Introduction

For this project, we are focusing on linear regression and decision trees with some form of pruning. The data set that I'll be using is titled, "MTA Subway Hourly Ridership: Beginning February 2022" (Metropolitan Transportation Authority) from New York State. New York State makes their data openly, and freely, accessible for public use. This data features 11,802,622 rows and 12 columns. The twelve columns are 'transit_timestamp', 'station_complex_id', 'station_complex', 'borough', 'routes, 'payment_method', 'ridership', 'transfers', 'latitude', 'longitude', 'Georeference' and 'itsuid'.

The MTA Subway Hourly Ridership data came with a dictionary that provides context on each of these columns, which you'll find recreated below (Metropolitan Transportation Authority).

Data Label	Data Type	Data Description
transit_timestamp	DATE	Timestamp payment took place in local time. All transactions here are rounded down to the nearest hour. For example, a swipe that took place at 1:37pm will be reported as having taken place at 1pm.
station_complex_id	ALPHANUMERIC	A unique identifier for station complexes
station_complex	TEXT	The subway complex where an entry swipe or tap took place. Large subway complexes, such as Times Square and Fulton Center, may contain multiple subway lines.
borough	TEXT	Represents one of the boroughs of New York City

		serviced by the subway system (Bronx, Brooklyn, Manhattan, Queens).
routes	TEXT	Refers to the different subway routes that stop at a particular subway station.
payment_method	TEXT	Specifies whether the payment method used to enter was from OMNY or MetroCard.
ridership	NUMERIC	Total number of riders that entered a subway complex via OMNY or MetroCard at the specific hour.
transfers	NUMERIC	Number of individuals who entered a subway complex via a free bus-to-subway, or free out-of-network transfer. This represents a subset of total ridership, meaning that these transfers are already included in the preceding ridership column. Transfers that take place within a subway complex (e.g., individuals transferring from the 2 to the 4 train within Atlantic Avenue) are not captured here.
latitude	DECIMAL	Latitude for specified subway complex
longitude	DECIMAL	Longitude for the specified subway complex

I chose this dataset because it featured multiple types of data that I thought would work well for linear regression. Additionally, this dataset was free and publicly available with no restrictions on the usage of the data. The quality of this dataset was also taken into consideration for this project. This dataset was rather clean, with only two null values, which is untypical for a set of this size. The size of the dataset is much larger than previous sets I've

worked with but I thought that it would be a good challenge and help with the complexity of the project.

For this project, I used a multiple libraries within Python, the first of which is pandas. Pandas is an open source library used for data manipulation, reading and writing data and more. It's main use in this project was to read in the data and perform basic data exploration, including finding the shape of the data frame, searching for null values, looking at the data types, looking at the unique values and finding summary statistics. The second library that was used is scikit- learn which is a machine learning library that helps with predictive analysis. Additionally, it is used for data preprocessing, classification, regression, model selection and more. In this project it was used for model selection, pre-processing, a linear model and metrics when performing linear regression. When making our decision trees, scikit-learn was used to create the tree and metrics. It was not used for model selectin or pre-processing in this case because that had been done previously for linear regression and is not something that was necessary to repeat. To create some of the charts within the project, Matplotlib was used which is a Python library used for creating interactive visualizations.

Linear Regression

The Linear Regression problem that I was focusing on was ridership. To focus on this, the features for the model were 'borough', 'routes', 'payment_method', 'transfers', 'latitude', 'longitude' with the target variable being 'ridership'. Once those variables were established, 'borough', 'routes' and 'payment_method' were encoded since they're all categorical data, not numerical. Once they're encoded we are able to split our data into training and testing sets. After predicting on the test set and creating prediction variables, we were able to calculate the training and testing rate. The rates are as follows:

