**Predicting Ridership For The Metropolitan Transportation Authority (MTA) Using A Random Forest Regression Model**

**Report By: Anthony Szydlik, Esteban Penagos, Jack Halpern, Kendall Dowdie**

# I. Proposal

## There are a lot of times when we take the subway, and it is packed, dirty, and we have long wait times to get to our destination. We wanted to see how we can improve the service and predict the busiest times. By doing this, we can see when peak hours are and when there is a need for better service. We can also see when it is empty and perhaps be able to clean the trains to offer a better service.

## With data collected from the “MTA Subway Hourly Ridership for the year 2025”, we decided to predict ridership for the state of New York. We wanted to see what the times were that the subway was the busiest, the days of the week, and the location.

## The model we are going to build is going to give us predictions on the ridership. This supervised learning model is going to help us predict the peak hours, the average ridership by hour, and the ridership distribution by days of the week.

## For the model, the target variable is ridership, which we will see when we explain and show the model. This will help the city of New York and the State to know when their busiest times are every day of the week. It will help to reduce wait times. It will also help to perhaps add more frequent subways in certain locations during their peak hours. Hopefully, with proper resources from the state, we will be able to train this model to predict ridership years in advance to help with long-term planning and sustainability. Overall, this prediction could help the city and the state to offer better MTA Subway services.

# II. Business Understanding

## A. Business Problem

The MTA faces the challenge of predicting subway ridership patterns as levels of ridership fluctuate constantly, and it is hard to predict accurately. Some of the problems that arise include overcrowded trains during peak travel hours and underutilized service at off times. (Bass, 2025) The ability to accurately predict ridership patterns would help the MTA run more efficiently. For example, being able to predict peak hours of ridership would allow the MTA to run more trains when they are needed, reduce wait times, and prevent overcrowding, which are all problems the MTA faces. Also, off times are essential times for maintenance and cleaning operations, overall contributing to operational and general safety and efficiency.

## B. Data Mining Solution

To tackle this problem, a supervised machine learning approach is the best option, and we are using a Random Forest Regressor model to make our predictions. This model will leverage data about time of day, day of the week, and station information to predict fluctuating ridership levels. The strength of Random Forests lies in the ability to capture non-linear relationships, which is crucial when we are evaluating many different factors. Predicting peak and off-peak hours will allow the MTA to make decisions for operations based on data and the relationships between that data, as opposed to general understanding and intuition. In turn, their resources will be used more efficiently.

# III. Data Understanding

The dataset used for this analysis comes from the MTA Subway Hourly Ridership for 2025, which includes detailed records for subway ridership at different stations across New York City. The dataset includes several key variables: Timestamp, which marks the specific time the ridership count was recorded; Transit Mode, indicating whether the service was local or express; Station Information, including station IDs, borough details, and names to analyze geographic trends in ridership; Ridership, the target variable representing the actual number of subway passengers at a given time; and Transfers, which reflects the number of passengers switching lines at specific stations, possibly signaling higher traffic at interchange points.

## B. Data Quality Assessment

The dataset is mostly complete, with only a few irrelevant columns (like latitude, longitude, and georeference) that were removed during preprocessing. After cleaning, we retained the essential variables: timestamp, station information, and ridership data. Categorical features, such as station ID and transit mode, were converted to one-hot encoded variables to make them compatible with the machine learning model. We also split the timestamp into meaningful components like hour, day of the week, and day of the month, which will help capture daily and weekly patterns in ridership. Assessing the distribution of the level of ridership, as shown in [Figure 9](#bookmark=id.bq2tw4ie3em1), most of the dataset corresponds to low levels of ridership. This skewness reflects our model performance which is detailed in later sections.

# IV. Data Preparation

## A. Data Cleaning

At the beginning, our data set included many columns that included time\_stamp, transit\_mode, station\_complex\_id, station\_complex, borough, payment\_method, fare\_class\_category, ridership, transfers, latitude, longitude, and georeference. Knowing that our model is seeking to accurately represent ridership, we dropped the columns that had longitude, latitude, and georeference. For better results, we also split the timestamp into separate columns, including hour, day\_of\_week, day\_of\_month, and time-period. We will also use this later for better visualizations. Once we had the dataset columns cleaned, we attempted to cut down on the number of records in the dataset to try and avoid overfitting or poor prediction results due to the immense amount of rows and columns. To do this, we sliced the dataset only to provide ridership values from January, as we can then use this model with the other months of the dataset to try and create a complete model to predict ridership for any time of year.

