# Predicting Bike Share Demand in San Francisco Bay Area

Haneef Pervez Mahwish Jawed Michael Yen Sriya Vudata

**0. ABSTRACT**

The growth of bike sharing in cities around the world presents the challenge of uneven bicycle distributions between stations in bicycle-sharing system (BSS) networks. BSSs typically resolve this by physically relocating bicycles from one station to another, but this is a costly and reactive endeavor. Bicycle imbalance is highly dependent on local bicycle and dock availability, so modeling these two components would enable greater comprehension of the statuses of bicycle stations and bicycles at future times, which may enable key rebalancing strategies. This paper is the first to take a look at predicting local bicycle and dock availability in the San Francisco Bay Area using XGBoost. We propose a spatio-temporal bicycle and station availability model trained with XGBoost, an optimized gradient boosting framework, using historical data from a two-year data set with 70 stations and over 660,000 bike trip records along with meteorological data to measure the model's efficacy. In our evaluation we used cross-validation and sensitivity analysis to prove the model's accuracy, and compare its performance and error to other methods.

**1. INTRODUCTION**

Bike sharing systems (BSSs) have been growing in popularity due to an increased need for convenient transportation but also because of a concern for the environment and personal health. They are especially useful in cities, where bikes can freely travel distances much more quickly than walking or public transport but are also more efficient in space than cars [3]. Typically, BSSs are characterized by *stations*, where the bikes are stored unattended in-between uses. Users do not have to bring the bike they used back to the original station they borrowed from; they can take it to any station in the *network* as long as there is a space available. These features of BSSs allow them to provide service *on demand* with a *decentralized* structure [4].

However, bike sharing comes with common problems, including bike shortages at stations from uneven customer mobility patterns and the heavy dependence on the climate. To ensure there are enough bikes at each station, it is often necessary to even out the distribution of bikes by physically moving them from one station to the other. This is done via trucks or trailers, which can be costly.

To mitigate the costs of rebalancing, studies have been conducted with different approaches to create bike mobility or demand prediction models that can enable new programs to redistribute bikes. Such measures include service optimization, monetary incentives, or changing the layout of stations in the network. Past studies [6] and [7] look at modeling bike availability using dynamic linear models and machine learning (Random Forest, Least-Squares Boosting, Partial Least-Squares Regression). So far, work to find the best performing fine-grained and localized prediction models for the availability of bikes and spaces at each station has proven to be challenging. The difficulty of this task results from prediction models having to take into account timing and weather in addition to spatial and temporal factors. Undesirable pedaling conditions such as light rain or snow discourages bicycling and it is outright unsafe during storms with high wind or precipitation [8].

Here, we are going to train a localized bike and dock availability prediction model. We aim to formulate availability predictions for each station by using both historical bike station data and meteorological data. Additionally, our work differs from past studies in that our prediction model was trained through XGBoost, which utilizes *gradient boosting* in an advanced manner and may be able to outperform past studies. Predicting availability allows redistributors to know if there will be a shortage or excess of bikes at any station in the future.

This paper makes the following contributions:

- We define the bike and station availability modeling problem and establish a dynamic spatio-temporal availability model for BSSs

by analyzing interactions between all stations;
- To our knowledge, we conduct the first work on bike and station availability prediction models using XGBoost;
- We develop and evaluate our model through cross-validation and sensitivity analysis with anonymized data from Bay Area Bike Share obtained from Kaggle, a two-year dataset of the San Francisco Bay Area, with 70 stations and over 660,000 bike trip records.

The rest of this proposal paper is going to be organized in the following manner: We first provide a look at related work that has been done regarding this topic in Section 2. In Section 3, we go over our motivation, and in Section 4, we explain our main design choices for feature extraction and prediction modeling of the data. Section 5 describes and evaluates our results, Section 6 offers insights and limitations to our work, and we wrap up our discussion in Section 7.

## 2. RELATED WORK

In this section, we will discuss some of the related work in this area, and how our approach differs from other works.