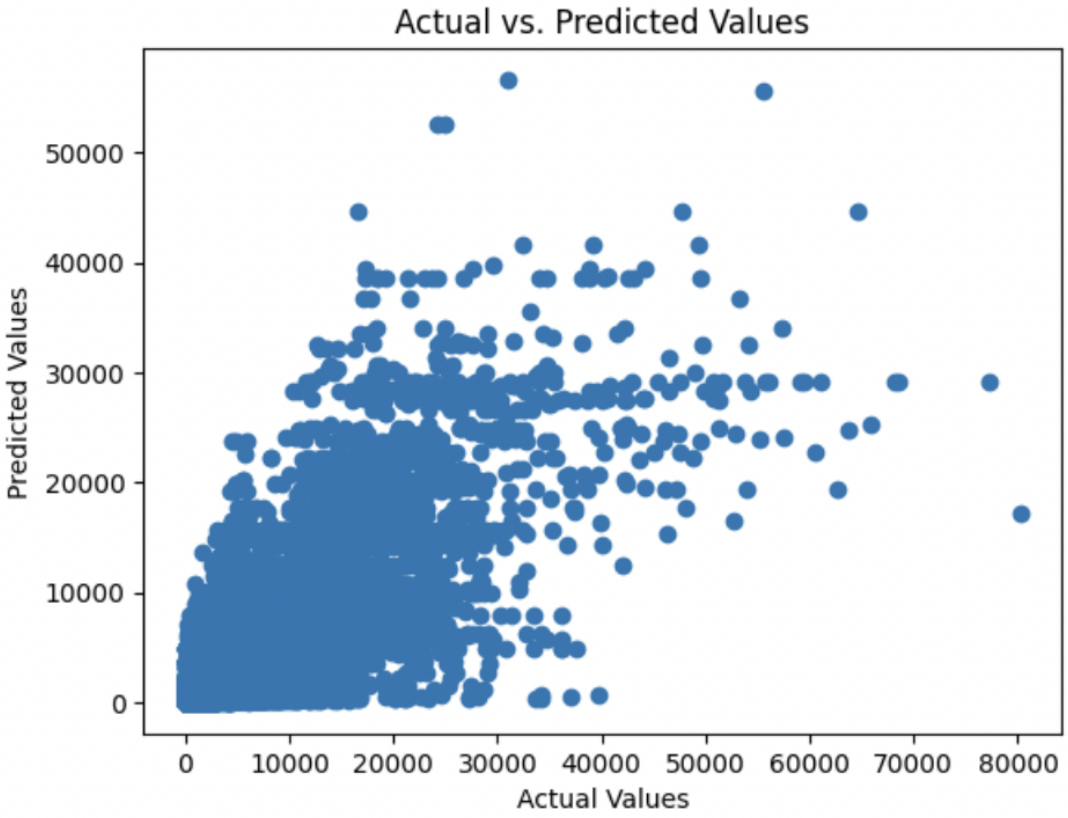
MSE train: 781.587, test: 777.423

RMSE train: 252.601, test: 252.185

Linear Model training R^2 score: 0.333

Linear Model testing R^2 score: 0.335

The actual and predicted values can be seen below. There is a large cluster of data in the bottom-left corner of the chart with a few outliers seen throughout. This allows us to assume our model is making reasonable predictions.



Decision Trees

Next, we evaluated our model with decision trees. The features and target variables remind the same as before for linear regression. Additionally, our testing and training sets remain the same as before. The first decision tree had a max depth of 5, which will force some pruning. The second decision tree had a max depth of 10. For the tree with the max depth of 5, the training and testing rates were as follows:

Mean Squared Error (MSE) testing: 425697.49269635003

Mean Squared Error (MSE) training: 433875.63065112435

R-squared (R2) testing: 0.5317188599590656

R-squared (R2) training: 0.5263860624477059

For the tree with the max depth of 10, the training and testing rates were as follows:

