Predictive analysis of the MBTA Commuter Rail using Linear and Random Forest regression

Kyle Shott | github.com/kyshott  
*Wentworth Institute of Technology School of Computing and Data Science*

*Abstract*—This paper is an analysis of the MBTA (Massachusetts Bay Transportation Authority) Commuter Rail lines and the various factors that play into the changes in ridership and reliability. Specifically, the effects of COVID-19 on ridership, the increase/decrease of reliability across all rail lines over time, and the average ridership trends were all analyzed as part of this project. Using the Python programming language, a variety of visualizations and prediction models were developed to provide insight into these topics using data pulled directly from the MBTA open data source.

Keywords—Machine learning, Data science, Linear regression, Random Forest regression, MBTA

# Introduction

The MBTA Commuter rail is used daily by 100,000 people and is relied on to be a reliable and consistent public transport system. However, many express frustrations with the commuter rail due to incorrect track arrivals and information, unforeseen stop delays, and a lack of “real” information about schedules and arrivals for those unfamiliar with the commuter rail. By using the readily available, (and very large) sets of data provided by the MBTA on their open data site, an analysis can be conducted to provide further insight into multiple things. The effect of COVID-19 on Commuter Rail ridership, the effect of time on the reliability of Commuter Rail lines, and the predicted ridership for the 2025 are all analyses conducted in this paper, with two of which being backed by low-level machine learning models.

Section II will cover the datasets used in this project, as well as the characteristics of and methods used to munge the data. Section III explains the methodology in developing the models and the tools/formulas used. Section IV contains the results in using the predictive models, accompanied by visualizations. Section V contains discussion about these results and the benefits and shortcomings of the models, and Section VI contains the conclusion of the findings as well as the references used for this project.

# Datasets

## Source of datasets

The dataset(s) used were all fetched through the MBTA’s official “Blue Book” open data portal, where historical and real time metrics are updated daily. In this portal, there are numerous sets of data in varying formats available for download by the public. For the purposes of this analysis and model development, there were two key datasets pulled from the portal:

* A CSV file containing the reliability for *every single* trip done by all 13 MBTA Commuter Rail lines, starting from 2016 to the present.
* A CSV file containing the monthly ridership for all MBTA Commuter Rail lines, starting from 2016 to the present.

It is important to note that the MBTA uses a large variety of methods for collecting its data. However, when it comes to the Commuter Rail, there aren’t any automated systems to tally data, so most of the Commuter Rail data is typically collected from conductor counts and rider surveys. This means that the data for the Commuter Rail may *not* be as exact as the subway system due to human error, but it is still credible.

## Characteristics of the datasets

Thankfully, all data collected was in CSV (comma- separated value) format, which is particularly easy to format and work with in the context of Python programming. For example, the first data set used, titled “commuterridership2024.csv” was in a format that, when imported into a software such as sheets or excel, would look something like what is shown in Table I:

1. snippet of ridership csv file

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Month / Year | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024 |
| January | 134653 | 126517 | 118585 | 120964 | 114518 | 13530 | 36496 | 82041 | 107322 |
| February | 124534 | 124570 | 120498 | 121856 | 114241 | 11930 | 44531 | 80625 | 102957 |
| March | 134126 | 119236 | 116213 | 121569 | 45602 | 13498 | 57827 | 82163 | 108131 |

Iterating through this data was quite easy and was done using a Pandas data frame in Python. Pandas is a highly intuitive Python package that allows for data to be converted into a workable column/row format, which works perfectly for reading data from a CSV file. The ridership data itself was not aggregated or combined in any way and was instead used as-is for a Random Forest regression model. The other data set, titled “commuterreliability.csv” CSV file, totaled at around 80,000 lines, and was formatted in a way as seen in Table II:

1. snippet of reliability csv file

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| service\_date | gtfs\_route\_id | gtfs\_route\_long\_name | gtfs\_route\_desc | mode\_type | peak\_offpeak\_ind | metric\_type | otp\_numerator | otp\_denominator |
| 2024/10/31 04:00:00+00 | CR-Franklin | Franklin Line | Commuter Rail | Commuter Rail | OFF\_PEAK | Headway / Schedule Adherence | 14 | 18 |
| 2024/10/31 04:00:00+00 | CR-Fitchburg | Fitchburg Line | Commuter Rail | Commuter Rail | PEAK | Headway / Schedule Adherence | 8 | 10 |
| 2024/10/31 04:00:00+00 | CR-Franklin | Franklin Line | Commuter Rail | Commuter Rail | PEAK | Headway / Schedule Adherence | 6 | 8 |

