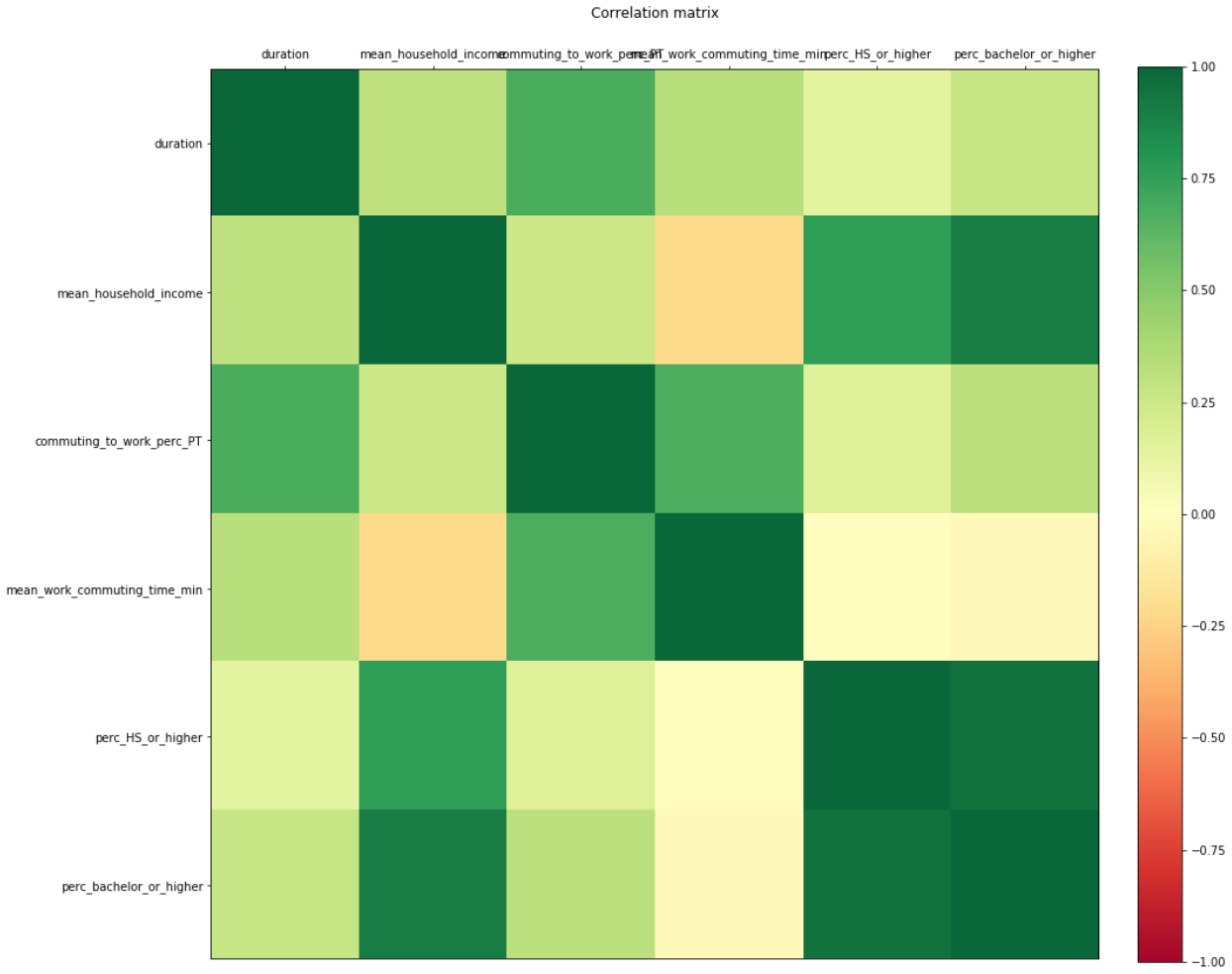


Fig. 6.3.1-1 Matrix among Income & Edu Features

From the correlation matrix plot above, we can see that among the three features from the income census, “Commuting to Work Percentage” (Public Transportation) has the highest correlation score (0.74). From the educational attainment census, “Percentage of Bachelor or Higher” has the higher correlation score (0.29).

6.3.2 Lifecycle vs. Mean Household Income

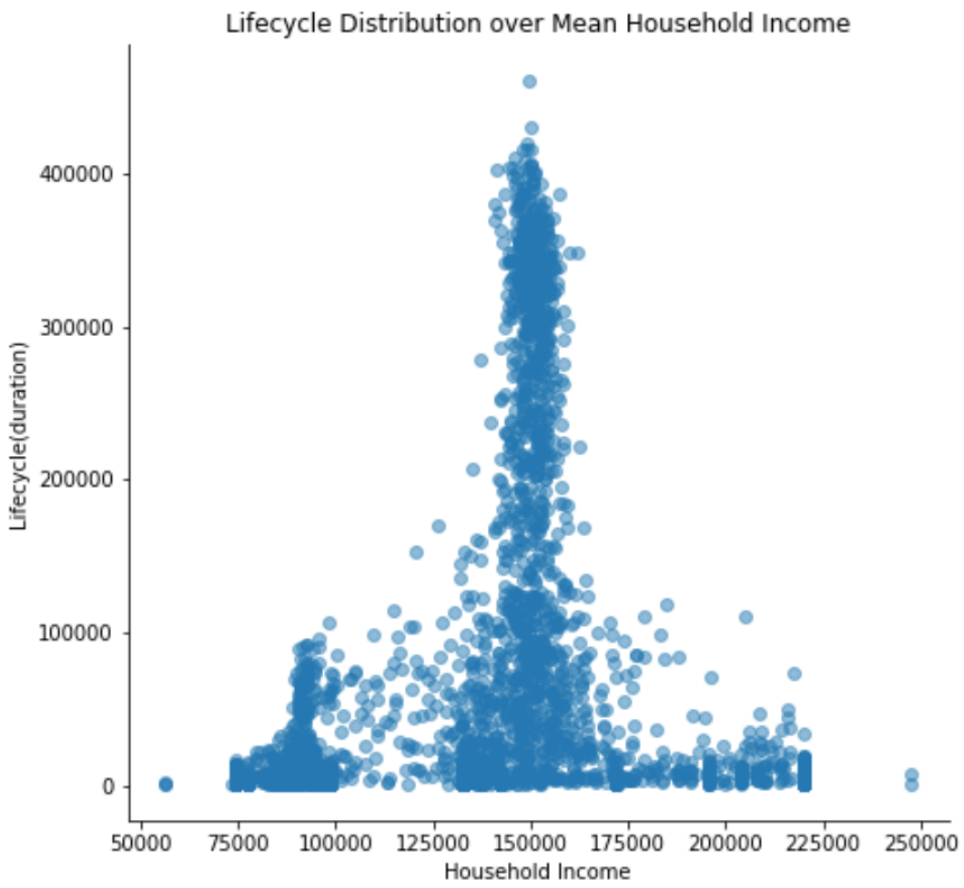


Fig. 6.3.2-1. Lifecycle vs Household Income

From the Lifecycle vs Household Income plot above, we can see that the bike life cycle spikes in areas of mean household incomes between 130K - 160K. For areas with mean household incomes between 75K - 100K, bike life cycles tend to be much shorter.