Mean Squared Error (MSE) testing: 288693.49443903076

Mean Squared Error (MSE) training: 433875.63065112435

R-squared (R2) testing: 0.682427730917501

R-squared (R2) training: 0.5263860624477059

Analyses of results

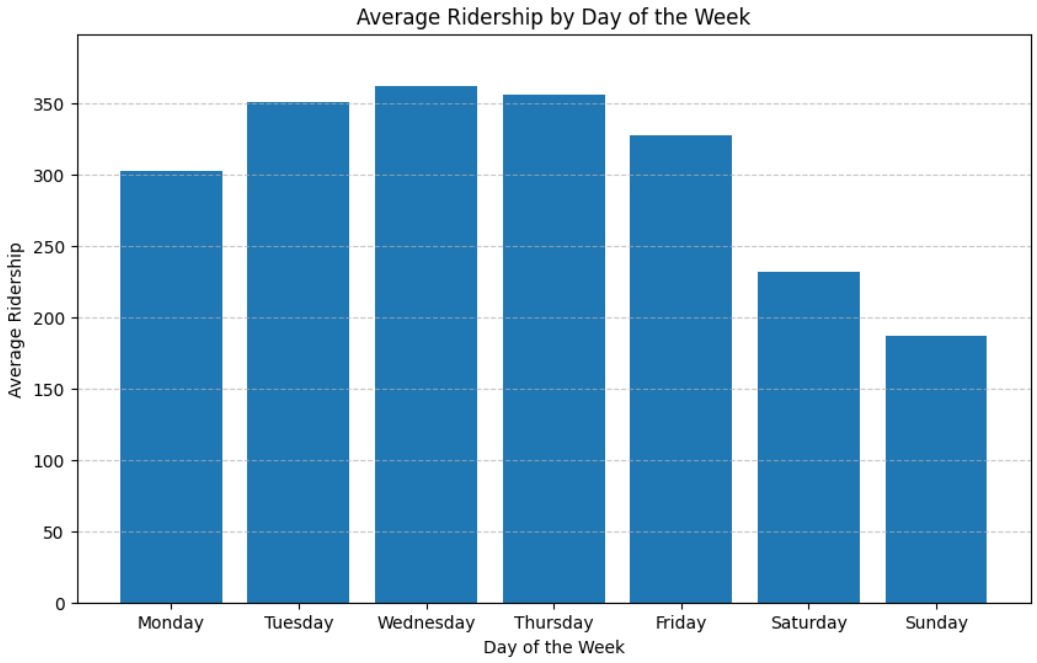
The results for Linear Regression were very close. The MSE train was 781.587 and test: 777.423. This number being so close for the training and testing could mean that our model is performing well. The same thing is seen when looking at RMSE train: 252.601, test: 252.185. The close numbers means once again that our model is performing well or that there is not enough data variability, however, this may not be the case because of the size of our dataset. Finally, the Linear Model training R^2 score: 0.333 and the Linear Model testing R^2 score: 0.335. These low score in both cases indicate that the model is not fitting the training or testing data well.

For the decision tree with a max depth of five, we did not see the same results as with linear regression. In this case the Mean Squared Error (MSE) testing is 425697.49269635003 and the Mean Squared Error (MSE) training is 433875.63065112435. These numbers show that the that our model is underfitting since there is such a large difference between the testing and training score. When looking at the R-squared (R2) testing score of 0.5317188599590656 and R-squared (R2) training score of 0.5263860624477059, we can conclude that the model is balanced since the scores are so similar. For the decision tree with a max depth of ten, we saw a more drastic separation of the scores. The Mean Squared Error (MSE) testing was

288693.49443903076 and the Mean Squared Error (MSE) training was 433875.63065112435. This signifies that the model could be overfitted because the testing MSE is so much lower than the training MSE. When looking at the R-squared (R2) testing of 0.682427730917501 and R-squared (R2) training of 0.5263860624477059, which tells us that the model is not balanced.

