Where are Blue Bikes Going?

Audrey Nichols

December 8, 2024

Abstract

This paper investigates the usage patterns of blue bike transportation systems through the analysis of a large dataset containing over 8 million bike rides. We employ various machine learning techniques, including k-nearest neighbors (KNN), linear regression, and random forests, to model and predict ride duration. By comparing the performance of these models, we identify the strengths and limitations of each approach in forecasting demand and optimizing resource allocation within the bike-sharing network.

Key words: blue bikes, transportation, knn, linear regression, random forest

1 Introduction

Public transportation is a key piece of infrastructure of any great city. Over time, many cities have grown from having buses and trains as the key sources of transit, to adding bicycles, ebikes, and even electric scooters into the mix.

This project was born out of the inspiration of public data, and taking transportation everyday. I was curious what sort of things we can predict using the publicly available blue bike data, and what insights may be available to benefit the larger infrastructure of public transportation in Boston.

For this analysis, we will be trying to predict the duration of blue bike rides based on start and end station locations.

2 Data

2.1 Source of Dataset

The data that was used for this project was sourced from the official Blue Bikes website. The data consists of a single row for every ride taken during a month of operation. After the end of each month, each dataset is uploaded to the Blue Bikes website at bluebikes.com/system-data. For this project, all of the data available from 2023 and 2024 was utilized.

2.2 Characteristics of the Dataset

In 2023, the Blue Bikes system introduced electric bikes, causing an update to the format of data collection. Due to this and the large number of files, the data required a large amount of cleaning before they were ready to use.

2.2.1 Data Cleaning

In order to clean the data, we first must approach the task of converting all of the data to the same format. All of the files came as .csv, but column names and data collection changed in 2023 with the introduction of electric bikes, so we had to account for that in our data cleaning. Any files that came prior to the introduction of this new field were given the new column 'rideable_type' and assigned 'classic_bike' to easily be merged with the remaining data files.

Once all files were merged, some were missing a 'ride_id', so the column was deleted and new randomly generated ride id was assigned to each row. Finally, in order to finish the data cleaning, five new columns were added, 'year', 'month', 'day', 'time', and 'duration_min'. All five columns were parsed using the original 'start_time' column, in order to determine the individual characteristics of the starting time of the ride and the total duration of the ride in minutes.

The total duration of the ride was calculated by converting the ride start and end times into minutes, and computing the difference.

Once cleaned, the concatenated data set has 8254372 rows and 18 columns. Each row contains the data for a single unique ride; the following table contains the column names and descriptions of each variable.

Variable Name	**Description**
ride_id	unique ride id
rideable_type	type of bike
started_at	start time of ride
ended_at	end time of ride
start_station_name	name of start station
start_station_id	unique id of start station
end_station_name	name of end station
end_station_id	unique id of end station
start_lat	latitude of start station
start_lng	longitude of start station
end_lat	latitude of end station
end_lng	longitude of end station
member_casual	rider membership status
year	year of ride
month	month of ride
day	day of ride
time	start time of ride (hh:mm)
duration_min	duration of ride (minutes)

3 Methodology

The different types of modeling that we will be using to predict ride duration is K-Nearest Neighbor, Linear Regression, and Random Forest.

For each of these models the response variable *duration_min*, and the predictor variables were *start_lat*, *start_lng*, *end_lat*, *end_lng*.

3.1 Libraries

The following libraries were used for plotting and analysis;

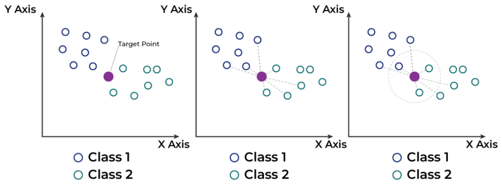
Library	**Use**
numpy	multi-dimensional arrays
pandas	read/write data
matplotlib	creation of graphs
seaborn	plotting assistance
scipy	algorithms + calculations
google.colab	connect to Drive

3.2 K-Nearest Neighbors

K-Nearest Neighbors is a supervised machine learning method used for classification and regression.