6.3.3 Lifecycle vs. Commuting-to-work Percentage

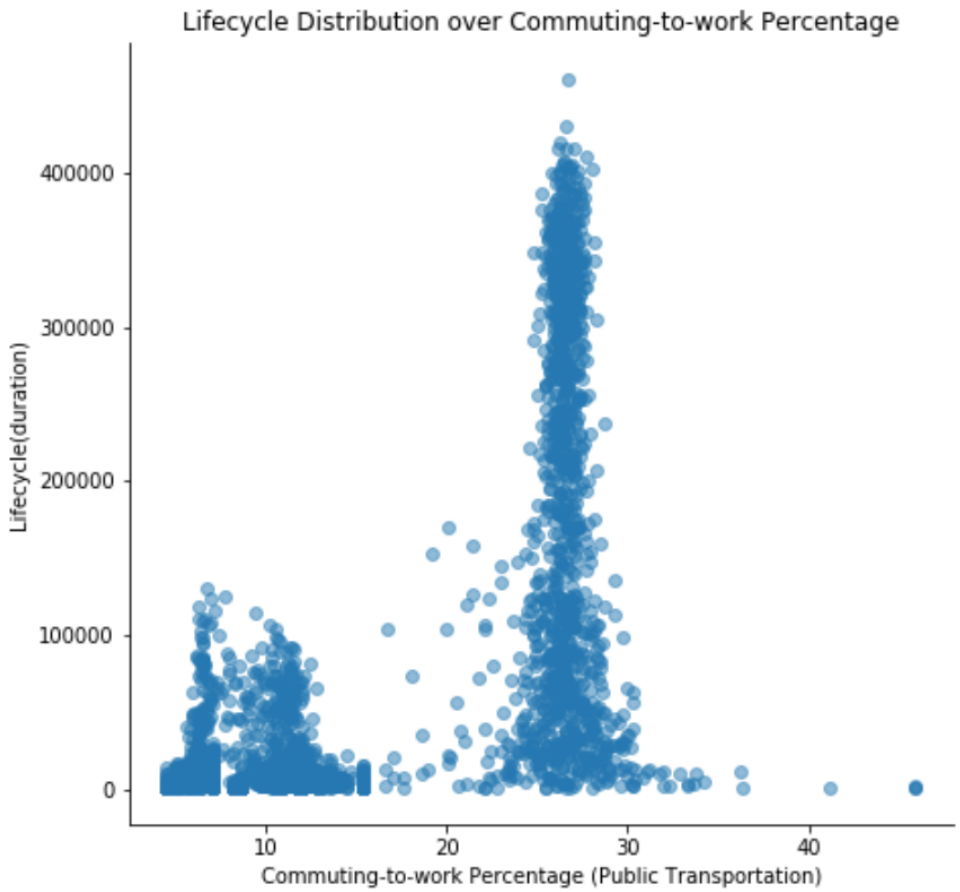


Fig. 6.3.3-1. Lifecycle vs Commuting-to-work Percentage

From the Lifecycle vs Commuting-to-work Percentage plot above, we can see that bike life cycles spike in areas with a 25%-30% (Public Transportation) Commuting-to-work rate. For areas with a (Public Transportation) Commuting-to-work rate lower than 15%, bike life cycles tend to be much shorter even though the total number of trips is large.

6.3.4 Lifecycle vs. Work Commuting Time

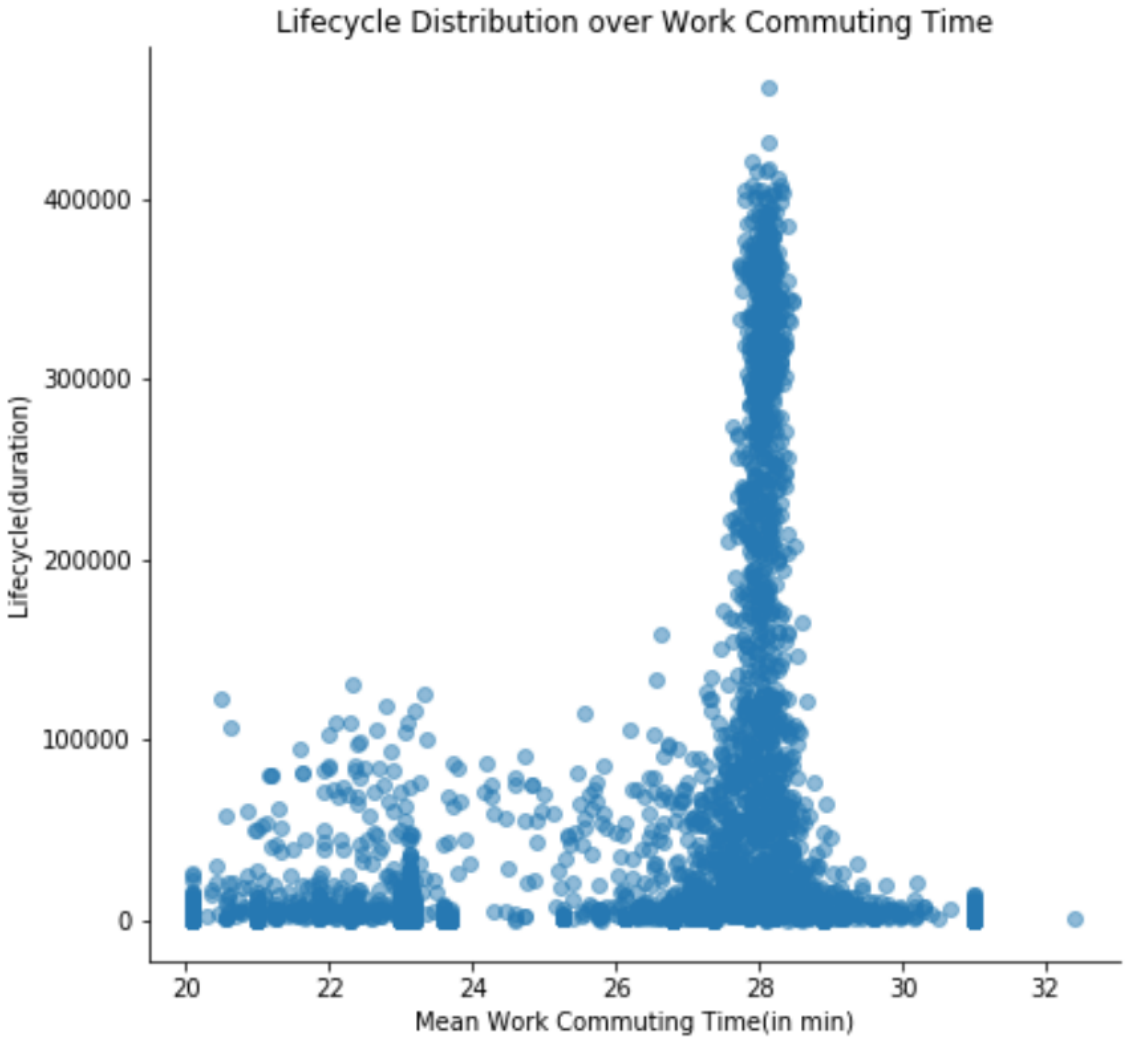


Fig. 6.3.4-1. Lifecycle vs Mean Work Commuting Time

From the Lifecycle vs Mean Work Commuting Time plot above, we can see that bike life cycles spike in areas with around a 28-minute work commuting time. For areas with less than a 24-minute work commuting time, bike life cycles tend to be much shorter even though the total number of trips is large.