### 2.1 - SOA Approaches

Different methods have been used when trying to model bike share demand. Some of the more recent approaches have used weather data with historical data on which stations and what times bikes are being checked in and checked out. In [5], researchers used data from Hangzhou, China to predict demand by splitting check-in and check-out into two separate actions. They then used random forest to create their model. In [2] the researchers used data from CitiBike in New York City as well as weather and holiday data to create a neural network model to predict bike share demand. There have also been approaches that use a similar dataset to ours, [7] uses machine learning to predict bike availability, and [6] uses dynamic linear models. All of these methods looked at historical checkin and checkout data and used weather and event data to create a prediction model.

### 2.2 - Our Approach

We used similar types of data including historical trip data and weather data, however our approach differs in that we use XGBoost to create our model. XGBoost provided us with both a computational advantage and an accuracy advantage when compared to random forest or a neural network. Understanding bike share demand can lead to crucial insights into problems such as bike rebalancing, service optimization, and human mobility, and by using a new method we hoped to build on these earlier techniques by providing a model that is more accurate and can analyze more data with less computing power.

## 3. MOTIVATION

In this section, we will discuss several applications of predicting bike demand and availability that will give insight into the importance of solving this problem.

### 3.1 - Bike Allocation

One motivation for predicting bike availability and demand for bikes is to allow city planners and bike share service providers to meet the demand for bikes on a given day. For example, if we can predict using our model that the Mountain View City Hall station will need 10 bikes for tomorrow, but currently has 5 bikes with a small chance of any bikes being checked in at the station, then the corresponding service can allocate more bikes to this station from another station that is predicted to not need as many bikes. This can especially help in cases where there is a shortage of bikes during the morning rush hours, or when docks are full and users are unable to return their bikes at the station. The model is able to access commuting and recreational trends from the data collected, which can thus help predict the uneven distribution of bikes at the end of a work day.

### 3.2 - Expanding the Network

Predicting demand can also help in designing and expanding the bike sharing network, especially for building new stations. For instance, if there is high demand for bikes predicted from a certain region where stations are located, this can be a good indicator for where to build a new station. Additionally, knowing the volume of demand at a specific station can also be an indicator of whether more docks should be added to a station.

**3.3 - Keeping up with Demand over the Years**

For the future, this is especially important as more people opt for bikes over cars for commuting. In 2019, San Francisco experienced a 14% increase from the previous year in the number of cyclists during AM and PM peak periods [1]. If this trend continues over several years, then it will be even more important for bike sharing services to have a means of predicting and meeting their demand. Especially during COVID-19 times, where public transportation is greatly discouraged, metro areas across the US have seen an increase in bike activity, some as much as twice the activity in May 2020 compared to May 2019 [2]. Figure 1 shows the difference in miles traveled from May 2019 to 2020 during quarantine, and as we can see, there is generally an increase in several locations. This highlights the significance of being able to predict bike sharing and take actions necessary to meet future demand.


Figure 1: Change in total bicycle miles traveled during quarantine in the US [2].

**4. MAIN DESIGN**

In this section we will discuss our main design for predicting bike availability and demand at a particular bike docking station. All of our work was implemented in Python using external modules for XGBoost and scikit-learn for the baseline models. Before building our prediction model, we assumed that bike demand will be high during weekday mornings, when the temperature is warmer, and on weekends and holidays. In other words, given these conditions, there will be more bikes checked out than checked in at a particular station. To verify our assumptions, we extracted two sets of data from different seasons for training, one from Summer 2014 (July 01-September 30), and the other from December 2014 (December 01-31), and performed cross-validation.

**4.1 - Problem Statement**

Using anonymized bike trip data in San Francisco's Bay Area from the dates as described, we performed feature selection and built a prediction model using XGBoost. We predicted bike availability at a future time interval through the analysis of historical data on the number of bikes and docks available at a particular station. Such predictions will enable city planners and bike share service providers to adjust bike and dock supply for the needs of the community. Formally, given the station id of a bike station and other parameters such as the time and the weather, we predict the number of bikes available using our XGBoost model.

