CitiBike Trip Duration Predictor: Final Report

Allison Grimsted, Callie Busch & Jasmine Ng.

Problem Specification and Evolution

Bike ridership in New York City has increased due to recent policies and partnerships implemented to meet Mayor Bill de Blasio's goal of increasing bike trips to 6% of trips in the city by 2020. A contributing factor to the rising popularity of biking was the NYC Department of Transportation's contract with CitiBike. Since its inception in 2013, the platform has grown, with CitiBike averaging 80,000 trips per day (Source). In order to support and sustain green bike share systems, it is critical to analyze the data collected on the bikes to optimize operations.

We analyzed a dataset of Citibike logs from 2018 to determine what factors influence trip duration. Initially, we set out to construct a model that would predict trip duration based on information collected when bikes were checked out of docks. The intended application was predicting if a trip would be excessively long (greater than two hours), in order to flag that bike as at risk of being stolen. A warning message could be sent to the user, reminding them of the consequences of bike theft. We first attempted this as a regression problem, then as a classification. We soon realized, however, that extreme outliers in our dataset and lack of examples of long trips made our initial models extremely inaccurate. Furthermore, a model like this could perpetuate social biases. If bikes were often stolen from certain stations, for example, in neighborhoods with a lower socio-economic status, our model could flag other trips taken from that station as risky more often, and discourage people from these communities from biking.

We then pivoted our focus to predicting trip duration as an indicator of where a bike may be returned within the city. (For example, if a bike is only out for 5 minutes, it cannot reasonably travel more than 1 mile.) Citibike currently has a map that shows the public how many bikes are at any given station, but this prediction could help build a model to predict if a bike will be available to a user within 10, 20,..., 60 minutes. This application would be similar to transit apps that tell users when buses will be coming to stops near them. This will save users time, since they can better decide if it is worthwhile to wait for a bike or to take a different transit method.

Dataset

We constructed out dataset by compiling CitiBike data from CitiBikeNYC.com. The data was downloadable by month, with each .csv file containing 1–4 million rows. Using the entire population of rides in 2018 would yield a dataset too large to manipulate on a PC. Therefore, we instead took a random sample of 50,000 rows from each month and appended them together to build the final dataset with 600,000 rows. Since the data doesn't have a field for customer ID, selecting a random sample of rows helps to ensure that the sample was representative of all rides and users.

The following features were extracted:

Column Name	**Data Type**	**Description**
Trip_duration	Integer	Dependent variable; a nonnegative real number representing the time of how long a bike has been out of dock (seconds).
Start_month	Integer	A numerical feature between 1 and 12. This is the month when a bike has been taken out for a ride.
Start_hour	Integer	An integer from 0-23 that represents the hour of the day when the bike is taken out of the dock.
Start_station_id	String	Nominal value; it is the geographic feature of our model. This indicates where the bike was checked out.
User_type	String	This is a binary variable. The user is either a single-time "customer" or a "subscriber".
Birth_year	Integer	An integer between 1918 and 2002, which denotes the year in which the user was born.
Gender	String	A nominal variable with three options: male, female, and unspecified.
Week_day	String	Nominal variable. Weekends correlate with different behavior than weekdays.
Total_precipitation_inches	Float	Total precipitation in inches during the day a bike was checked out.
Average_temperature_farenheit	Float	This feature indicates the average temperature in fahrenheit during the day a bike was checked out.
Total_snowfall_inches	Float	Indicates the total snowfall during the day a bike was out of dock.
Median_rental_price	Float	The median rental price in the zipcode of the start station.