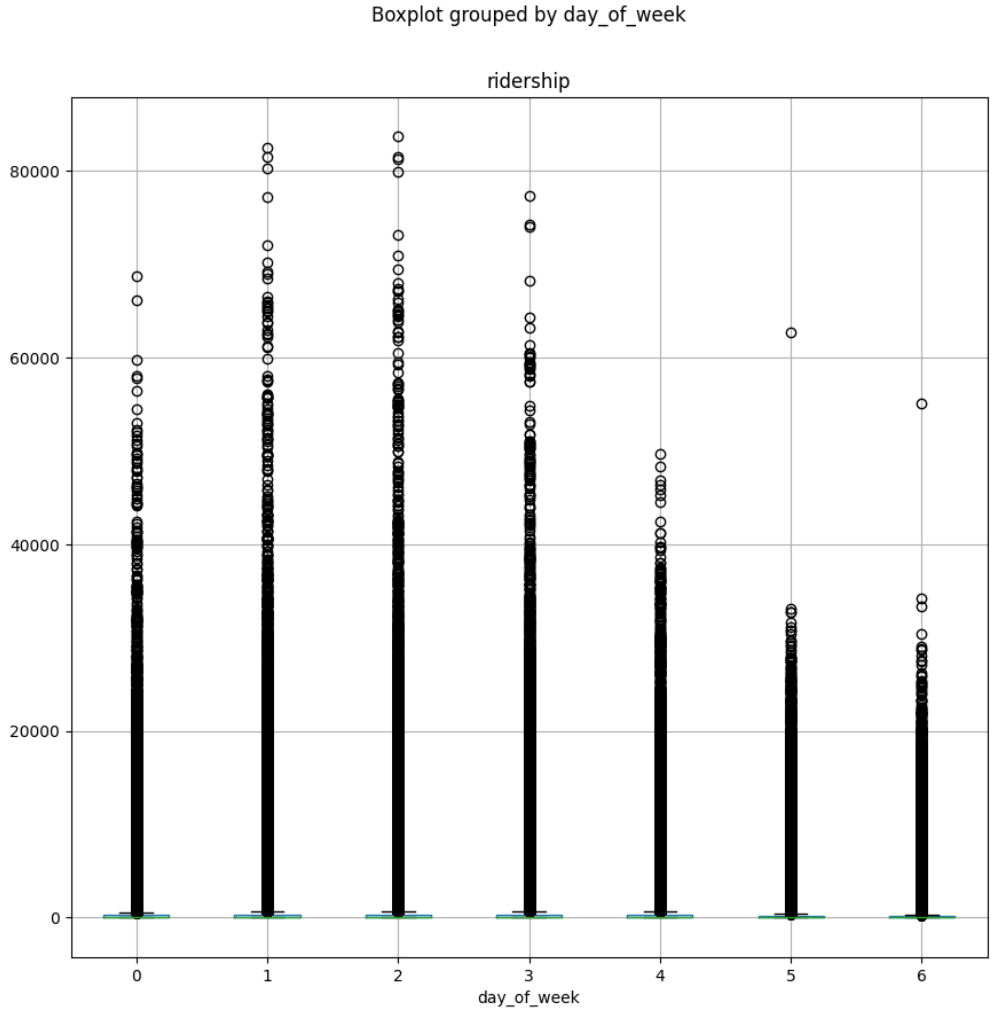
To improve performance for decision trees, we could limit the max depth which will simplify the complexity of our tree and could improve the R-squared scores. Since the scores were better when the max depth was equal to five, we could limit the max depth to 7 (between our two attempts of five and ten) to see if the scores improve. One approach would be to test each depth separately and look at the scores to find the best scores. To improve the linear regression scores, the outliers in the data could be removed. It could also be improved by including more data and different feature selection.

With this dataset, we were able to answer many questions, such as the average ridership by day of the week. By looking at this chart, it is seen that there are more riders on the