**4.2 - Data Collection**

We utilized a Kaggle dataset to access Bay Area Bike Share data [10]. The dataset contains information on 70 stations and over 660,000 bike trips. The dataset also provides valuable information on each docking station, bike trips taken by commuters, and the weather (maximum, minimum, mean in regards to temperature and humidity on a given day) on days where bike sharing services were utilized by the public. It provides latitude and longitude coordinates of the location of each docking station, the number of bikes and docks available in a certain time interval, the total number of docks, and the city the station is located in. Daily weather statistics from the weather file were used to monitor bike and dock availability during varying temperatures and humidities. After performing data cleaning, we performed feature selection and obtained features from the weather and Status tables to train our model.

| # id | A name | # lat | # long | # dock_count | A city | A installation... |
|---|---|---|---|---|---|---|
| 2 | San Jose Diridon Caltrain Station | 37.329732 | -121.9017820000 0001 | 27 | San Jose | 8/6/2013 |
| 3 | San Jose Civic Center | 37.330698 | -121.888979 | 15 | San Jose | 8/5/2013 |
| 4 | Santa Clara at Almaden | 37.333988 | -121.894902 | 11 | San Jose | 8/6/2013 |
| 5 | Adobe on Almaden | 37.331415 | -121.8932 | 19 | San Jose | 8/5/2013 |
| 6 | San Pedro Square | 37.336721000000 004 | -121.894074 | 15 | San Jose | 8/7/2013 |

Figure 2: Station data via Kaggle dataset

| # station_id | # bikes_avail... | # docks_ava... | time |
|---|---|---|---|
| 2 | 2 | 25 | 2013/08/29 12:06:01 |
| 2 | 2 | 25 | 2013/08/29 12:07:01 |
| 2 | 2 | 25 | 2013/08/29 12:08:01 |
| 2 | 2 | 25 | 2013/08/29 12:09:01 |
| 2 | 2 | 25 | 2013/08/29 12:10:01 |
| 2 | 2 | 25 | 2013/08/29 12:11:01 |

Figure 3: Station status data per minute via Kaggle dataset

Figure 4: Individual trip data via Kaggle dataset

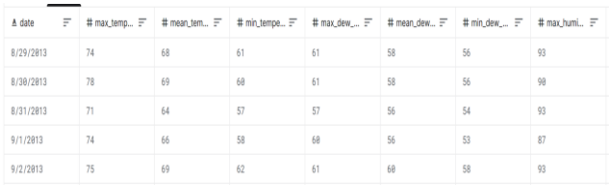


Figure 5: Daily weather data via Kaggle dataset

### 4.3 - Data Cleaning

We stored the data from the Kaggle dataset on an Amazon RDS MySQL database. Since the dataset was around 4 GB, we extracted chunks of data from the Station, Status, and Weather tables for our analysis. We looked at bike share data from July to August 2014 and December 2014. We merged the Status and Weather data into one table for each chunk, and used the merged tables for training purposes.

### 4.4 - Feature Selection

We performed feature selection using Pearson's Correlation Coefficient, given by Equation 1,

$$r_{x,y} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2}}$$

Equation 1

and univariate linear regression, which takes the correlation and converts it to an F score then to a p-value. In Equation 1, $x_i$ and $y_i$ are the paired data points being tested for correlation: *bikes_available* and a regressor variable.

First, relevant data including all columns was extracted from the database into a pandas DataFrame. To prepare the data, the datetime feature was separated into different numerical categories, such as the hour of day or the day of week. Then, all rows containing a NaN value in any column besides the categorical weather-related events column (no precipitation, rain, etc) was removed. The amount of rows under this condition was negligible compared to the total amount of rows. Lastly, to ready the data for regression, the weather-related events column was converted to integers using one-hot encoding. The inputs to feature selection were the various regressor variables (including station id) and the target variable, *bikes_available*. By applying feature selection, we obtain as output the top 17 highest scores and their corresponding features to be used for evaluating the baseline and proposed models.