To prepare the data for modeling, we applied preprocessing techniques to standardize the dataset and ensure that all variables contribute equally to the prediction process. We started with categorical encoding using the pandas package's dummy to convert categorical variables (e.g., 'transit\_mode', 'station\_complex\_id', 'station\_complex', etc.) into a binary format for machine learning compatibility. This one-hot encoding enabled the model to interpret these variables without assuming ordinal relationships. We also scaled all continuous numerical features using sklearn’s standard scaler to normalize any numerical features present. This normalization ensured that the feature transfers did not skew predictions.

## B. Data Transformation

Before we were able to use our data to create a predictive model, we had to transform the data we had into more regularly available variables for model creation and visualization. We began this process by splitting the dataset into X and y variables, which we will further split later to create a training and testing set. We ensured that all variables, including the encoded categorical variables, were included. Our X split excludes ridership to ensure that the predicted values are not in the training data, and our y variable includes only ridership, as it is the desired outcome. We then took these two variables and used sklearn’s train-test split package to produce a training and testing set for both X and y variables. The training set comprises 80% of the data, and the testing set includes the remaining 20% to be used to validate the original training set. To ensure that we could reproduce these results, we chose to use random state 42.

After the train-test split, we used sklearn’s standard scalar to normalize the categorical variables, including 'hour', 'day\_of\_month', 'day\_of\_week', 'is\_weekend', 'is\_peak\_hour', and 'transfers'. We do this to ensure that the vastness of their integer values does not throw off the predictions from the model. This is especially useful for columns like ‘transfers’ where there is a large range of possible values. This will help the efficiency of the model and ensure that there is less bias.

# V. Modeling

## A. Model Selection

To predict ridership accurately based on the cleaned and preprocessed dataset, we selected the Random Forest Regressor as our primary modeling algorithm. Random Forest is an ensemble-based method that builds multiple decision trees and aggregates their predictions, which helps to reduce overfitting and improve generalization (Brownlee). This model was particularly well-suited for our dataset due to a number of its abilities. Firstly, it handles both numerical and one-hot encoded categorical features without the need for additional transformations. It also captures complex, non-linear relationships between independent variables and ridership while avoiding skewing from multicollinearity and noise within the data. Additionally, Random Forest provides feature importance measures, which allow us to assess the contribution of each variable toward the overall prediction, helping us understand their relationship with ridership more clearly.

## B. Model Comparison

Throughout the model development process, we tested a range of baseline and advanced machine learning models to determine the best-performing approach. We started by testing a simple linear regression model, which provided a useful baseline, but struggled to capture the non-linear interactions present in our dataset. This produced low R² values and higher error metrics. We also tested a decision tree model, which improved upon linear models by incorporating branching logic, but tended to overfit the training data due to its sensitivity to noise. This finally brought us to the Random Forest Regressor mode, which we found to outperform both previous models in R² values, root mean squared error (RMSE), and mean average percent error (MAPE). Its ensemble nature, alongside the ability to be fine-tuned, helped it generalize better to unseen data, making it the most reliable option for our final model.

## C. Model Parameters

We implemented hyperparameter tuning using GridSearchCV from sklearn’s model selection to enhance model performance further. We tested various combinations of the following parameters: n\_estimators: (tested 100, 200, 500), max\_depth: (tested 10, 20, None), min\_samples\_split: (tested 2, 5, 10), min\_samples\_leaf: (tested 1, 2, 4) The grid search was performed on the training data using 5-fold cross-validation to ensure the model's reliability across different data subsets.

We also used cross-validation to evaluate model consistency and avoid overfitting. A 5-fold cross-validation approach ensured that every data point in the training set had a chance to be in a validation fold. This technique provided a more reliable estimate of our model’s performance and allowed us to spot any variance issues. We tracked both R² score and RMSE to compare performance.

## D. Model Optimization

Based on the cross-validation results and grid search, we selected the hyperparameter configuration that yielded the best balance between performance and computational efficiency. The final model was trained on the entire training set using these optimized parameters.