6.3.5 Lifecycle vs. Percentage of High School or Higher

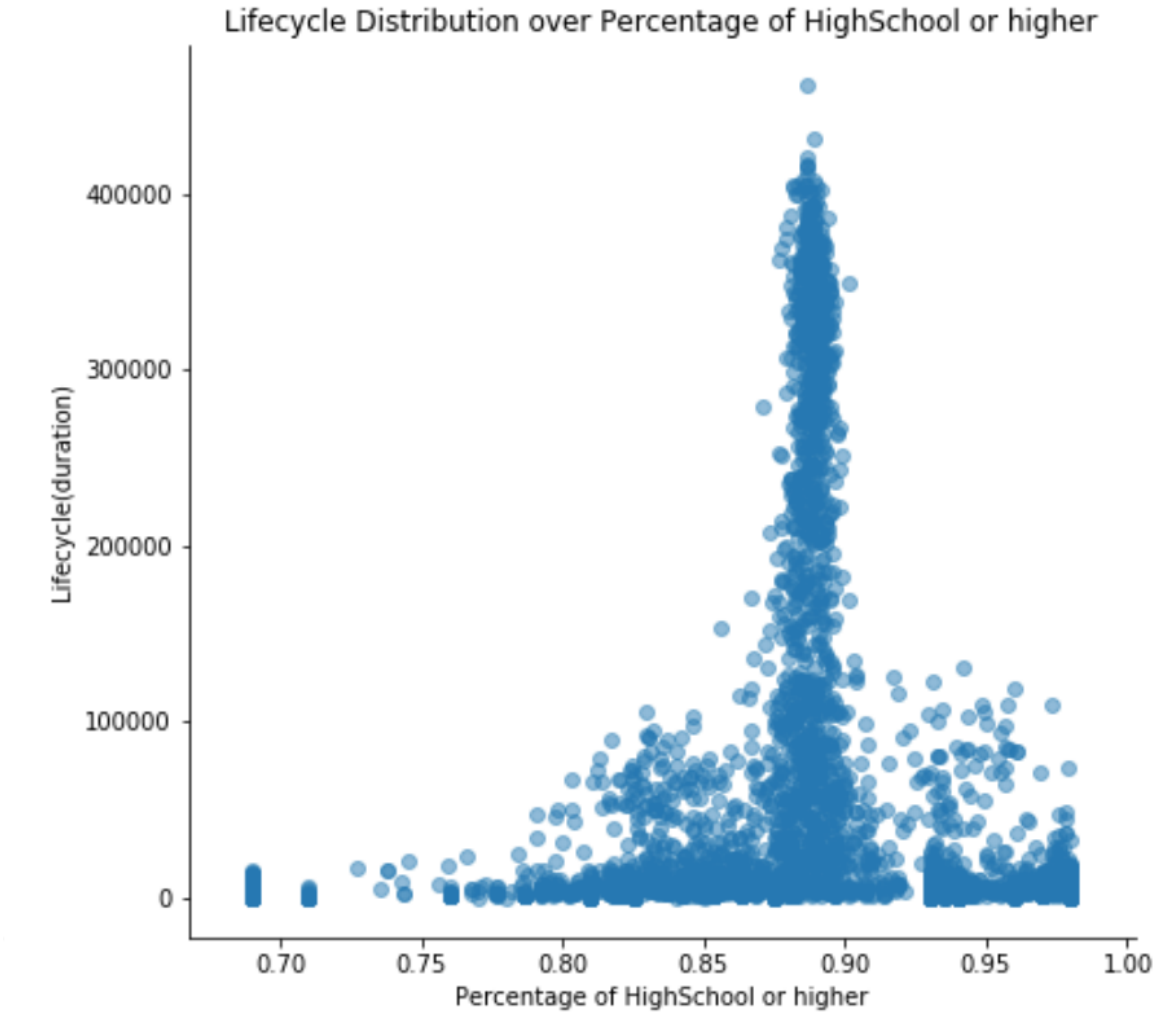


Fig. 6.3.5-1. Lifecycle vs Percentage of High School or Higher

For educational attainment, from the Lifecycle vs Percentage of High School or Higher plot above, we can see that bike life cycles spike in areas with 87% - 90% of High School or higher educational attainment. Even though the lifecycle is lower for areas with an even higher percentage (>90%) of educational attainment, we can still see the trends that within the range of 0-90%, the higher percentage of High School or higher educational attainment, the longer the bike life cycle.

6.3.6 Lifecycle vs. Percentage of Bachelor or Higher

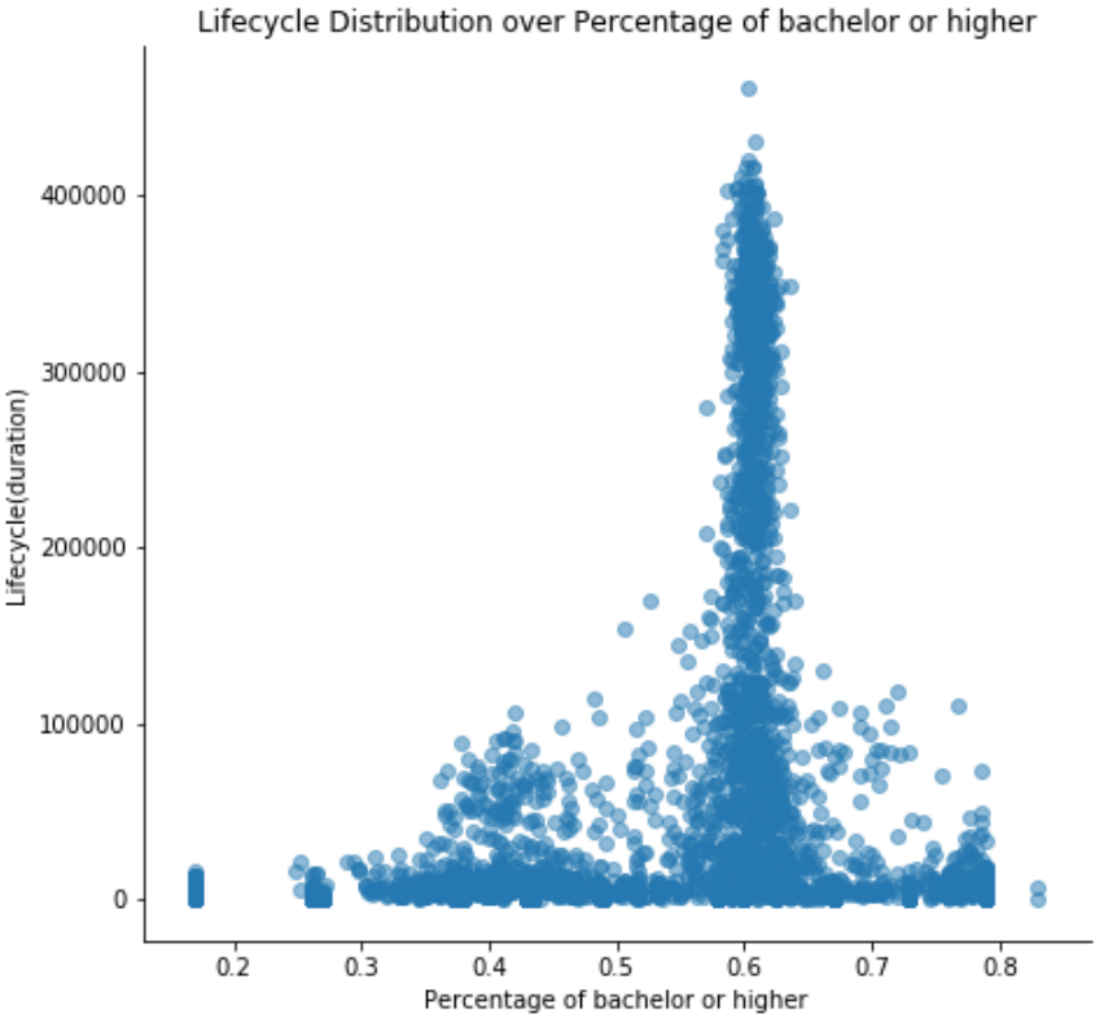


Fig. 6.3.6-1. Lifecycle vs Percentage of Bachelor or Higher

From the Lifecycle vs Percentage of Bachelor or Higher plot above, we can see that bike life cycles spike in areas with around 60% of High School or higher educational attainment. As we could infer from the correlation matrix, this feature is less impactful than Percentage of High School or Higher when analyzing bike lifecycle.

6.4 Crime

Using data taken from the San Francisco, Santa Clara County, Palo Alto, and Redwood City Police Departments, we created crime scores for each bike station. We filtered the police department data for crimes related to property damage and burglary because those would affect whether or not bikes need to be collected for maintenance the most. For each zip code with a

bike station, we averaged the number of these crimes that occurred in 2013, 2014, and 2015. We called this average the crime score.

First, we looked at how the crime score of the start station might affect the lifetime of a bike. We took the average crime scores of the start stations for all trips within each bike's lifetime. The graph below shows that most bikes start trips at stations with crime scores around the 3,200 mark. There are also a number of bike trips starting at stations with crime scores below 1,000.