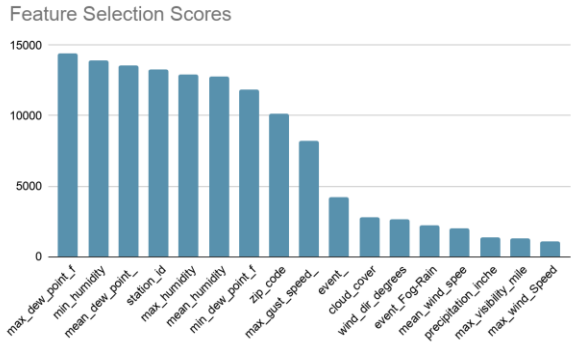



Figure 6: Feature scores as obtained from feature selection

### 4.4 - Prediction Methods

We trained our model via the XGBoost prediction modeling algorithm. XGBoost utilizes gradient boosting, which iteratively combines several prediction models. A loss function is computed and utilized to craft a fit for a new model in an attempt to reduce loss. In this experiment, the loss function will compute the mean squared error. We compared the

performance of XGBoost to the comparatively naive decision tree and random forest models [9]. XGBoost has a number of useful parameters. For our project, we set the learning rate = 0.05 for more accuracy, and set n_estimator (number of different trees in gradient boosting) to 200. As Figure 7 shows, several values for n_estimators were tested while setting early termination to 5 rounds to optimize the runtime. The value 200 proved to be the lowest value with the lowest mean absolute error, which optimized the runtime of the model. Since we used XGBoost, we compared its performance to that of the models used by other researchers. We established the decision tree and random forest models as baselines, and implemented the default version of each in Python, as previous studies utilize these two models for their predictions.

| n_estimators | mean absolute error |
|---|---|
| 50 | 2.5136918403581 |
| 100 | 2.3901617519956 |
| 200 | 2.3827874996785 |
| 500 | 2.3827874996785 |
| 1000 | 2.3827874996785 |

Figure 7: MAE for XGBoost on December 2014 for different numbers of trees

One thing to note is that all models used were trained at the network level, meaning they were shown data points across all stations. However, the models predict at the station level for a given data point.

### 4.5 - Model Testing and Validation

Both our datasets are ordered by time, and within the same time are ordered by station. Figure 8 shows how our dataset is ordered after undergoing data cleaning. For both datasets, we used 80% of the data from our dataset for training purposes and the remaining 20% for validation without cross validation. For a month of data, this roughly translates into the first three weeks used for training, and the final week used for validation. With cross validation, for the December 2014 set, we divided our data into five chunks of 20% of the data. Four chunks were used for training, and one chunk was used for validation. It was repeated five times, with a different chunk used for validation each time. This process, called 5-fold cross validation, provided a more accurate depiction of the model's quality. Furthermore, we performed sensitivity analysis to prove the significance of including certain features in our model.

| | station_id | bikes_available | docks_available | time | date |
|---|---|---|---|---|---|
| 129600 | 27 | 6 | 9 | 2014-12-01 00:00:04 | 2014-12-01 |
| 148320 | 28 | 13 | 10 | 2014-12-01 00:00:04 | 2014-12-01 |
| 167040 | 29 | 12 | 11 | 2014-12-01 00:00:04 | 2014-12-01 |
| 185760 | 30 | 6 | 9 | 2014-12-01 00:00:04 | 2014-12-01 |
| 204480 | 31 | 7 | 8 | 2014-12-01 00:00:04 | 2014-12-01 |
| ... | ... | ... | ... | ... | ... |
| 3023114 | 31 | 5 | 10 | 2014-12-31 23:59:03 | 2014-12-31 |
| 3047581 | 32 | 3 | 8 | 2014-12-31 23:59:03 | 2014-12-31 |
| 3072048 | 33 | 7 | 8 | 2014-12-31 23:59:03 | 2014-12-31 |
| 3097968 | 80 | 7 | 8 | 2014-12-31 23:59:03 | 2014-12-31 |
| 3123888 | 84 | 7 | 8 | 2014-12-31 23:59:03 | 2014-12-31 |