We then evaluated the model on the hold-out test set (20% of the original data) to assess real-world predictive performance. The final Random Forest model demonstrated strong predictive accuracy, as shown in [Figure 1](#bookmark=id.3u1iamwbbme3), with a high R² and relatively low RMSE. This confirmed its effectiveness in capturing the underlying patterns of ridership behavior. We did find that the MAPE was high at around 60% in the test set shown in [Figure 2](#bookmark=id.7bc9kqn3sc), so we investigated our model further to find where this error was occurring. Referring to [Figure 3](#bookmark=id.2u1xjpsx04qu), we can see that the majority of this error is in the low ridership prediction, where the MAPE is around 100%. The MAPE for high ridership or 50-100+ is relatively low at only 35-30%. We also created a graph showing residuals from the prediction model, which shows the difference between predicted and actual values. As we can see from them in [Figures 4](#bookmark=id.vzj5gejel6cz) and [5](#bookmark=id.80cbyayaa4t7), our ridership predictions tended to skew towards high ridership. With this information, we decided to break our model into two models, one to predict low ridership and one to predict high ridership.

In addition, we generated a feature importance plot to visualize which variables had the most influence on ridership predictions in [Figure 6](#bookmark=id.tmrx2uilctvv). Features such as hour of day, day of week, station complex, and transit mode emerged as critical drivers, aligning with real-world commuting patterns and transit usage behavior.

# VI. Evaluation

## A. Model Performance

As mentioned in the Model Optimization section, with our prediction model skewed towards high ridership, we decided to craft two different Random Forest Regressor models, one to predict low ridership and one to predict high ridership. To create the groups, we used K-Means clustering to automatically cluster the data into two groups, low and high ridership. Doing this allowed us to achieve better model performance. As shown in [Figure 7](#bookmark=id.k2ja2g7qr4z), the low ridership model performed well with a high R² value of 0.76 on the test set. There is room for improvement of our model in terms of the error metrics. Although we were able to significantly decrease the RMSE in the low ridership model, suggesting a slightly better fit model, it is still a relatively high value and suggests there are still errors in the prediction model. Taking a closer look at the performance of our low ridership model, there might be some overfitting in our model, as the training set performance is slightly higher than that of the test set. Despite this, our prediction accuracy ranges are strong for the low ridership model. There is nearly an 80% prediction accuracy within 10 riders.

Although creating a low ridership model created a better prediction model for predicting low ridership, there is a significant amount of room for improvement for our high ridership model. As shown in [Figure 8](#bookmark=id.9il626x8u7uo), although we achieved a high R² value for the performance of the training set in the high ridership model, when the model was used to predict unseen data, the R² value significantly decreased to 0.35. Additionally, the high ridership model has a significant amount of error with an MAE of 217.23 and an RMSE of 276.94, suggesting that the predictions deviate significantly from the actual ridership value. Similarly to the low ridership model, this high ridership model is likely overfitting as it has better performance on the training dataset compared to the performance on the testing dataset. Lastly, compared to the low ridership model, there was a significant decrease in prediction accuracy with the high ridership model, with only 3.33% of accurate predictions within 10 riders.

To improve the accuracy of our low ridership and high ridership model and make it a more reliable model for MTA, the first course of action would be to address the overfitting issue and see if that increases the model's performance. Steps to do this could include tuning the hyperparameters such as the “max\_depth” and the “min\_samples\_split”. Additionally, using regularization could also help improve the accuracy and reduce overfitting. Also, determining the best way to partition the data into the low and high ridership groups could help increase the performance of the model. Taking more time to analyze each of the 11 features within the model to potentially refine some of the features within the model, could also help improve the model's performance.

## B. Business Impact

As explained when outlining the business problem, the ability for MTA to accurately predict ridership patterns is vital for efficient operation. Additionally, being able to accurately predict ridership would allow for better utilization of resources. Our model would allow for better allocation of trains, more efficient staffing, and ultimately a better customer experience by allowing MTA to predict when they will have the highest number of riders and when they will have the lowest number of riders.

