MTA Ridership Analysis

By:
Brandon Hernandez
Lam Nguyen
Sonal Parmar
Murad Khan

Introduction

- What are we analyzing when we say “MTA Ridership”?
  - The Metropolitan Transportation Authority (MTA) subway system, a crucial component of New York City's urban transportation, faces the ongoing challenges of maintaining operational efficiency, accommodating increasing demand, and ensuring affordability for its diverse ridership
- Why are we analyzing “MTA Ridership”?
  - The goal of this project is to analyze how fare increases affect hourly ridership in the MTA subway system, aiming to offer actionable insights for policy-making.
  - This involves analyzing historical data to detect trends for which stations, time of day and day of the week. This information can be used to schedule maintenance and improve crowd management.
    - More than 22.5 Mil records

What Does the MTA Ridership Data Consist of?

Column Name	Data Type	Purpose
transit_timestamp	Plain Text	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	Plain Text	A unique identifier for station complexes
station_complex	Plain 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.
routes	Plain Text	Refers to the different subway routes that stop at a particular subway station.
borough	Plain Text	Represents one of the boroughs of New York City serviced by the subway system (Bronx, Brooklyn, Manhattan, Queens).
payment_method	Plain Text	Specifies whether the payment method used to enter was from OMNY or MetroCard. The value all is temporarily being used while this data is being made available.
ridership	int64	Total number of riders that entered a subway complex via OMNY or MetroCard at the specific hour.
transfers	int64	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	float64	Latitude for specified subway complex
longitude	float64	Longitude for the specified subway complex
Georeference	Point	Open Data platform-generated geocoding information from supplied address components. Point-type location is the centroid of the address components provided and does not reflect a specific address if the street address component is not provided. Point-type location is supplied in "POINT ( )" format.

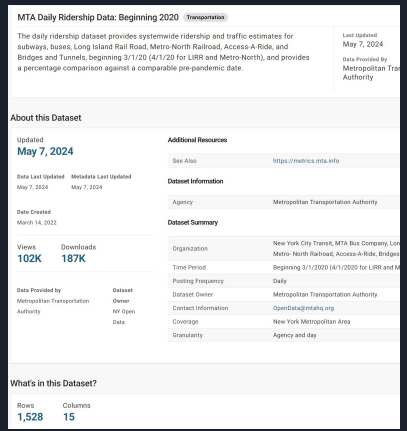
Data Collection

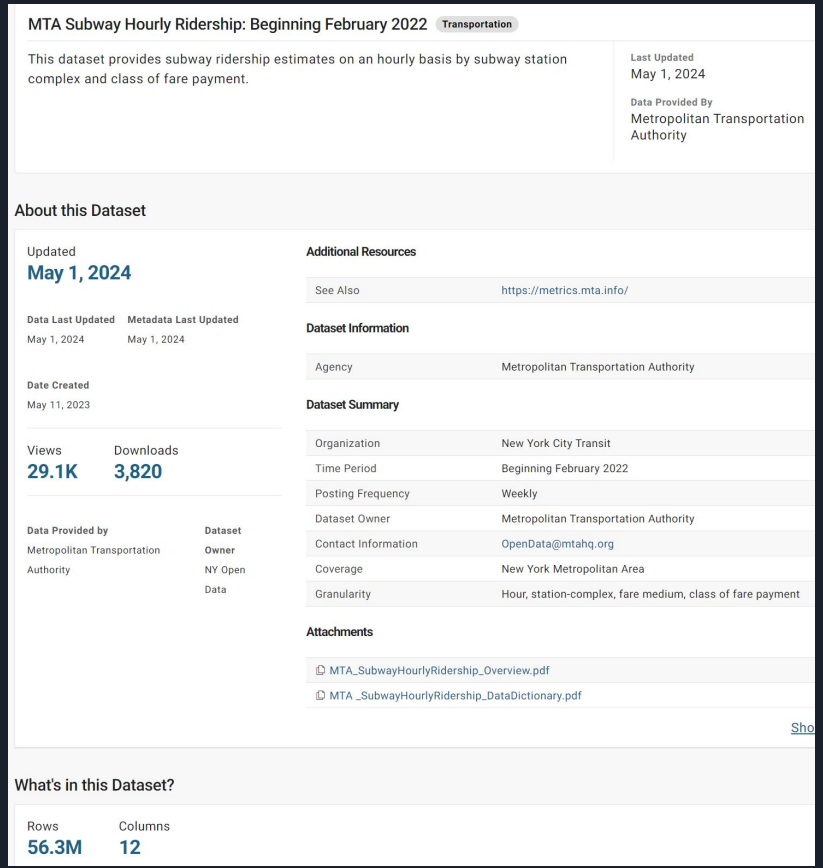
Data.gov data sets

- Hourly ridership [data.gov]
  - Kaggle (Burak Akay)
- Daily ridership [data.gov] (For Overview Part)

The hourly ridership file was too big to download under normal means and had to be chopped into smaller portions to unify the datasets together ourselves.