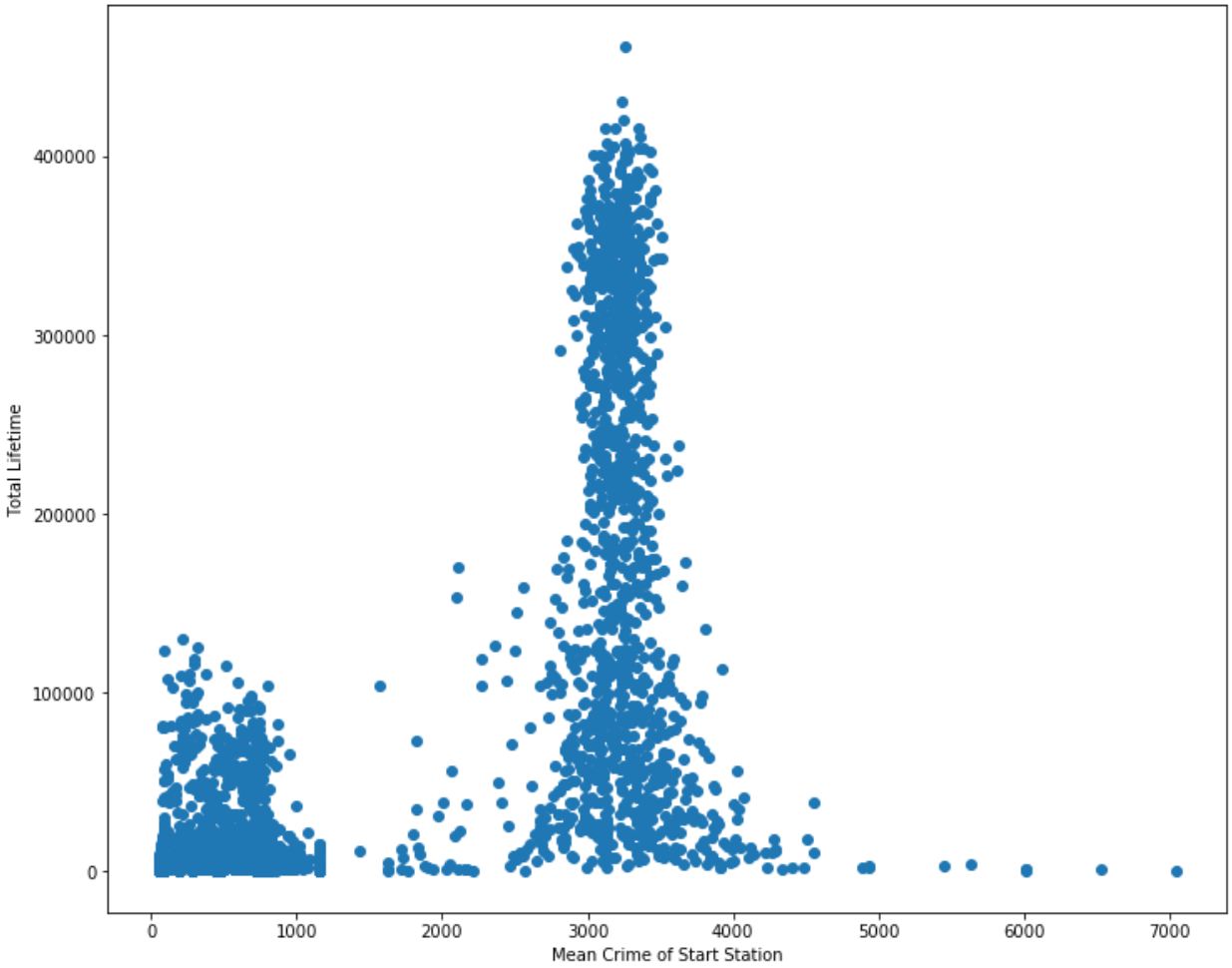


Fig. 6.4-1. Relationship Between Lifecycle and the Mean Crime of the Start Station

Then, we looked at how the crime scores of the end stations might affect the lifetime. The results for start and end station crime scores are largely the same since bike stations seem to be located in zip codes with similar crime scores (either below 1,000 or centered around 3,200).

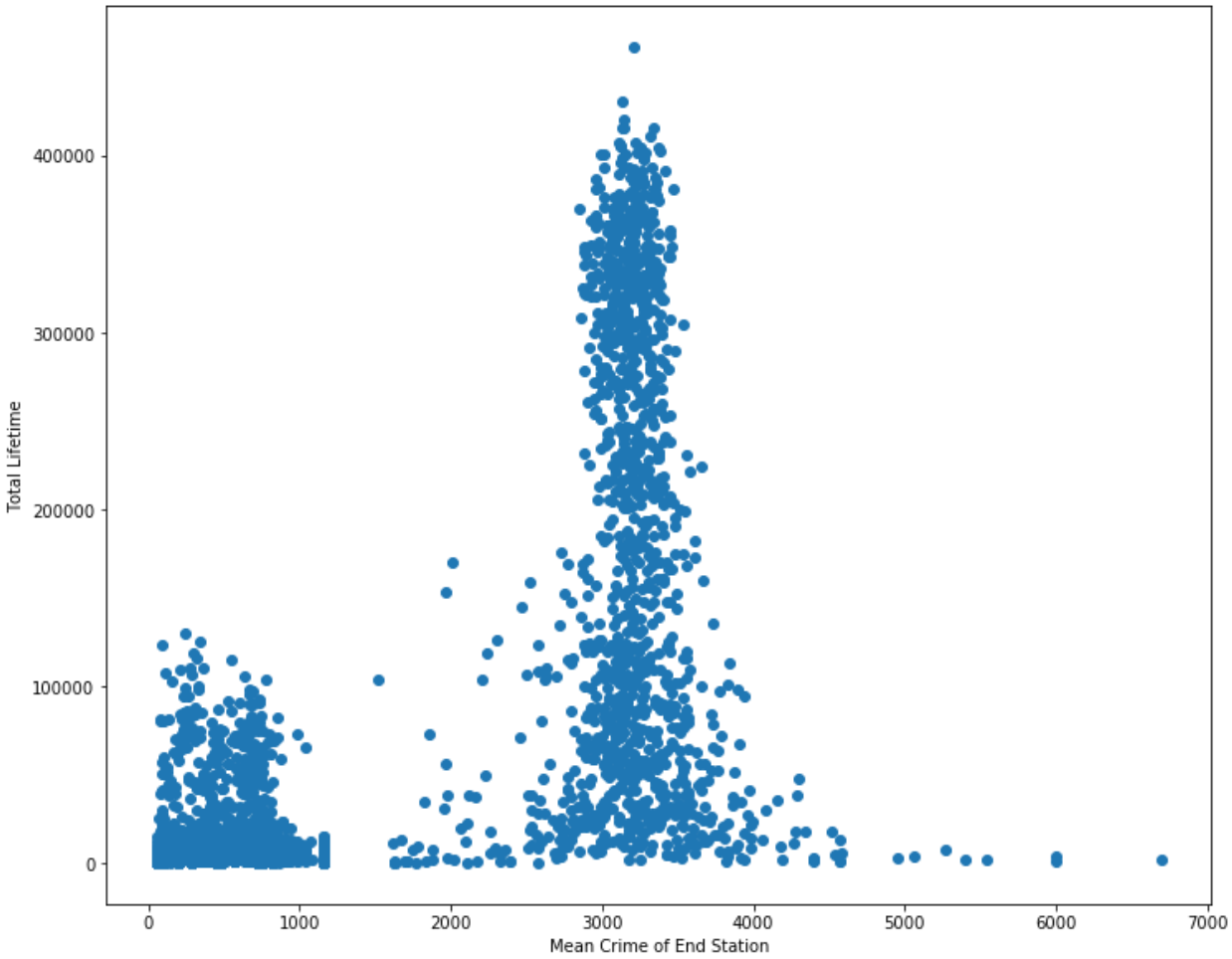


Fig. 6.4-2. Relationship Between Lifecycle and the Mean Crime of the End Station

7 Modeling

After feature engineering and data analysis, we decided to use linear regression as our first model. We used two metrics to compare the results of the models. At first, we calculated the percentage of the deviation of predicted values to the actual values. We used the median of the absolute value of the percentages as our first metric, calling it the “Error percentage”. Moreover, the percentage of RMSE to the mean duration of lifecycle was our second metric, calling it “RMSE percentage”. We used these two metrics since the error percentage could tell us the deviation of each sample while RMSE percentage is more general. From these two perspectives, we could get a better idea of how our models work.

7.1 Linear Regression

7.1.1 Modeling with the original dataset

To be more specific, for linear regression, we also used Lasso to perform the regularization and GridsearchCV to find the best hyperparameters. And all features were standardized.

At first, we decided to use all features for training. After getting the optimal hyperparameters, we then used all the training dataset to train the model and plot the feature coefficients.