According to the MTA website, the 2022 MTA Operating budget was projected to be nearly $19.4 billion (MTA). This is a significant portion of their budget, roughly 58%. Our model would allow MTA to have a more accurate idea of their operational costs and allow them to allocate their budget more effectively.

Similar research titled “Forecasting Ridership for a Metropolitan Transit Authority”, was conducted in 2011, using data from the Metropolitan Tulsa Transit Authority. The researchers used three different models to predict ridership. This research was utilized to help justify the investment in public transit. Although there is room for improvement within our low ridership and high ridership models, our models could be utilized for the same type of impact within New York City.

# VII. Deployment

## A. Implementation Strategy

Our model has valuable capabilities for MTA, and partnering with them could provide accurate ridership predictions, allowing them to increase efficiency in their operations. There are many different ways that this model could be deployed for MTA. One deployment option could be weekly or monthly forecasts, collecting the most recent data and using it to provide accurate predictions for the following week or month. Another, more advanced deployment option could be real-time predictions, creating an application where MTA could know the predicted ridership for a given hour, day, week, month, quarter, or year. These predictions could be used to understand staffing needs for that given time period. The second option may require more features implemented into our model, such as weather conditions and events happening. This is a deployment strategy that we believe would provide great value to MTA.

## B. Risk Assessment

As with all data mining solutions, there are always risks associated that should be known. We will continue to make improvements to the model to enhance the accuracy, but these models can still make mistakes, so it is crucial to leave room for potential errors in ridership predictions. Additionally, privacy protection is important. Data mining collects and utilizes a lot of data, so any organization needs to ensure they have a means of keeping the data safe and secure. For future versions of this model, including a real-time prediction model, it is crucial to ensure the organization has the bandwidth and budget to implement the model into its decision-making workflow. Overall, our model can be used as an incredibly powerful tool, and ensuring it is implemented correctly will allow the organization to get the most value out of it.

# IX. Conclusion and Recommendations

After testing different models and understanding the performance of each, a random forest regressor model is the best model to predict ridership with the MTA data. Furthermore, creating two different models, one to predict low ridership and one to predict high ridership, allows for more accurate ridership predictions. This model is a powerful tool and can be used to greatly help MTA improve its operations. There are some limitations to our model, and continuing to refine features and tune parameters will allow us to continue to improve the model and allow MTA to make better predictions. There are many different implementation strategies for this model, but we believe working with MTA to create a model for real-time predictions would be a highly valued deployment strategy.

# Appendix

Team contributions: All team members contributed to the cleaning of data and code, which allowed for the development of our models. For our written report, each team member contributed the following.

Esteban: Proposal (I)

Kendall: Business Understanding & Data Understanding (II & III)

Anthony: Data Preparation & Modeling (IV & V)

Jack: Evaluation, Deployment, & Conclusion and Recommendations (VI, VII, IX)