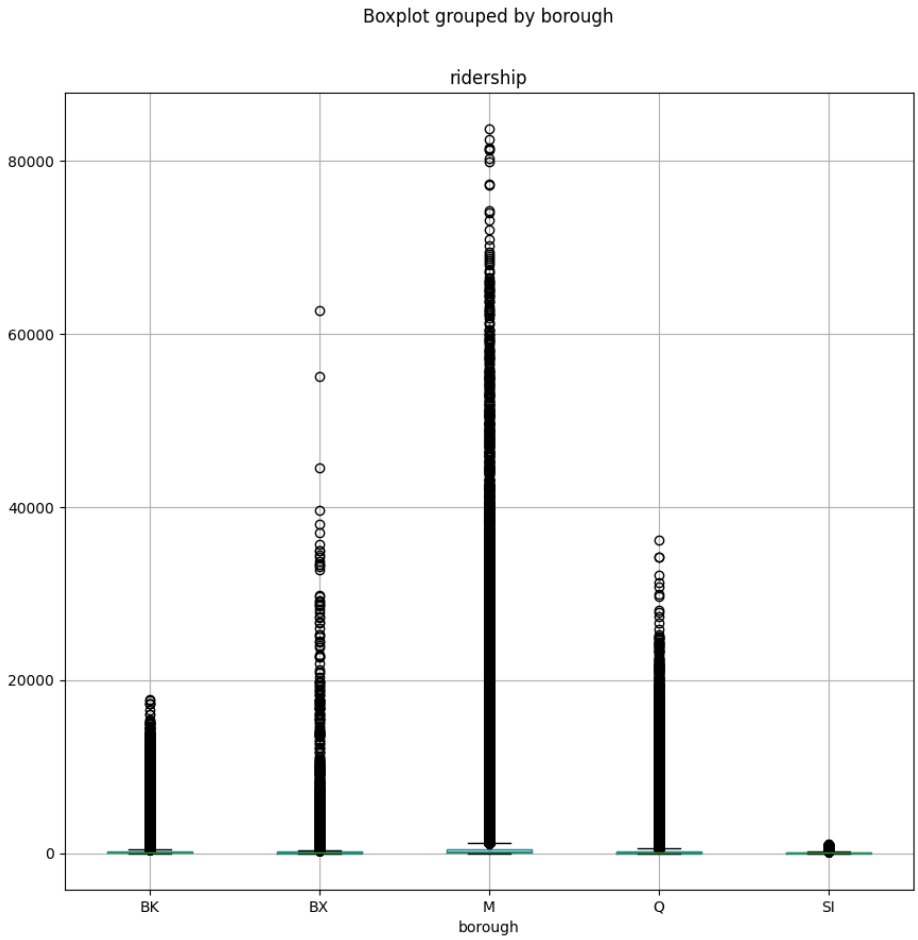
MTA on Wednesdays and ridership drops off significantly during the weekend. We can make the assumption from this chart that more people take the subway during the week do to school

and work. When looking at all ridership, instead of the average, we see the same results. Ridership declines on the weekends and is most used on Tuesday, Wednesday and Thursday.