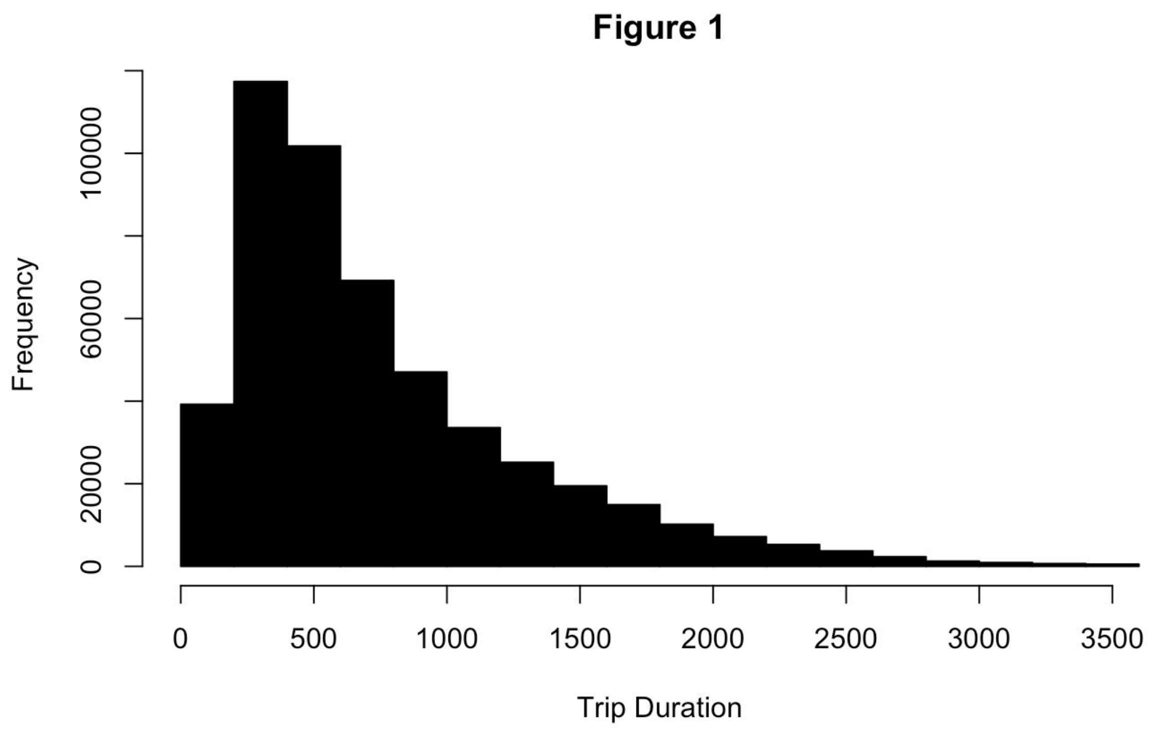
Table 1: Descriptions of features in our dataset

To clean our dataset, we cast all variables to their appropriate type (displayed in Table 1); nominal values such as gender were one-hot encoded. There were data points with birth years before 1918, which seemed implausible. We deduced that a user who would lie about their age may lie about other demographic data, and removed these unreliable rows from the dataset. We also deleted redundant features such as longitude, latitude, and zip code because they confound with start_station_id, which was more granular. Overall, we removed 2,220 rows out of 600,000.

In addition to the features given by CitiBike, we added weather and real estate data from Weather Source via DataTiles, a product from Nitrogen.ai. We joined total precipitation, average temperature, total snowfall, and median rental price on zip code and trip date from the original dataset. Using the longitude and latitude of the starting station, we used Google's Geocoding API to reverse geocode the coordinates. After one hot encoding all the features, we had 839 predictor variables and an offset.

Exploratory Analysis: Plots and Feature Exploration

Figure 1 shows the distribution of trip duration, our response variable. Most of the data lies under 3600 seconds.



Figures 2-6 show the relationships between our predictor variables and trip duration. Please note that for readability we are only showing trips with a duration of less than one hour. (This encompasses over 99% of the data.)