Figure 1: Performance statistics for TotalDatasetRandomForest.ipynb

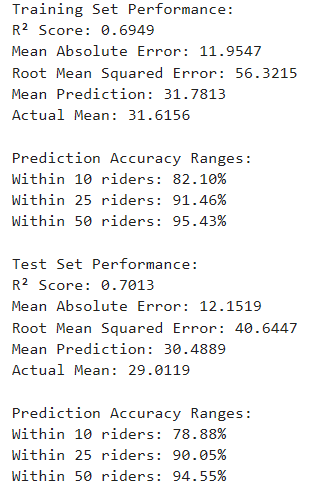


Figure 2: MAPE statistics for TotalDatasetRandomForest.ipynb

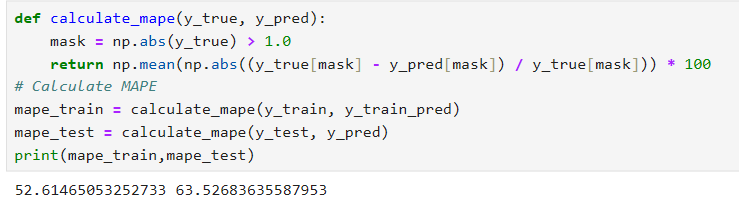


Figure 3: MAPE value range statistics for TotalDatasetRandomForest.ipynb

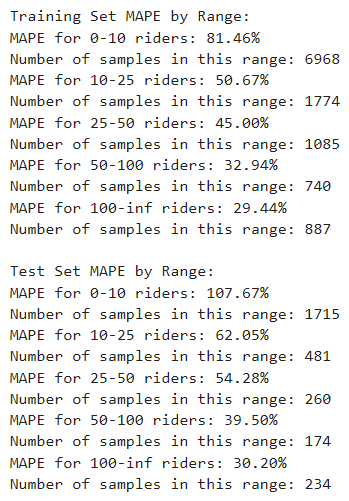


Figure 4: Residual Plot for TotalDatasetRandomForest.ipynb Training Set

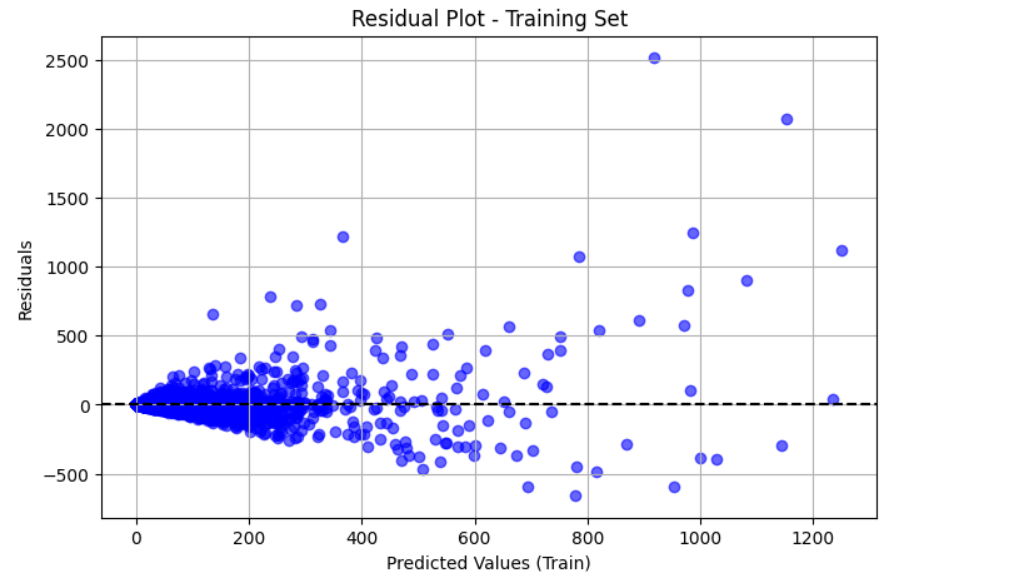