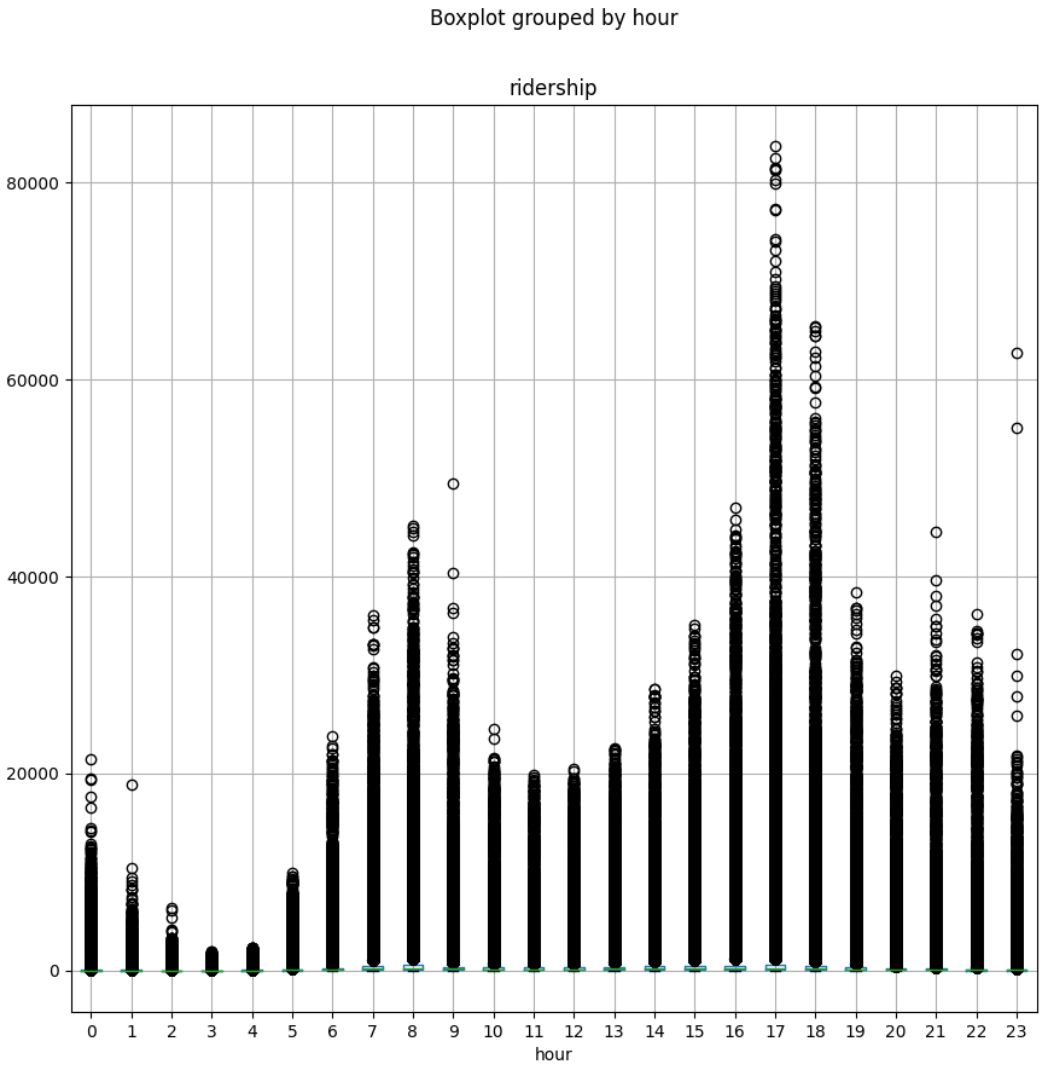


The next thing explored in the dataset was the ridership by borough. Manhattan had a significantly higher ridership than the other boroughs. Bronx followed behind Manhattan, then Queens, Brooklyn and Staten Island. There are many factors that could influence the ridership that could be explored in the future such as how does the population of each borough reflect in the ridership, does the economic status of each borough show through the ridership, for example, are more people walking in boroughs that are not as well off as Manhattan, and how

many subway stations are in each borough - as well as the amount of days the station was opened as opposed to being closed, or what routes in which broughs had significant delays.



After looking at the boroughs data, we explored the ridership data by hour. We saw that the ridership peaked between 8 and 9 am as well as 5 and 6 pm. Since these are the standard commuting times for many people, it was not surprising to see. In the future, ridership by hour by borough could be explored as well as ridership by hour by day could also be explored.



The performance of the dataset chosen worked rather well and as seen in the charts allowed us to draw conclusions from the data. While the data worked well for basic exploration, it was not the best fit for linear regression and decision trees. The algorithm that performed best in terms of best fit, but not most accurate would be the linear regression model. However, the most accurate model with a slight overfit was the decision tree model with a max depth of five. As mentioned previously, there are ways for this to be improved in future projects.

References

Analytics Vidhya. “8 Proven Ways for Boosting the “Accuracy” of a Machine Learning Model.” *Analytics Vidhya*, 28 Dec. 2015, www.analyticsvidhya.com/blog/2015/12/improve-machine-learning-results/.

Matplotlib. “Matplotlib: Python Plotting — Matplotlib 3.1.1 Documentation.” *Matplotlib.org*, 2012, matplotlib.org/.

Metropolitan Transportation Authority. “MTA Subway Hourly Ridership: Beginning February 2022: State of New York.” *MTA Subway Hourly Ridership: Beginning February 2022 | State of New York*, 11 Oct. 2023, data.ny.gov/Transportation/MTA-Subway-Hourly-Ridership-Beginning-February-202/wujg-7c2s.

Pandas. “Python Data Analysis Library — Pandas: Python Data Analysis Library.” *Pydata.org*, 2018, pandas.pydata.org/.

“Scikit-Learn: Machine Learning in Python.” *Scikit-Learn.org*, 2019, scikit-learn.org/stable/. Accessed 11 Oct. 2023.