Here, it can be observed that the data is also quite easy to iterate through, as it is also in CSV format. The particularly important parts of this data lie in the otp\_numerator and otp\_denominator columns, with the combination of the two giving a reliability “score” for that Commuter Rail line for that day. For example, in the first row, for the Franklin Line, it scored a 14/18 for January 31st, 2024. This data is particularly valuable, as it gives insight into just how reliable each Commuter Rail line really is, and whether the passage of time or time of year has any effect on their reliability scores. This data did take a fair bit of manipulation – specifically in aggregating the reliability scores. An extra “year” column was added to the data to allow for easier sorting by year. Then, an extra column “reliability score” was added, which was a calculation of the otp\_numerator divided by the otp\_denominator. Then, all data for each year for each line was aggregated and given to a linear regression model to generate a prediction for the reliability over time.

# Methodology

The Python programming language was used for all methods explained in this section. In addition to Python, a variety of “packages” or modules were used, such as NumPy, Pandas, Matplotlib and Scikit Learn. NumPy was used for array/list manipulation, Pandas was used for its useful “data frame” data structure for importing and manipulating CSV file data, Matplotlib was used for all graph visualization, and, most importantly, Scikit Learn was used as the basis for the predictive model training.

## Linear regression

Linear regression is a technique used in predictive model training to create what is known as a “line of best fit” in a set of potentially scattered, but still somewhat linear, data. It can be a useful tool in determining trends for data that has a consistent and seemingly linear relationship between variables. The Python module used for this particular model training was Scikit Learn’s “sklearn.linear\_model.LinearRegression”. Scikit Learn’s linear regression model is particularly intuitive since it allows for variable pairs to be “fit”, then “predicted” using its built-in functions. The method used for the model’s predictions is what is known as “Ordinary Least Squares” (OLS) regression. The formulas are as follows:

In (1), y is the vector of predicted values, X is the matrix of input features, (arguably the most important part of the equation) is the learned coefficient (that is, the coefficient that characterizes the linear relationship between variables), and being the error of the model, or the difference between the predicted values and actual values.

(2) is a representation of how the vector of predicted coefficient values, , is obtained. The transpose of the feature matrix, XT, is multiplied by its non-transposed self, X, and the inverse of this product is then once again multiplied by XTy, the product of the transposed feature matrix by the vector of observed (actual) values. This vector of coefficients is then brought in to (1), where , the vector of predicted values according to the line of best fit, can be obtained.

In the context of this project, X is the matrix of values obtained from the CSV files, and is ultimately the coefficient that is used to predict and create the line of best fit. This model was used to determine and predict the reliability of each Commuter Rail line over time, and, ultimately, for 2025. The reason Linear Regression was used in this case is specifically because the reliability for each Commuter Rail line seem to follow a linear trend, either being mostly consistent, worsening, or getting better with reliability over time.

## Random Forest regression

Random forest regression is a distinctively different method from linear regression, as it is typically suitable for large, single data sets with complex relationships. Simply put, Random Forest regression splits the data given into smaller “decision trees”. This allows for data that may have “before/after” or other features that would be difficult to implement via linear regression to be processed and predicted in a far more accurate manner. The Python module used for this model training was Scikit Learn’s “sklearn.ensemble.RandomForestRegressor”. Just like the linear regression module, Scikit Learn’s Random Forest regression model only requires the data to be passed in to its functions as parameters to make predictions. The formulas used by this model are as follows:

(1) represents the formula for individual decision trees. Specifically, N is the number of samples within the current node in the tree, yi is the actual value within the subset of data, and is the mean of the target or predicted value. For all nodes in each decision tree, this formula is used. What makes this particularly useful is that each tree ends up with its own prediction, which means that separate trees can be made for separate features of the data.