[3123889 rows x 29 columns]

Figure 8: An example of our cleaned dataset for December 2014

## 5. EVALUATION

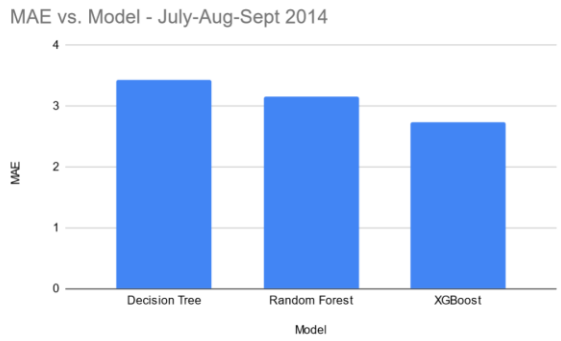
This section will discuss our results from running the baseline models and the XGBoost model. Our evaluation method is split into three parts: a baseline comparison, where we run all three models on both of the datasets; cross validation, which was run on all three models for December 2014; and sensitivity analysis of the XGBoost model on July-September 2014. For all evaluation methods, we calculate the mean absolute error (MAE) on the validation data between the predicted values for *bikes_available* and the corresponding actual values for *bikes_available*. The formula is as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |x_i - y_i|$$

where $n$ is the total number of data points, $x_i$ is the actual value of *bikes_available* for data point $i$, and $y_i$ is the predicted value of *bikes_available* for data point $i$. A lower MAE indicates a more accurate model.

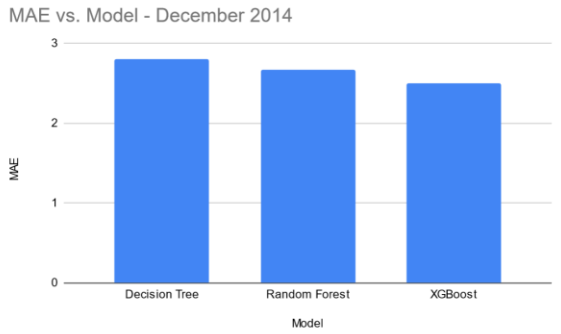
### 5.1 - Baseline Comparison

For the July-August-September 2014 and December 2014 datasets, all three models were run without cross validation and outputted the MAE. The results are shown in Figures 9-10. As shown, out of the three models used, XGBoost had the lower MAE across both datasets, and thus predicted the number of bikes available with more accuracy. In [6] and [7], these studies implemented prediction models using advanced versions of Random Forest. While our baseline is more basic, we were interested in seeing how well XGBoost performed in comparison.




*Decision Tree: 3.440707212*
*Random Forest: 3.156420391*
*XGBoost: 2.741021095*

Figure 9: MAE for models on July-September 2014 data




*Decision Tree: 2.800259*
*Random Forest: 2.674820*
*XGBoost: 2.382787*

Figure 10: MAE for models on December 2014 data

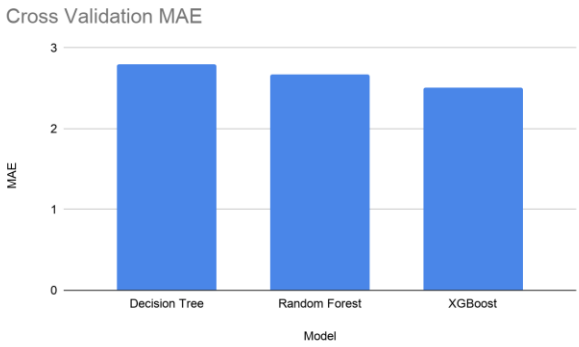
| Model | Time |
|---|---|
| Decision Tree | 52.67 s |
| Random Forest | 72 min |
| XGBoost | 10 min |

Figure 11: Run time for models on July-September 2014

Figure 11 also shows the runtime for the three models on the July-September 2014 dataset. The Decision Tree model took the shortest amount of time, but had the largest MAE. In comparison, XGBoost took much less time than Random Forests and had a lower MAE. As such, we found that XGBoost significantly outperformed the baseline models in predicting bike availability.