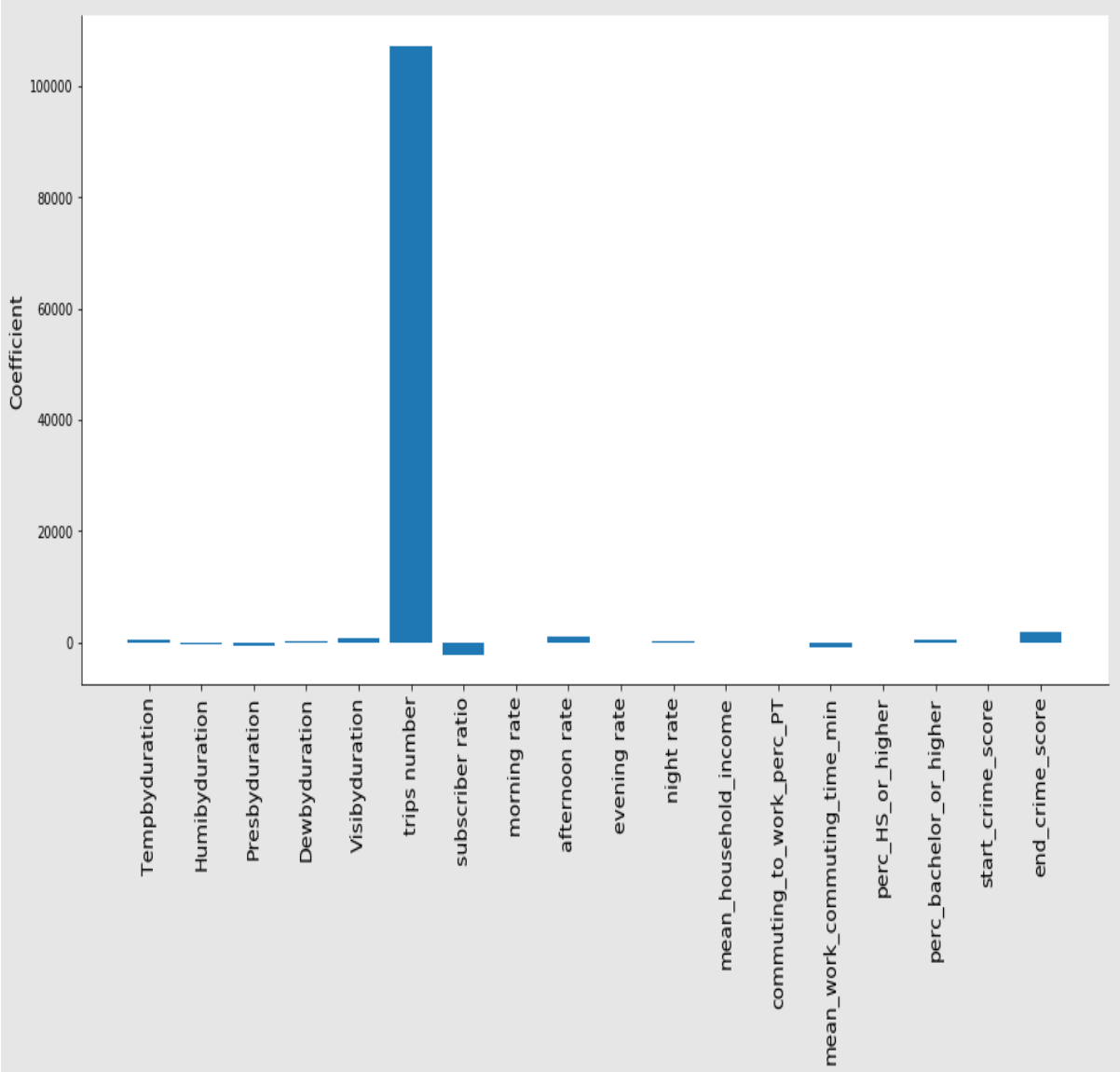


Fig. 7.1.1-1. Feature Coefficients with all features

We can see trip number's (count of all trips during a lifecycle) importance extremely exceeds others. Compared to trip number, other features are insignificant. This matches the result of our data analysis that only trip number has a clear linear relationship to the lifecycle duration.

Moreover, the RMSE percentage for the training dataset is 21.24% and the error percentage is 34.21%, while the RMSE percentage for the test dataset is 21.53% and the error percentage is 38.60%.

Then, to remove the huge impact of trip number and check the other features' impacts, we tried to use all other features except trip number to train a new model. As a result, we got more balanced coefficients compared to the model with all features.

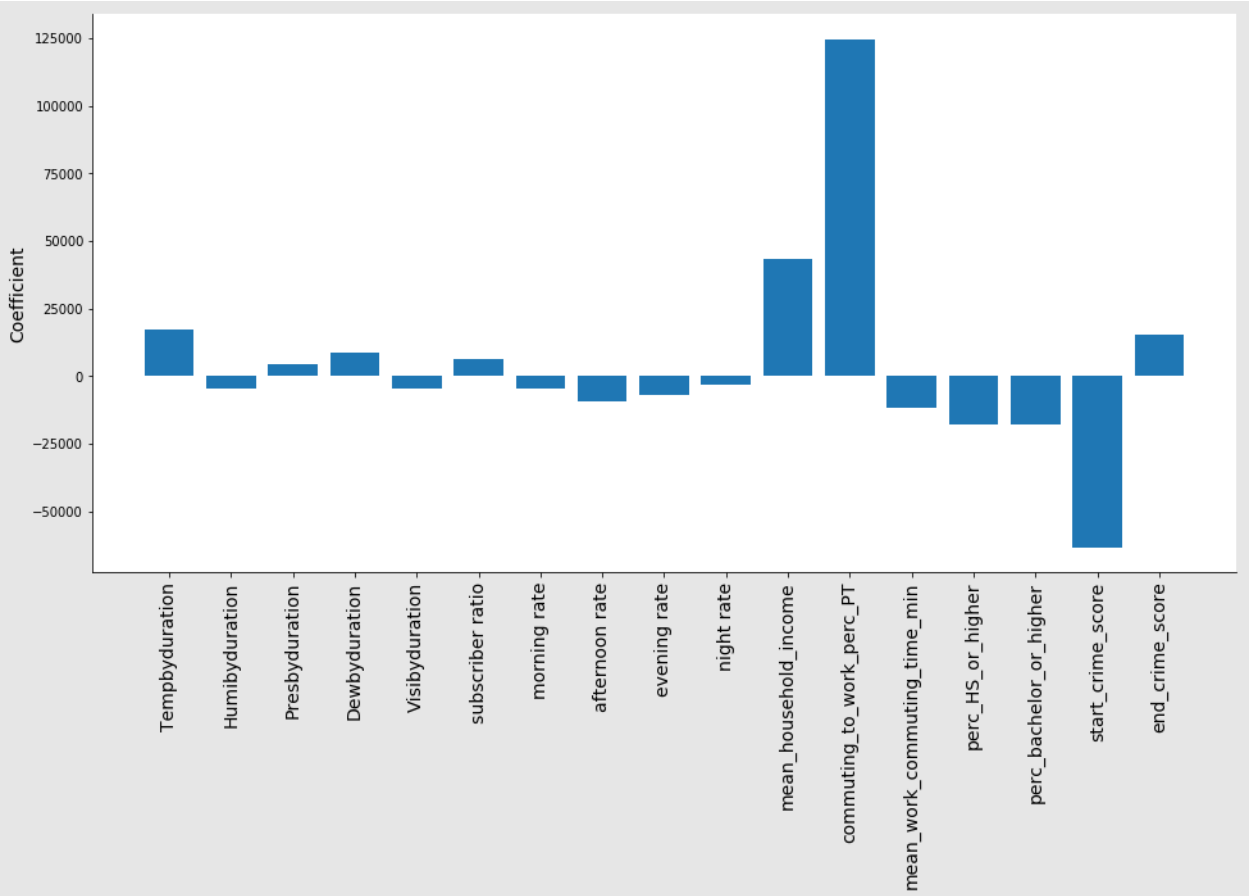


Fig. 7.1.1-2. Feature Coefficients without trips number

However, using this model, we failed to get a good results. All metrics are above 100% for the training dataset and testing dataset. Thus it is difficult to perform analysis based on this model.