(2) is the next and final formula of the Random Tree regression process, and summarizes all decision trees, where T is the number of trees and is the prediction for the -th tree in the predicted tree data from (1). (2) is what will generate the final predicted value for any value in the original data set.

In the context of this project, the CSV file data mentioned previously is split into 100 trees, with splits in each tree for features based on Month, Year, and whether the ridership data point was pre-COVID-19 or post-COVID-19, as COVID-19 had a significant effect on ridership (see results). The data used in these trees was split into four categories, x\_train, the feature matrix for training, y\_train, the actual data corresponding to the x\_train data, x\_test, a feature matrix for testing, and y\_test, the actual data corresponding to x\_test. 20% of the data was set aside for testing, while the other 80% was used for training. The reason in doing this is to test the efficiency of the model and see how far off the predictions are (see next section). The reason this model could be used in the context of the data is that the ridership data is aggregated across all lines, meaning there was no need for specification of Rail Lines, as was done in the linear regression model. Therefore, the data here can be considered complex due to the large number of factors that play into yearly and monthly ridership.

## Mean Absolute Error and R2

Some important metrics to note regarding this analysis are the mean absolute error (MAE) and the R2 value of a predictive model. Mean absolute error demonstrates the average error the model has in predicting values. So, a lower MAE is desirable in this case since it means that the model is very close to the actual data. The R2 value demonstrates how well the model displays the correlation between the features and predicted values. What this means is that, the closer the value is to 1, the more of an effect the model’s features had on the predicted data. If the value is closer to 0, then the features do not have much of an effect on the data.

A graph of different colored lines

Description automatically generated

In the following section, both values are referred to as MAE and R2, and are used to display the effectiveness of the models used as well as make conclusions about specific parts of the datasets. In both models, a smaller MAE is always desirable, as it means the model used is highly accurate; however, the R2 value is not inherently meant to be higher or lower but is instead analyzed to make well-informed conclusions regarding the data.

## Matplotlib and Visualization

Finally, for the visualization of all graphs and charts in this project the matplotlib Python module was used. Matplotlib is a very easy to use and effective module for taking any form of data (in this case, a Pandas data frame) and converting it into a easily understood visual format. Taking being conciseness and its relevance of the topic of visualization to the project, all explanation for the visualization process using matplotlib will be omitted from this paper.

# Results

To start, it was deemed necessary to generate some basic visualizations about the data to provide the necessary context into what exactly the models themselves were going to be using for data. For the reliability of the MBTA Commuter Rail, the CSV file containing the reliability data was aggregated together based on the Rail Line and was generated into a bar chart to effectively “rank” the Rail Lines by reliability. Fig. 1 represents this bar chart.

A graph of different colored lines

Description automatically generated

A graph of multiple purple lines

Description automatically generated

Fig. 1. A bar chart representing the reliability “scores” for each line.

In Fig. 1, it can be observed here that the Fairmount line has remained the most reliable since 2016 with an average reliability score of .96 and the Framingham/Worcester line has remained the least reliable with a score of about .85. This data will be used later to make further postulations about the model predictions.

Next, a line chart was generated to display the difference in Commuter Rail ridership since 2016, with each individual-colored line representing a particular year. This is particularly useful, since it provides insight into what features needed to be implemented into the Random Forest regression model. Fig 2 is the line chart.

Fig. 2. A line chart representing ridership for each month by year

In Fig. 2, it can be observed that the purple line representing the year 2020 has a massive decline between the months of March and April, indicating a large decline in ridership due to the advent of COVID-19. There are also noticeable trends due to the months of the year as well, which can be attributed to school seasons, holidays, and particular line shutdowns. This graph is important to note since it provides a basis into the feature engineering that will go into the Random Forest regression model.

## Linear Regression model

The linear regression model was used to predict what the reliability for 2025 will be given the reliability data for each Rail Line, for each year. The R2 and MAE were calculated as well. Fig. 3 is the visualization for the prediction results.

Fig. 3. A line chart showing the predicted reliabilities by year

In Fig. 3, it is clear the linear regression was successful in creating linear lines-of-best-fit for each rail line. Each rail line pictured retains a linear slope, and terminates at the 2025 tick mark, showing that the model was successful in generating predictions. Fig. 4. represents the legend for the graph shown in Fig. 3.