### 5.2 - Cross Validation

5-fold cross validation as described above in Section 4.5 was used on the December 2014 dataset to ensure the best evaluation accuracy for the smaller dataset. Results are shown in Figure 12; XGBoost had the lowest MAE with cross validation showing that it is more accurate than the baselines.




*Decision Tree: 2.800259*
*Random Forest: 2.674820*
*XGBoost: 2.506730*

Figure 12: Cross Validation MAE for models on December 2014 data

### 5.3 - Sensitivity Analysis

Through feature selection, we were able to extract the top 17 features from the merged status and weather tables for both the July to August 2014 and December 2014 data sections. We removed one feature at a time and ran XGBoost with 16 features. While we observed some increases in the Mean Average Error, the removal of certain features led to a decrease, which indicated the possibility of overfitting our model. We realized that the top 17 features often included both the mean and the maximum or minimum for some weather-related columns. For example, when we performed feature selection for the July to August 2014 section, we found that the columns *max_temperature_f* and *mean_temperature_f* were both present. Thus, we re-ran XGBoost after removing either the maximum or minimum value if present alongside the mean value for the same characteristic. Before removing redundant features, the Mean Average Error for the July to August 2014 dataset was 2.7410210946360647. After removing them, it was

2.7407746530121124. Thus, there was only a slight decrease in the Mean Average Error, indicating overfitting and confirming the presence of unnecessary features in training.

| Feature Removed - July-Aug-Sept 2014 | Mean Average Error |
|---|---|
| **station_id** | 3.128474761 |
| max_temperature_f | 2.734431179 |
| mean_temperature_f | 2.733845304 |
| max_dew_point_f | 2.725011334 |
| min_humidity | 2.734964747 |
| max_visibility_miles | 2.741021095 |
| mean_visibility_miles | 2.736117303 |
| **max_wind_Speed_mph** | 2.746182244 |
| mean_wind_speed_mph | 2.729745636 |
| max_gust_speed_mph | 2.739144059 |
| **cloud_cover** | 2.744234484 |
| **zip_code** | 2.748782133 |
| **time_month** | 2.751311782 |
| time_week | 2.71100345 |
| **time_hour** | 2.835032363 |
| event_ | 2.728538469 |
| event_Rain | 2.731235222 |

Figure 13: Sensitivity analysis results for July-August 2014; bolded feature indicates removal of that feature results in increase of Mean Average Error

## 6. INSIGHTS

Being able to predict bike availability by station can be used in a number of applications. The largest obstacle when starting a bike share program in a city is the issue of rebalancing. Because a characteristic of bike share programs is that bike availability fluctuates depending on a number of factors, bike rebalancing is a costly and inefficient process. By being able to predict bike availability at a station, bike rebalancing becomes easier because a bike share company can predict what the bike availability at a specific station will be depending on day, time, weather, etc. For example if they know a specific station will be busy on a Tuesday in summer weather, they can redistribute bikes from a station that is less busy during those same conditions. This model can also help with building new stations and expanding the bike share network by indicating which stations have a high demand.

This type of research can also be applied to other transit options that work similarly to bike share programs. For example, e-scooter programs where users can pick up a scooter and drop it off at a station has similar redistribution issues. This sort of research can be expanded to work for scooters as well.

### 6.1 Limitations

Bike availability is also affected by factors such as elevation and city area function. Unfortunately, we were not able to find accurate data on these factors. Our model is limited in the fact that there are still some factors that affect availability that are not accounted for, however as technology progresses and this sort of data becomes available the model can be expanded to provide an even more accurate prediction.