7.1.2 Error analysis

Thus, using the model with all features, we performed error analysis based on the error percentage. At first, we plotted the relationship between the actual lifecycle and the error percentage as below:

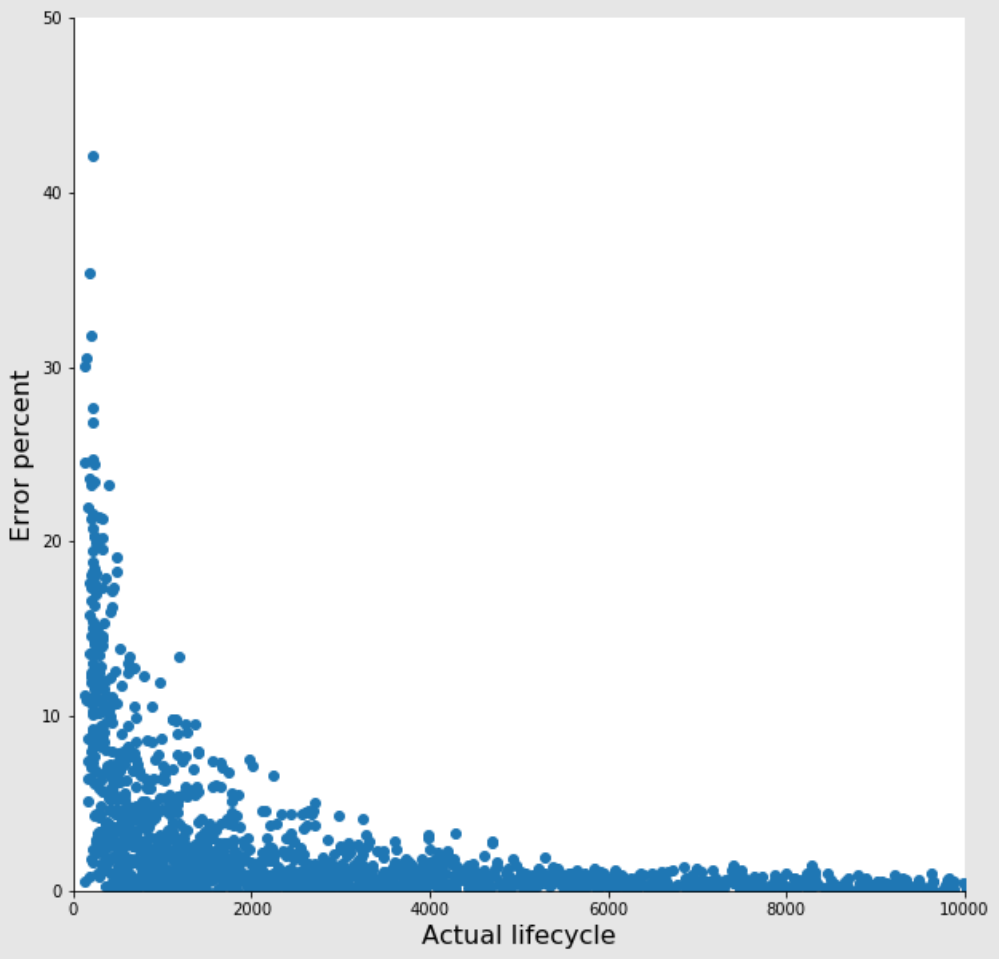


Fig. 7.1.2-1. Relationship between Actual lifecycle and Error percentage

Based on the plot above, we could see that the large error percentage happens when the actual lifecycle is very small. Furthermore, we checked all samples whose error percentage is larger than 10 and found 90% of their lifecycles are shorter than 600 seconds. At last, we checked all samples whose lifecycle is shorter than 600 seconds. We found the mean error percentage for these samples to be about 9.45.

In conclusion, when the the lifecycle is shorter than 600 seconds, the samples tend to be noise and outliers to the whole dataset. Thus we decided to remove these samples from the original dataset.

7.1.3 Re-modeling and comparing result

With the trimmed dataset, we re-trained our models with all features and without trip numbers. Below are the new results combined with the results in 7.1.1:

	Error_percent_test	Error_percent_train	Rmse_percent_test	Rmse_percent_train
original_allfeatures	38.60%	34.21%	21.53%	21.24%
original_withouttripcount	226.12%	214.22%	111.63%	112.39%
trimmed_allfeatures	30.33%	29.70%	20.10%	20.20%
trimmed_withouttripcounts	180.40%	172.31%	113.83%	104.70%

Fig. 7.1.3-1. Results of different models

We found that, after the dataset is trimmed, the performance of the model is improved, especially the model with all features. The error percentage is now 29.70% (from 34.21%) for the training dataset, 30.33% (from 38.60%) for the test dataset. The RMSE percentage is 20.20% (from 21.24%) for the training dataset, 20.10% (from 21.53%) for the test dataset. Removing the noise works!

However, although the performance of the model without trip numbers also improved, all metrics are still above 100% which is far below the result of the model with all features. Thus, we have to conclude that, except trip counts, other features do not fit linear models well.

Furthermore, since we can rely on only one feature in the linear models, we think it is better to try some models using different algorithms which can fully utilize the features in our dataset. Thus, as our next step, we used a decision tree to see if we could get a better result.

7.2 Decision Tree

7.2.1 Reason for this Model Selection

After the error analysis from Linear Regression, we decided to remove the outliers and get a refined dataset. Given this updated and more reasonable dataset, we wanted to see how tree-based models would work with it and we decided to try it out on the simplest one -- Decision Tree model. This also helped us figure out which features are most important in determining bike lifecycle.

7.2.2 Applying All Features then Select the Top N Features

To get an overview of the decision tree's performance, we first started by applying all the features in our dataset to train the model. In order to get the optimal hyperparameters, we used GridSearchCV to tune the hyperparameters and determine the best hyperparameters. Using the optimal hyperparameters, we used all the training data to train the DecisionTreeRegressor and plotted the feature importances:

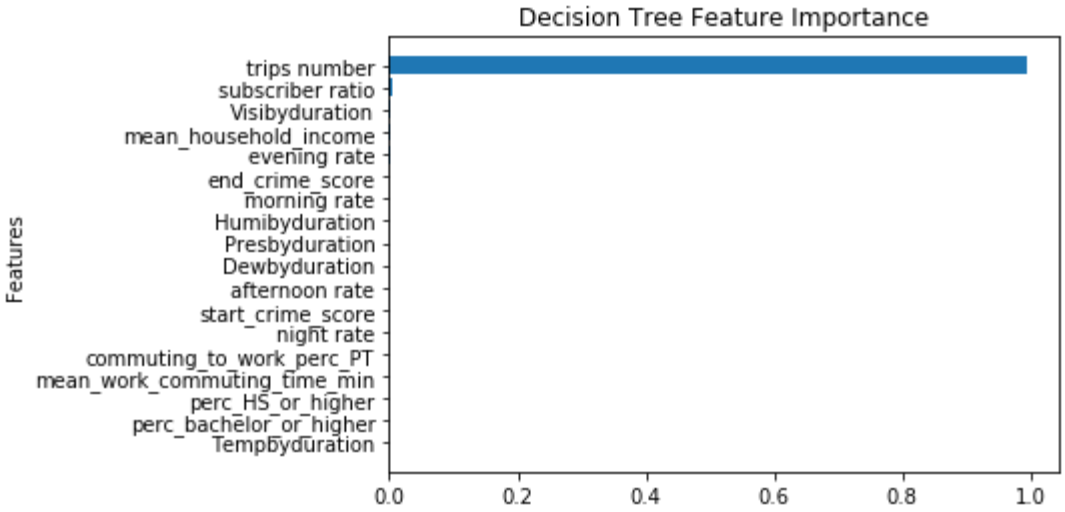


Fig. 7.2.2-1. Decision Tree Feature Importance

Again, we found that the “trips number” feature has such a high feature_importances score that it overshadows all other features. In order to further analyze other feature importances, we decided to remove the“trips number” feature then re-run all the training and fitting processes again as the next step.