A graph of different colored lines

Description automatically generated

Fig. 4. The legend for figure 3, representing the MAE and R2

In Fig. 4, the MAE and R2 for each individual rail line can be seen. The MAE for each line is always below 0.1, which, for the context of the data, is very good. This means that the largest error here is 4%, which means the model did a very good job at predicting the values.

The more important part here is the R2 values. In the context of reliability, a higher R2 means that the rail line is less reliable over time, meaning that it is more susceptible to change the longer time passes. For example, the Fairmount Line has an R2 of .04, which means it is very stable and almost never changes in reliability over time. This is reflected in the graph, as the Fairmount Line is a fairly horizontal line with a small slope. The Framingham/Worcester line has an R2 of .82, which is quite high and means it has seen the most change in reliability due to the passage of time, which can be attributed to its very minimal slope on the graph.

## Random Forest regression model

The random Forest regression model will be used to predict what the Commuter Rail ridership (across all lines) will be for the year 2025, given the data from the CSV file. For this general prediction, the R2 and MAE were calculated. Fig. 5 is a visualization of the predicted trend for the ridership throughout 2025.

A graph with blue lines

Description automatically generated

Fig. 5. The line chart for the predicted ridership for 2025, by month

It can be observed that the MAE for the predicted ridership is approximately 5718. The ridership for any given year is typically around 100,000 making this roughly a 6%-15% mean average error for the model, which is quite good. Even more interesting, the R2 value is approximately .93, which is very close to 1. This means that the features implemented into the Random Forest regressor did an exceptionally good job at distinguishing the variance in the data.

With that being said, the prediction here is highly accurate. Looking at the graph alone, a lot of seasons are accounted for, as well as holidays. June, which is historically the most busy month every year for the MBTA is represented to have just short of 116,000 riders, a steady increase of about 6000 riders from June 2024. This displays how the prediction accounted for the steady increase in ridership since COVID-19.

# Discussion

As with any other predictive model, the two used for this project were far from perfect. For example, in the linear regression model, many Rail Lines did not have a high R2 value at all, meaning the model did effectively nothing in explaining why or how particular rail lines had remained consistent. Had more features been implemented into the model, Rail Lines such as the Fairmount Line, with an R2 of .04 (exceptionally small and close to 0), could have better conclusions made regarding its consistency.

The Random Forest regression model had very good results, explaining the variance in data very well and staying close to the historical data. However, the prediction was only done for the year 2025, and not any further. While the insight for 2025 is suitable, there is plenty of room for further prediction and perhaps further training of the model by creating a feedback loop with its own predictions.

Moving forward, these models can be scaled for larger sets of data and possibly even more specific data. Ridership data for specific lines could be implemented as its own feature matrix into the Random Forest regression model to further increase its accuracy, and more features could be implemented into the Linear Regression model. All in all, however, the models did do a good job at providing insight into the MBTA Commuter Rail.

# Conclusion

Overall, the predictive models were successful in providing the desired insight into the MBTA Commuter Rail. With these models, an accurate prediction for the 2025 ridership, as well as accurate predictions for Rail Line reliability for 2025 have been established. The models did have shortcomings and there is plenty of room for improvement, but the insight given from them is very valuable and applicable for current or prospective MBTA riders.

##### References

1. “MBTA Bus, Commuter Rail, & Rapid Transit Reliability,” *Arcgis.com*, 2018. https://mbta-massdot.opendata.arcgis.com/datasets/b3a24561c2104422a78b593e92b566d5\_0/explore (accessed Dec. 08, 2024).
2. “Service Reliability | Performance Metrics | MBTA,” *www.mbta.com*. https://www.mbta.com/performance-metrics/service-reliability
3. “MBTA Blue Book Open Data Portal,” *Arcgis.com*, 2018. https://mbta-massdot.opendata.arcgis.com/search?q=ridership&tags=commuter%2520rail (accessed Dec. 08, 2024).
4. MBTA, “V3 API | Developers | MBTA,” *Mbta.com*, 2024. <https://www.mbta.com/developers/v3-api>