Figure 5: Residual Plot for TotalDatasetRandomForest.ipynb Test Set

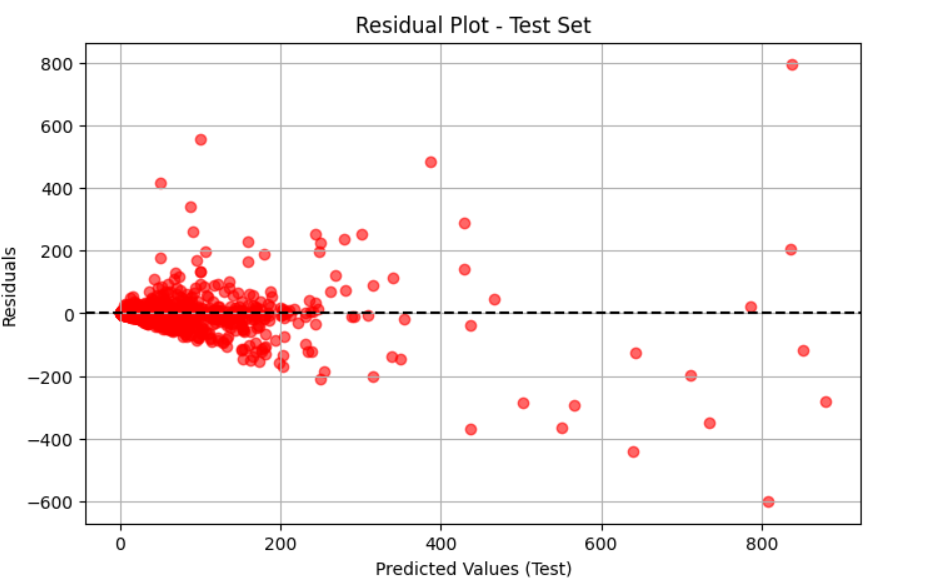


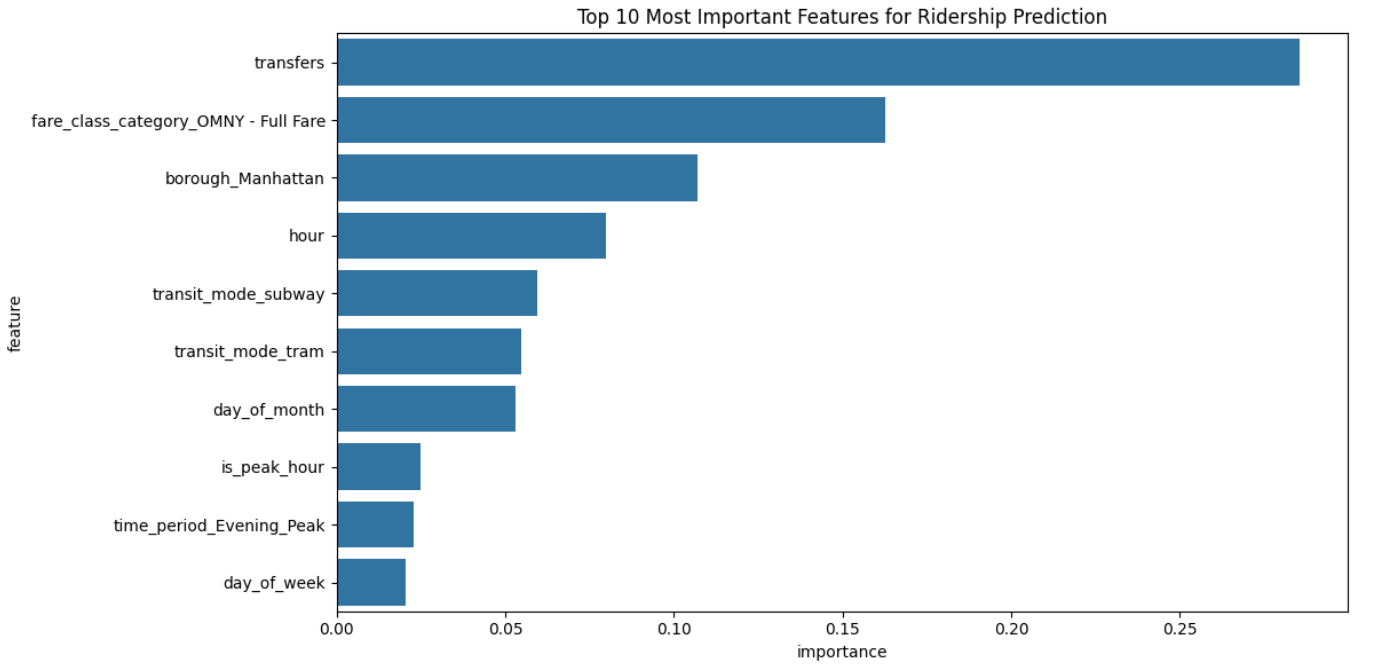
Figure 6: Important Features Plot For TotalDatasetRandomForest.ipynb

Figure 7: Performance statistics for Low Ridership Model in RidershipModel.ipynb

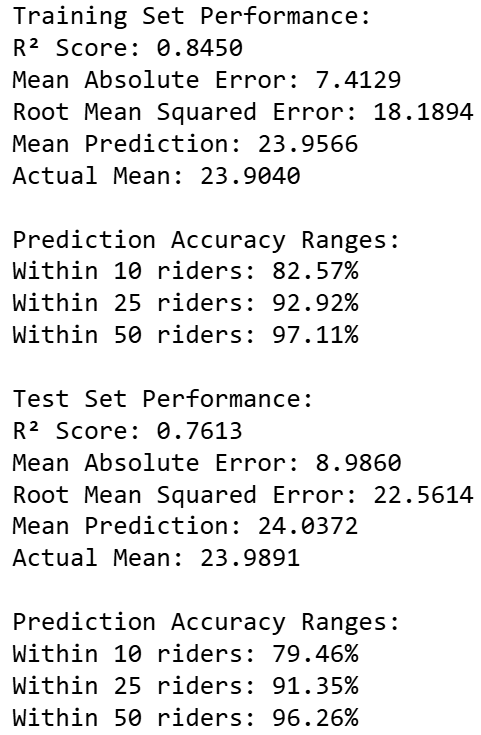


Figure 8: Performance statistics for High Ridership Model in RidershipModel.ipynb