Figure 2

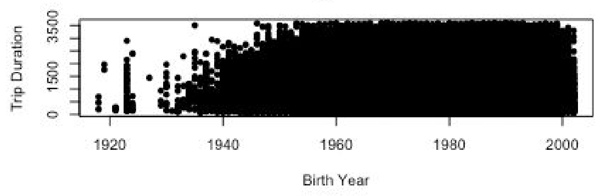


Figure 3

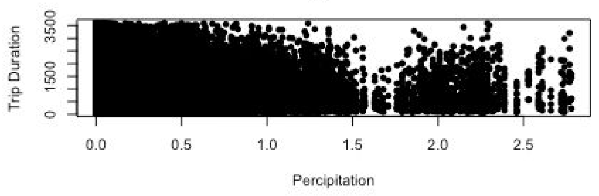


Figure 4

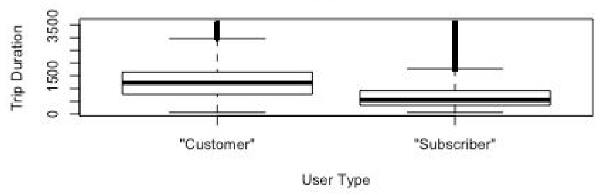


Figure 5

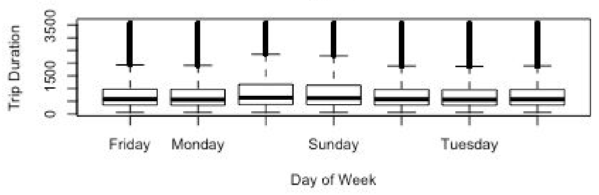


Figure 6

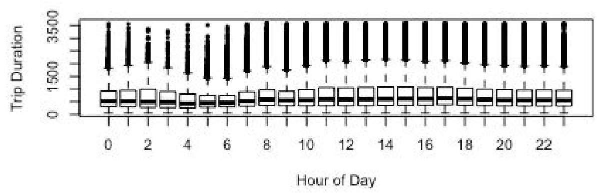
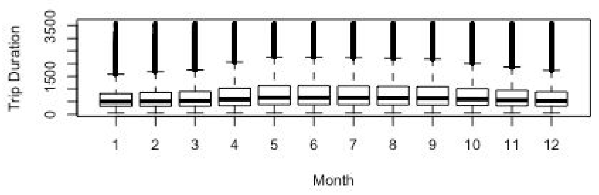


Figure 7



The graphs above show the most promising relationships between trip duration and our predictors. No numerical features have a linear relationship with our outcome variable, and nonlinear relationships seem weak. For example, we see a quadratic relationship between start month and the mean trip duration where there is less variability in the colder months. However, there are still many outliers.

The scatterplots show the relationships between the precipitation in a day (in inches) or the rider's birth year, and the trip duration. Both scatter plots show a fan pattern. Users are more likely to take longer bike rides when there is limited precipitation. A tree or step regression model could potentially predict shorter trips if precipitation is greater than 1.5 inches that day. Even still, it appears that the spread of the variables within these groups would result in high error rates. The categorical predictors have slightly better separation, but still have broad ranges. This may raise an issue with how well our models will perform.

Analysis

1. Regressor including entire range of trip durations

First, we built regression models to predict trip duration, including trips that were as long as over 2 days. We took the real valued features and one hot encoded the categorical variables for the initial model and added a squared term for month and hour because of a quadratic trend in the plots above. The mean absolute error of this model is 11 minutes and 22 seconds. We also built a tree regressor which yielded an even worse result: 12 minutes.

2. Classifier to find trips greater than 1 hour

After investigating the large test set error from approach 1, we determined that having extreme outliers in our dataset was one of the main contributors to poor performance. Our approach evolved to answering a slightly different question: can we predict if a trip will simply be long or short? This cutoff fits in our model because Citibike users will be fined at least $5 for riding more than an hour. Since this is a classification problem, our next step was to use different loss functions to build classifiers.

Since the response is boolean, the ideal loss functions are hinge loss, logistic loss, and weighted hinge loss. Therefore, we made models with these three approaches. In addition, we ran models that added L1 and L2 regularization. We also coded a logistic regression model with a quadratic regularizer and a logistic loss since there are many outliers in our data. A logistic regression model tends to have low variances and is robust to noise in the data. We decided to use a threshold of 0.5. The last model is a quadratic regularizer with hinge loss, making this a Support Vector Machine (SVM). SVM could help with overfitting and provide a safety margin on the classification boundary.