### 6.2 Privacy

All data collected in this paper is publicly available and anonymized, obtained from San Francisco Bay Area Bike Share. There is no identifying information that can be used to match bike station data or unused trip data with a user.

### 6.3 Potential Future Work

This model can be expanded to work with more features and data to provide a more accurate calculation of bike availability. Our training data consisted of the whole network of the SF bike share, so using training data at the station level might provide more accurate results.

We also considered using elevation data since elevation is correlated with bike availability, however we were not able to get accurate elevation and location data to use. As technology progresses and this sort of accurate data becomes available, it might be another feature that can help this model become even more accurate.

## 7. CONCLUSION

Bike share programs are becoming increasingly popular throughout the world, however they provide unique challenges. Since users of bike share programs check out bikes at a specific station and drop those bikes at a different station, bike supply can become unbalanced with some stations being drained of their bikes and not being able to live up to demand. By predicting bike availability at a station, bikes can be redistributed to stations that need them.

Over the course of this project we have learned how to utilize prediction models. More specifically, we have performed feature selection, cross-validation, and model evaluation to predict bike availability and assess our model. Our model uses XGBoost to predict bike and station availability based on a number of features such as day, time, and weather conditions. Through sensitivity analysis, we were able to detect overfitting, and made training adjustments accordingly. We found that XGBoost performed slightly better at predicting bike availability than the Decision Tree and Random Forest models, which we deemed baselines for comparative analysis. Lastly, we discussed some implications of these lessons to guide directions of future work.

**RESOURCES**

[1] Dunn B., Moran M., Munowitchk M. (2020, June 2). *Biking by the Numbers: San Francisco's 2019 Biking Statistics.* San Francisco Municipal Transportation Agency. https://www.sfmta.com/blog/biking-numbers-san-franciscos-2019-biking-statistics#:~:text=Learning%20to%20Share%3A%20Over%201.8,than%20the%202019%20monthly%20average.

[2] Editorial Team. (2020, September 1). *COVID-19 Puts Bike-Sharing in the Fast Lane.* Street Light Data. https://www.streetlightdata.com/streetlight-bloomberg-nef-bike-share-covid-19-trends/?type=blog/

[3] Martucci B. (2020 January 26). *What is Bike Sharing? How Bike-Share Programs Work, Pros & Cons.* Money Crashers. https://www.moneycrashers.com/bike-sharing-best-bike-share-programs/

[4] Yang, Zidong, et al. "Mobility Modeling and Prediction in Bike-Sharing Systems." *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, 2016, doi:10.1145/2906388.2906408.

[5] Yang, Hong, et al. "Use of Deep Learning to Predict Daily Usage of Bike Sharing Systems." *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2672, no. 36, 2018, pp. 92–102., doi:10.1177/0361198118801354.

[6] Almannaa, Mohammed & Elhenawy, Mohammed. (2019). Dynamic linear models to predict bike availability in a bike sharing system. International Journal of Sustainable Transportation. 14. 1-11. 10.1080/15568318.2019.1611976.

[7] H. I. Ashqar, M. Elhenawy, M. H. Almannaa, A. Ghanem, H. A. Rakha and L. House, "Modeling bike availability in a bike-sharing system using machine learning," 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Naples, Italy, 2017, pp. 374-378, doi: 10.1109/MTITS.2017.8005700.

[8] Gossett, Stephen. "Bike-Sharing Rebalancing Is a Classic Data Challenge That Just Got a Lot Harder." *Built In*, builtin.com/data-science/bike-share-rebalancing.

[9] Dwivedi, Rohit. *Introduction to XGBoost Algorithm for Classification and Regression*. 18 Jan. 2021, www.analyticssteps.com/blogs/introduction-xgboost-algorithm-classification-and-regression.

[10] Hammer, Ben. *SF Bay Area Bike Share* (2). *Kaggle,* 14 Nov 2019, Web 3 March 2021. https://www.kaggle.com/benhamner/sf-bay-area-bike-share