Before moving on to the next step,:
(1) We calculated the two defined metrics (RMSE percentage and Error Percentage) for both training and test sets respectively using all the features.
(2) We extracted the top 4 most important features with feature_importances score greater than 0.00025.
(3) We tuned the model hyperparameters using the extracted top 4 most important features to get best hyperparameters.
(4) We trained the model using optimal hyperparameters and the train data set with only top 4 features.
(5) We calculate the two defined metrics for both training and test sets with only top 4 features respectively.

7.2.3 Re-run Model after Removing “trips number” variable

As mentioned above, we wanted to explore all the other features that were overshadowed by “trips number”, we re-ran the model with all other features after removing the “trips number” variable and plotted the feature importances:

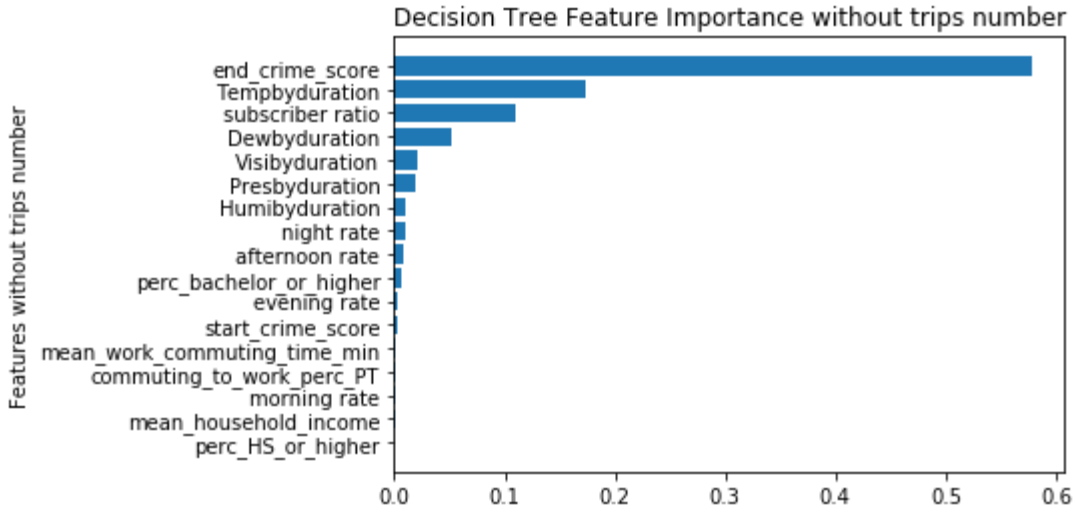


Fig. 7.2.3-1. Decision Tree Feature Importance without trips number

Now we can easily see how important each other feature is in determining the bike lifecycle (duration): “end_crime_score” ranks top 1, followed by “tempbyduration”, “subscriber ratio”, “Dewbyduration”, etc.

Similarly, we extracted the top 4 most important features with feature_importances score greater than 0.05. Then, we re-ran the the model according to the steps described in 7.2.2 to obtain the results for the two defined metrics (RMSE percentage and Error Percentage) for both training and test datasets respectively for the scenarios of applying all the features and applying only the top N features respectively.

7.2.4 Metrics Comparison

We wanted to see the overall performance of Decision Tree model under different scenarios, thus we drew a table to compare the metrics from different scenarios:

Metrics	All Features	Top 4 Features	All Features w/o tn	Top 4 Features w/o tn
Train RMSE perc	17%	12%	29%	35%
Test RMSE perc	21%	21%	58%	59%
Train err perc	25%	15%	22%	28%
Test err perc	26%	22%	36%	40%

Fig. 7.2.4-1. Decision Tree Model Metrics Comparison

From the table above, we can see when using all the features including “trips number”, we obtain an error percentage of 25% for training set and 26% for test set. When using the top 4

most important features, the error percentage for training and test set drops down to 15% and 22% respectively. As expected, both the RMSE and error percentage metrics are generally higher after removing the “trips number” variable.

Compared with the Linear Regression model, the Decision Tree improves a lot in performance. However, we still want it to be better and avoid overfitting. Thus we decide to move to the Random Forest model.

7.3 Random Forest

7.3.1 Reason for this Model Selection

As mentioned above, the Decision Tree overall performs better than Linear Regression, giving us a strong indication that the tree-based models would bring better performance to our dataset. Random Forest being a strong modeling technique and much more robust than a single decision as it aggregates many decision trees to limit the overfitting as well as bias, we decided to try out Random Forest and expect an even better result.

On top of that, it can also help us better observe which features are most important in determining bike life cycle.

7.3.2 Applying All Features then Select the Top N Features

Using the same method in model training and fitting as in 7.2, we first start with applying all the features in our dataset to train the model. In order to get the optimal hyperparameters, we used GridSearchCV to tune the hyperparameters and determine the best hyperparameters. Though we can expect better performance using more trees in a single forest, considering the computational power limitation of our machine, we set the maximum of n_estimators to 100. Using the optimal hyperparameters, we use all the training data to train RandomForestRegressor and plot the feature importances:

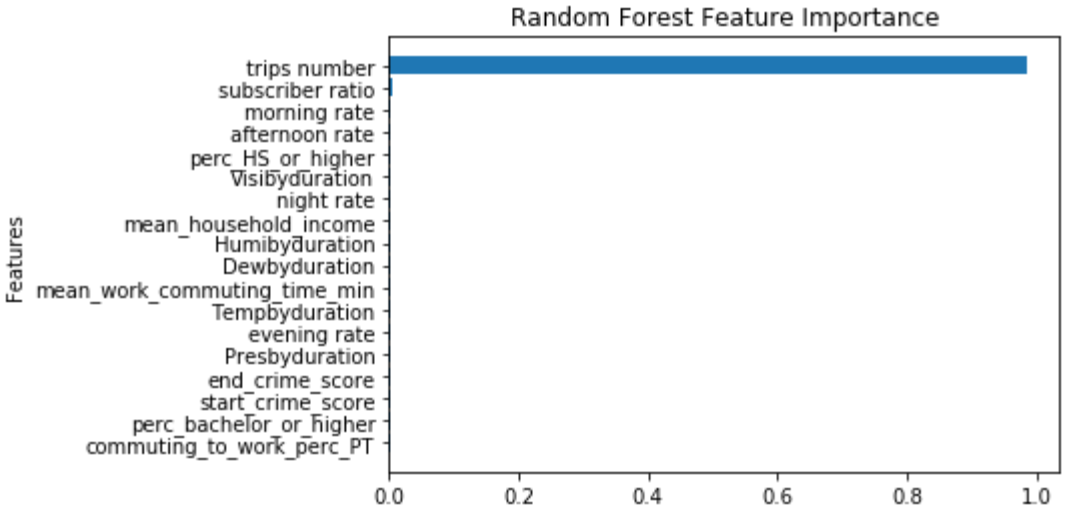


Fig. 7.3.2-1. Random Forest Feature Importance without trips number

Since “trips number” has a strong linear relationship with our target variable (“duration”), it overshadows all other features. In the next step, we remove the “trips number” feature for further feature importances exploration.

Before moving to the next step:
(1) We calculate the two defined metrics (RMSE Percentage and Error Percentage) for both training and test sets respectively using all the features.
(2) We extract the top 7 most important features with feature_importances score greater than 0.00062.
(3) We tune the model hyperparameters using the extracted top 7 most important features to get best hyperparameters.
(4) We train the model using optimal hyperparameters and the train data set with only top 7 features.
(5) We calculate the two determined metrics for both training and test sets with only top 7 features respectively.