The misclassification rates from all the models are similar to each other and close to 0. The exploratory analysis helps explain this phenomenon. Only 0.8635% of the data points are 1 hour, so our model typically misclassifies a majority of this small population. Our loss function model would work significantly better if we had a dataset that had more information about long distance trips.

We built models using cross validation with three folds for logistic regression with no regularizer, a quadratic regularizer, and an L1 regularizer, and for a model with hinge loss and a quadratic regularizer, and found that all models select the sample that happens to have no trips greater than an hour in the test set, meaning the classifier assigns everything as a short trip. Because of this consistent trend, we decided that cross validation was not a worthwhile technique for our dataset.

Despite our misclassification rates being close to zero, the number of false positives and negatives outweighed true negatives (See confusion matrix for our tree classifier below), and we concluded that we could not accurately predict whether a trip would be long or short. Thus, we narrowed the problem down to only consider trips less than an hour, and considered trips over an hour as outliers. This type of prediction is still significant for Citibike because bike rides are limited to 30 minutes for day passes and 45 minutes for annual membership passes. There are fees for every extra 15 minutes used (Citi Bike Membership & Pass Options).

	Predicted Positive	Predicted Negative
True Positive	114351	1088
True Negative	881	90

Table 2: Confusion Matrix for binary tree classifier.

3. Regressor predicting trip duration when trips were under 1 hour.

To predict trip duration on short trips, we built both linear models and tree based regressors. We also used a random forest model, which builds multiple decision trees, and averages results. This article details how to build random forests (Breiman). Error metrics are as follows:

Model	**Mean Absolute Error (Min.)**
Decision Tree	10.05
Random Forest	7.350
Linear Model with Feature Selection and Quadratics	6.733
Linear Model with Feature Selection, Quadratics and Interactions	6.717
Random Forest, Optimized Parameters, Feature Selection	6.542

Table 3: Mean absolute errors of corresponding models

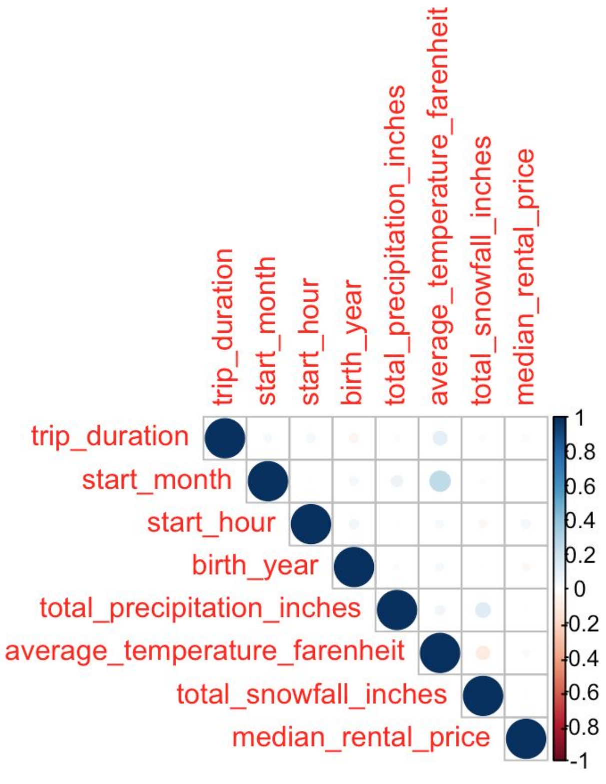
Our best performing model was a random forest with optimized parameters. We used a random grid search to determine the best parameters for the tree. The train MAE was 6.20 minutes, which tells us that this model is not overfit. We sourced our optimization code from this Medium article (Koehrson). Each feature and normalized importance are as follows:

Features	**Normalized Coefficients**
average_temperature_farenheit	0.230666
birth_year	0.206453
median_rental_price	0.167736
user_type	0.141789
start_hour	0.125079
total_precipitation_inches	0.079813

start_month	0.048464

Table 4: Normalized coefficients of the features using the Random Forest model

Discussion of the Statistical Significance of the Variables in the Data



Even though most of our variables, quadratic terms, and interactions were "statistically significant," (in quotes because the large training sample would make practically anything significant, further, at least for the linear models, many basic assumptions were broken.) we still had a maximum performance score of 0.10. This means that only 10% of the variability in trip duration was explained by our features. Simply put, our feature matrix, regardless of manipulation, did not contain enough information to accurately predict the variability within trip duration. (This is further proven by the fact that the test and training set errors were within 30 seconds of each other for our final models.) Perhaps having access to more user-specific variables such as a user's previous trip history could indicate trip duration better than weather and simple demographic data. We can further understand this by looking at a simple correlation matrix of our numeric variables with trip duration. At best, we have low correlations.

Is the model a weapon of math destruction?

The application of our model is that it predicts the number of bikes that will be available for redistribution or use for a given radius within the next hour if there is a severe shortage in some areas of the city. The outcome is measurable because Citibike already tracks where bikes are docked. Therefore, any change in the use of bikes is clearly measurable.

The model's predictions will not harm anyone because it is used to tell users how many bikes will be available near them in the next X minutes. This just provides more information to help them decide if they should wait for a bike to be dropped off nearby, similar to how transit apps tell users when the next bus is going to arrive.

The model will not create a negative feedback loop. Currently, the app shows how many bikes are at a station, providing complete, up-to-date transparency to users. However, users do not know if a bike will be arriving soon. If the model predicts that a bike is unlikely to be dropped off in a certain area, it can tell users, so they do not have to wait and become more dissatisfied when a bike doesn't arrive. The lost business due to riders discounting empty docks already exist in the system, so our model will only increase practical usability.

Is the model fair?
We determine demographic parity by examining the error of the model for different demographic groups. In this data set, the most prominent demographics are birth year (age) and gender. We found that the mean absolute error of the final model is 422 seconds for women and 387 seconds for men, which means that the model is more accurate for men, on average, by 35 seconds. However, the distribution of trip times for women has more spread than the trip durations for men (see figure 6). The demographic disparity for gender certainly is not ideal. In future models, we may try not training on gender and then see how the model performs comparatively on those two genders.

Conclusion
Our best answer to the predicted trip duration of a Citibike trip is our Random Forest with optimized parameters model because it produced the lowest misclassification metric (mean absolute error): 6.542 minutes. The linear model with feature selection/quadratics and the linear model with feature selection/interaction have close misclassification errors, respectively 6.733 and 6.717 minutes. If we were to recommend a model, we believe that the Random Forest with optimized parameters would perform the best in modeling trip duration. While the Random Forest model would take a long time to train, we don't think that this is a barrier if we avoid retraining the model after each trip.

If we were to use our final model to build an app that informs riders of an incoming bike being docked at their location, our predicted time would have an average error of 6.542 minutes. This isn't terrible in the real world sense, considering that bus apps aren't extremely accurate either. However, we believe a 6.542 minute error within a predictive range of 60 minutes is too high of an error to implement. Therefore, we currently do not trust our model to predict trip duration, but the model has lots of room for improvement.

Based on our Random Forest model, we identified the most important features as shown in Table 4. We were able to determine that the age of our riders and the average temperature have the most predictive powers in predicting trip duration. Though these features have the most influence among the features in our dataset, we think our data analysis would improve significantly if the dataset had contained features that were more personalized to the riders.

References

Bates, D. M., & Watts, D. G. (1988). *Nonlinear regression analysis and its applications*. New York: John Wiley & Sons. https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470316757.fmatter

Breiman, Leo. “Random Forests.” *Department of Statistics*, University of California at Berkeley, 15 June 2004, www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm.

Citi Bike, Motivate International, Inc, July 2019, www.citibikenyc.com/.

Koehrsen, Will. “Hyperparameter Tuning the Random Forest in Python.” *Towards Data Science*, Medium, 10 Jan. 2018, towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74.