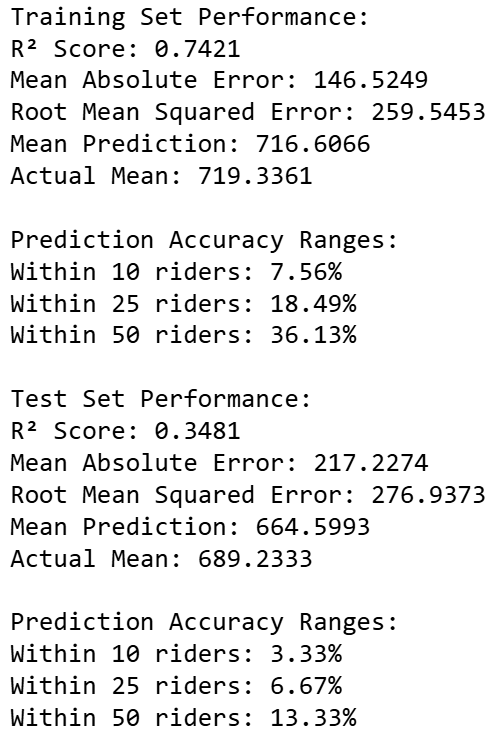
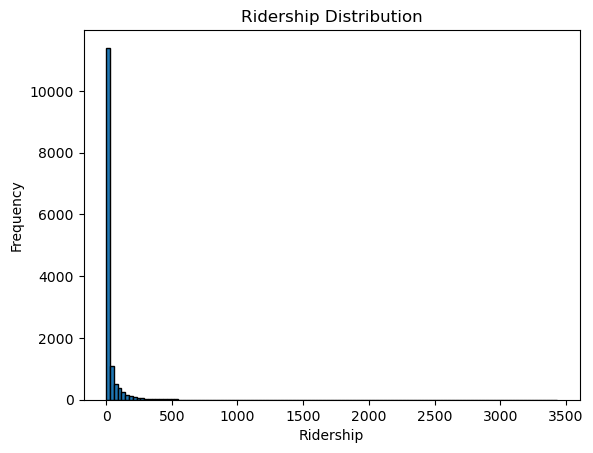


Figure 9: Distribution of Ridership



# Citations

Bass, Greg. "Subway, Commuter Rails See Overcrowding amid Congestion Pricing as

Riders Are 'Smashed up Against One Another': Straphangers." *New York Post*, 11 Jan. 2025, <https://nypost.com/2025/01/11/us-news/subway-commuter-rails-see-crowding-over-congestion-pricing-riders/>.

Brownlee, Jason. "How to Develop a Random Forest Ensemble in Python." *Machine*

*Learning Mastery*, 27 Apr. 2021, https://machinelearningmastery.com/random-forest-ensemble-in-python/.

Carson, David. “Understanding the Risks Associated with Data Mining.”

*RiskDataControl.Com*, RiskDataControl.com, 24 Mar. 2025, www.riskdatacontrol.com/understanding-risks-associated-data-mining/.

Chiang, Wen-Chyuan, et al. “Forecasting Ridership for a Metropolitan Transit Authority.”

Transportation Research Part A: Policy and Practice, vol. 45, no. 7, Aug. 2011, pp. 696–705, https://doi.org/10.1016/j.tra.2011.04.016.

Metropolitan Transportation Authority. "MTA Operating Budget Basics." *MTA.info*,

Metropolitan Transportation Authority, https://www.mta.info/budget/MTA-operating-budget-basics. Accessed 5 May 2025.

U.S. General Services Administration. *MTA Subway Hourly Ridership: Beginning 2025*.

Data.gov, 3 May 2025,<https://catalog.data.gov/dataset/mta-subway-hourly-ridership-beginning-2025>.