7.3.3 Re-run Model after Removing “trips number” variable

To further explore all the other features overshadowed by “trips number”, we re-run the model with all other features after removing “trips number” variable and plot the feature importances:

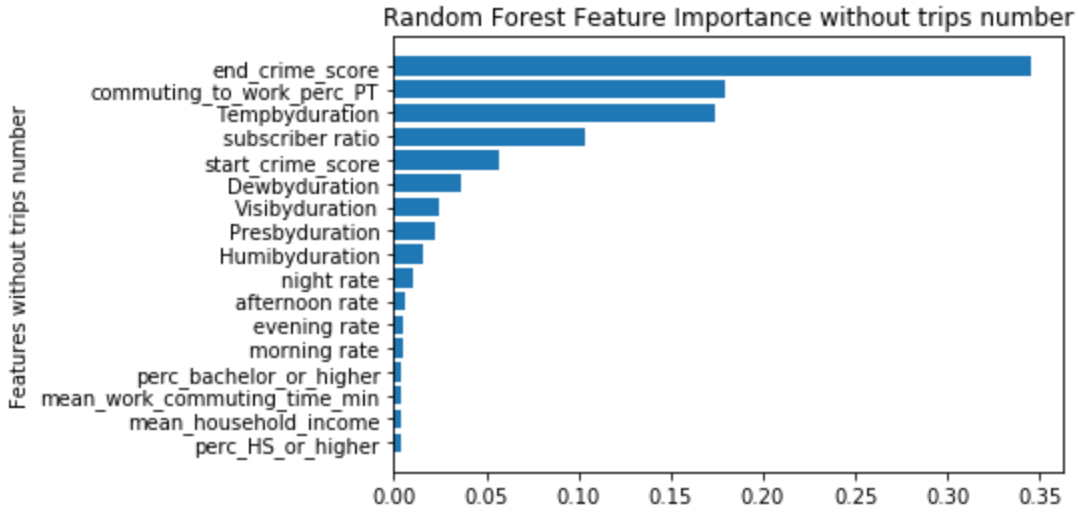


Fig. 7.3.3-1. Random Forest Feature Importance without trips number

As we can observe, the feature importances plotting under Random Forest is even more straightforward and more visible than the one from Decision Tree. Using this more robust model, we can determine that other than “trips number”, “end_crime_score”, “commuting_to_work_perc_PT”, “Tempbyduration” and “subscriber ratio” also matter a lot in determining bike lifecycle.

Similarly, we extracted the top 6 most important features with feature_importances score greater than 0.03 and re-run the the model according to the steps described in 7.3.2 to obtain the results for the two defined metrics (RMSE Percentage and Error Percentage) for both training and test datasets respectively, for the scenarios of applying all the features and applying only the top N features respectively.

7.3.4 Metrics Comparison

To better observe the overall performance of Random Forest model under different scenarios, we draw the following table for straightforward comparison using the two defined metrics:

Metrics	All Features	Top 7 Features	All Features w/o tn	Top 6 Features w/o tn
Train RMSE perc	8%	10%	15%	25%
Test RMSE perc	18%	17%	47%	52%
Train err perc	12%	11%	12%	22%
Test err perc	20%	19%	32%	36%

Fig. 7.3.4-1. Random Forest Metrics Comparison

We can see when using all the features including “trips number”, we achieve an error percentage of 12% for training set and 20% for test set. When using top 4 most important features, the error percentage for training and test set drops down to 11% and 19% respectively. As expected, both the RMSE percentage and error percentage metrics are generally higher after removing the “trips number” variable.

Compared with the Decision Tree model alone:
When using all features, the Error Percentage goes down from 25% to 12%, decreasing by half for the training set; from 26% to 20% for test set.
When using the top 7 most important features, the Error Percentage goes down from 15% to 11% for the training set; from 22% to 19% for the test set.
Meanwhile, the RMSE metric of the Random Forest is overall smaller in all scenarios than the corresponding one of the Decision Tree.

Compared with Decision Tree and Linear Regression, Random Forest achieves the best performance. Though a 20% error percentage is not perfect, the performance improves almost double from the first model we started off.

8. Challenges

In the process of completing this project, we encountered various challenges at different stages.

8.1 Data Collection Stage

At the data collection stage, we brainstormed all the possible related external datasets. After a lot of effort in narrowing down the related external datasets, we tried to collect all of them first. However, some of them are either incomplete or not downloadable. Thus we had to spent a significant time on collecting them, for example, we worked around it by scraping the weather data from a website.

8.2 Data Pre-processing Stage

The ETL process was also challenging when we had to decide various problems in integrating all the datasets, such as which zip code to use, which timestamp to use, which missing data to delete, how to set the thresholds, etc.

The biggest challenge in this stage is how to define the lifecycle of a bike. There is no clear definition of bike maintenance lifecycles. We reached out to Ford GoBike and their response is as follows:

"*We take a bike in for maintenance when we notice that it requires an in-depth repair that cannot quickly be completed in the field. We become alerted to bikes needing repair either through customers (who can push the red wrench button on the dock at the end of their trip to request service) or through our trained staff, who can identify repair needs."*

That doesn't help much in defining the bike lifecycle in our project. We moved on to try different ways such as K-Means to figure out a more reasonable way in this determination. For this challenge, we take an iterative approach to handle it throughout this whole project.

Another big challenge in this stage is trimming the data. It is hard to define a valid trip (one that is not unreasonably short or unreasonably long) and thus to trim the data. There is no certain threshold in this definition to trim data certainly. As with the lifecycle definition challenge, we had to try different methods and use different metrics to figure out a reasonable way of handling, undertaking an iterative approach.

8.3 Data Analysis & Feature Engineering Stage

After extracting all the possible related features, we then faced the challenge of correlating them. Too many features is not always good. We first tried to utilize all of them, however, we found out only a few of them have a linear relationship with the target variables. These few features easily overshadow others in some models. Due to this imbalance, correlating the available features is not easy, deciding which ones to apply and not apply to our model training is even harder. We tried different ways of feature engineering, however, evennow, we are still not very certain whether we got the optimized result.

8.4 Modeling Stage

With so many different models and algorithms available but with so limited time, it was hard to predict which models would work best. We decided to start from the simplest one -- Linear Regression. Then we also included Decision Tree, Random Forest and Neural Network in the early stages of our work. However, Neural Network didn't work out as well as we expected, giving us a very large RMSE compared with other models. The challenge is -- it is hard to

analyze the layers underneath to figure out what causes this large RMSE. Due to time limitations and the complexity of that model itself, we had to give up on Neural Network and focus on the Decision Tree and Random Forest.

9 Conclusion

From the three models that we trained, the random forest performed the best with a percentage error of only 20%. The top four features for predicting the length of a bike's lifecycle in this model were trip count, the mean crime score of the end stations, temperature by duration, and the subscriber ratio. When examining the features above, not many of them have a linear relationship with the lifecycle. This is likely why the random forest performs better than linear regression. In order to improve our results, we would want to include more features. For example, we could use a more granular definition of crime rather than an average crime score over three years. Furthermore, more information about the customers who use the bikes would be an interesting feature to explore and use to train our models. A 20% percentage error means that for the average lifecycle of 18 hours, our prediction for when the bike will need maintenance will lay within 3.6 hours of the true value. We believe this is a reasonable time frame target for the bike share company to check up on their bikes.

10 Appendix

[1] Jupyter Notebook: https://github.com/leon-lee-jl/MoBike/blob/master/MoBike%20V1.0.ipynb
[2] Raw Dataset:
https://drive.google.com/drive/u/1/folders/0B21m01hF4DL6MHd3eHQ4VWZrNFk
[3] Final Dataset: https://github.com/leon-lee-jl/MoBike/tree/master/Dataset

11 References

[1] https://www.kaggle.com/currie32/a-model-to-predict-number-of-daily-trips
[2] https://www.kaggle.com/ievgenii1101/analyzing-duration-of-the-trips
[3] https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset
[4] https://arxiv.org/pdf/1604.00664.pdf
[5] http://ascelibrary.org/doi/abs/10.1061/(ASCE)UP.1943-5444.0000273
[6] http://www.sciencedirect.com/science/article/pii/S1361920914000480
[7] http://www.bayareabikeshare.com/open-data
[8] Weather Underground Data: https://www.wunderground.com/
[9] San Francisco Police Department Data: https://data.sfgov.org/Public-Safety/Police-Department-Incidents/tmnf-yvry
[10] Santa Clara County Police Department Data: https://moto.data.socrata.com/dataset/Santa-Clara-County-Sheriff-s-Office/wrmr-tdyp
[11] Redwood City Police Department Data: https://moto.data.socrata.com/dataset/Redwood-City-Police/9wfx-9qes
[12] Palo Alto Police Department Data: https://moto.data.socrata.com/dataset/Palo-Alto-Police-Department/dnen-xaab

[13] Education Data from United States Census Bureau: https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=ACS_15_5YR_S1501&src=pt
[14] Income Data from United States Census Bureau: https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=ACS_15_5YR_DP03&src=pt