K-Nearest Neighbor was chosen as one of the models for this project due to its simplicity and versatility.

K-Nearest Neighbor is advantageous to analysis because it is easy to implement. Some of the negative sides to choosing the K-Nearest Neighbor model is that it is prone to overfitting, which can be seen when a model outperforms classification of the training data compared to the test data.



The K-Nearest Neighbor algorithm determines the value of a target variable by finding the k nearest points of data.

3.3 Linear Regression

The second model that will be used is Multiple Linear Regression, as more than one variable will be used.

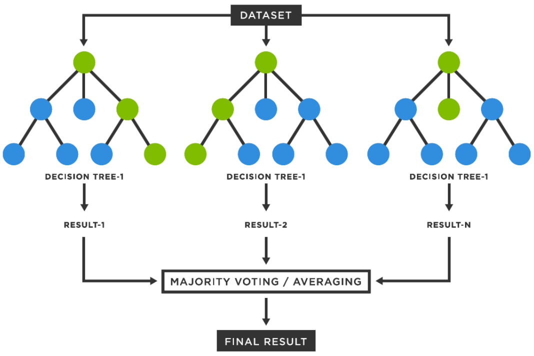
Often, Linear Regression is not the strongest model compared to other supervised machine learning models, but it is a great model to use as a starting point.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

The algorithm for multiple linear regression shows the response variable y, as a function of the predictor variables $x_1, x_2, etc.$ and their coefficients $\beta_1, \beta_2, etc.$ and the intercept, β_0.

3.4 Random Forest

The Random Forest supervised machine learning method uses decision trees to perform classification and regression.



This diagram shows us the process behind a random forest classification, different samples of our dataset are run to determine a majority classification or regression of a variable.

4 Results

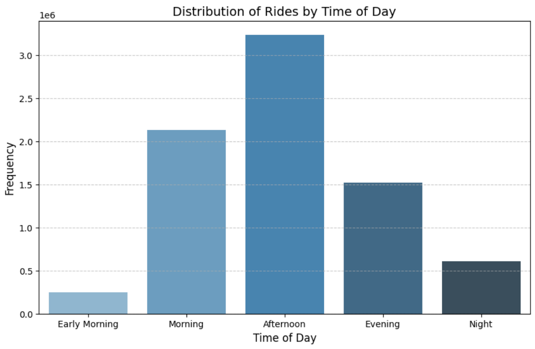
To implement the models, a sample of the data set was taken because it is so large, to avoid overfitting and also decrease runtime and number of computation.

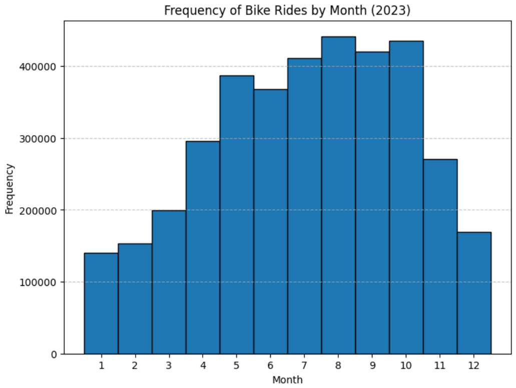
Outliers were also removed from the data set by determining inter-quartile range (IQR) and removing any values above the upper bound and below the lower bound as determined by IQR.

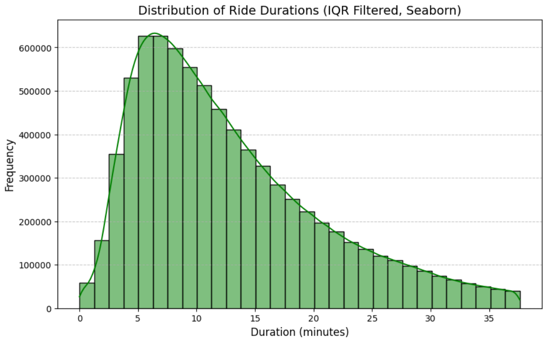
Each model was run using the same sample, and testing and training sets. Before the models were run some data exploration was completed to discover more about the data set.

4.1 Data Exploration

To begin the data exploration, we want to know more of our data set. The two following charts display Frequency of Rides by Month, and by Time of Day, respectively.







Since the chosen response variable is *duration_min*, we want to plot the distribution of our variable to determine if it is a good variable for prediction.

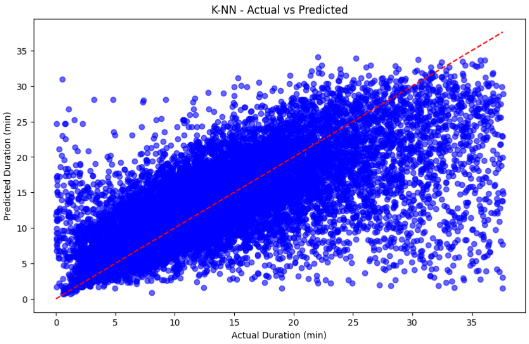
4.2 K-Nearest Neighbors

The K-Nearest Neighbors model was fairly strong, and did an okay job of predicting the test data set.

Metric	Train	Test
MSE	22.154	32.691
R^2	0.6488	0.4886
MAE	3.157	3.839

Looking at our table of metrics, since the R^2 and MSE values are higher for our training set, we can assume that the model is experiencing overfitting. However, we can attribute almost 50 percent of the variability in the data to the model.

Let's see what our scatterplot looks like:



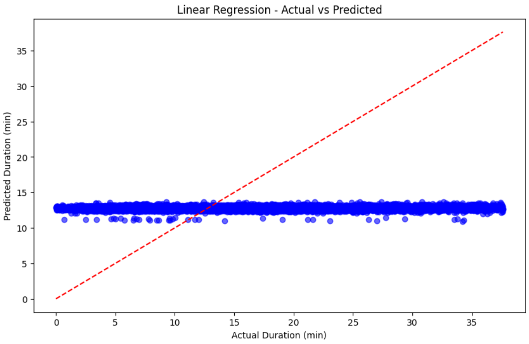
We can see the model working, as the heaviest concentration of data is around our line of best fit, but there are many data throughout the graph that are far from this line.

4.3 Linear Regression

The Linear Regression model is weak, failing to correctly predict many of the data.

Metric	Train	Test
MSE	63.047	63.890
R^2	0.0006	0.0006
MAE	6.356	6.409

This time, our MSE is much higher than in the KNN analysis, and the R^2 is practically nonexistent. This model is extremely weak, as can be observed in the scatterplot as well:



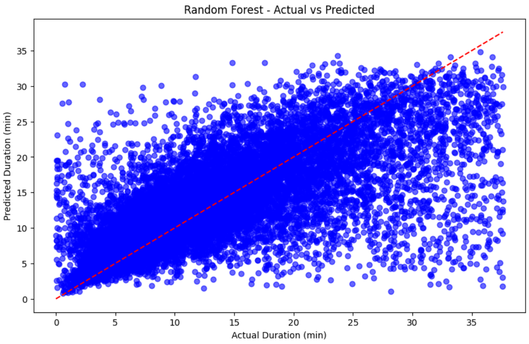
4.4 Random Forest

Finally, Random Forest, this model was the best fit for determining our response variable.

Metric	Train	Test
MSE	11.671	33.082
R^2	0.8149	0.4825
MAE	2.126	3.838

Like our KNN model, we are suspect of overfitting, due to the R^2 and MSE being much stronger for the training dataset.

Looking at our scatterplot, we can also see that the Random Forest model performed just a bit stronger than our other two models:



5 Discussion

If given the chance, I would attempt to restructure the data in a way that made more use of the variables, rather than keeping them in their raw formats.

A large challenge to this project was working with such large datasets, as often google colab was unable to handle the runtime, and would often restart causing me to have to start from the beginning as well.

6 Conclusion

Overall, the best fit model for this data set was determined to be the Random Forest model, although both the Random Forest and K-Nearest Neighbors models were overfitting the data.