The data we collected ranges from Feb. 2022 - Feb. 2024





Tools

pandas
numpy
seaborn
matplotlib
Sklearn :

- Random Forest Regressor
- Gradient Boosting Regressor

Tableau
Power BI

2. Data Cleaning

```
[ ] # change transit_timestamp to datetime object
    df['transit_timestamp'] = pd.to_datetime(df['transit_timestamp'])

[ ] #Make a copy before slicing the dataset
    # original_df = df.copy()

    # check the earliest and latest time of the data
    print(df['transit_timestamp'].min())
    print(df['transit_timestamp'].max())

    2022-02-01 00:00:00
```

Data Cleaning

converts the 'transit_timestamp' column in the DataFrame 'df' to datetime objects, facilitating easier manipulation and analysis of date and time data.

```
# prompt: check which column contain NaN number

# Check for NaN values in each column
df.isnull().sum()
# Drop columns with NaN values
df.dropna(axis=1, inplace=True)

[ ] # Check is there is any NaN values
    df.isnull().sum()
```

```
[ ] # prompt: get unique values of column borough. update value M into Manhattan, Q to Queens, BK to Brooklyn, BX t

    df['borough'].replace({'M': 'Manhattan', 'Q': 'Queens', 'BK': 'Brooklyn', 'BX': 'Bronx', 'SI': 'Staten Island'}

[ ] # prompt: get unique values of column borough.
    unique_boroughs = df['borough'].unique()
    print(unique_boroughs)

    ['Manhattan' 'Queens' 'Brooklyn' 'Bronx' 'Staten Island']
```

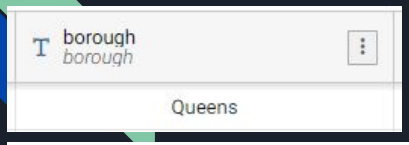
Clean the names of the stations by removing the route numbers

```
[ ] # remove the () and everything in between the () from each station_complex row in the df
    df['station_complex'] = df['station_complex'].str.replace(r'\(.*\)', '', regex=True)
```

Data Cleaning

Kaggle >

< Data.gov (original)



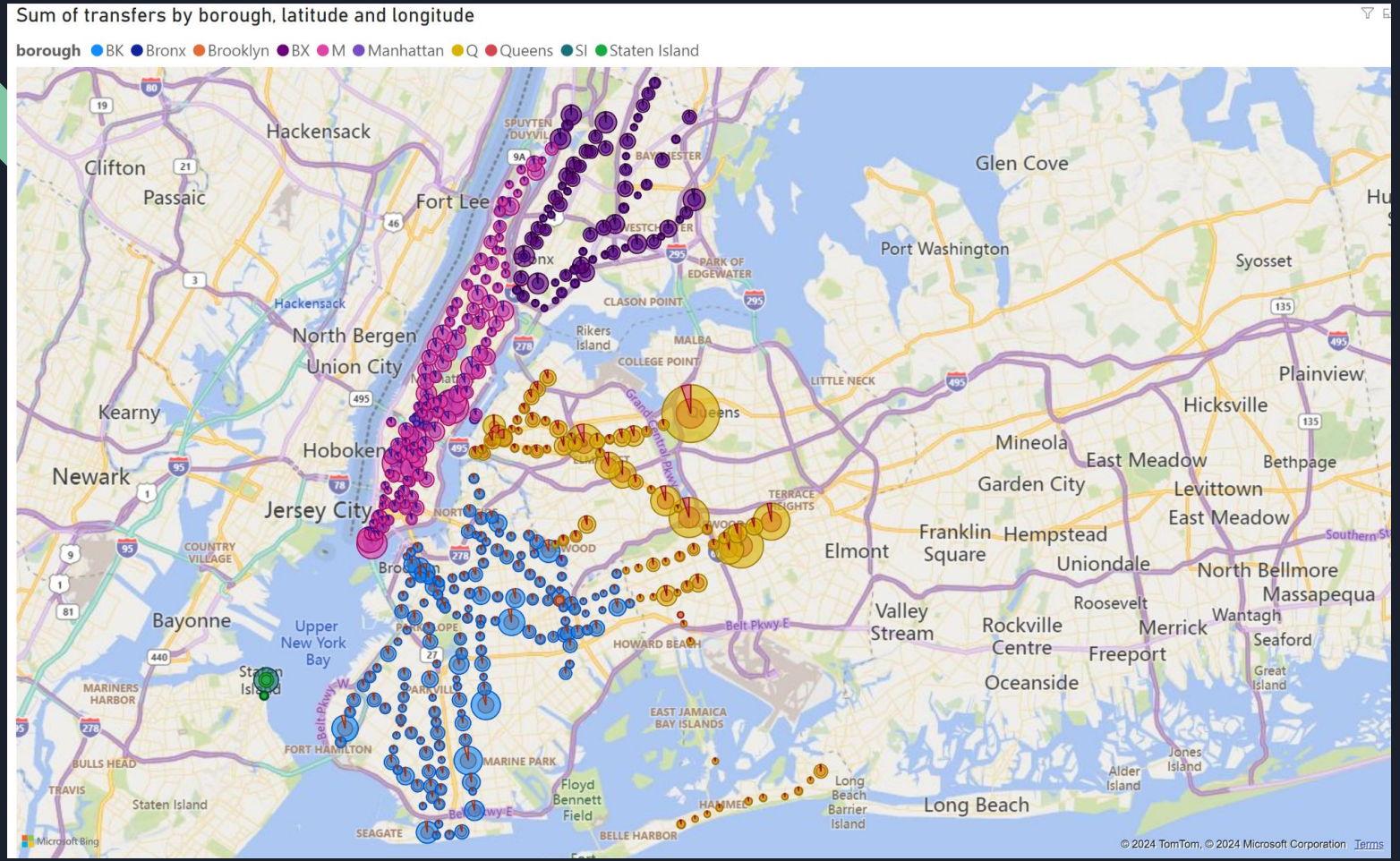
transit_timestamp	station_complex_id	station_complex	borough
2022-01-31 2023-09-28	425 unique values	425 unique values	BK 37% M 29% Other (4047199) 35%
06/24/2022 04:00:00 AM	H007	1 Av (L)	M

	transit_timestamp	station_complex_id	station_complex	borough	payment_method	ridership	transfers	latitude	longitude	Georeference
0	2022-06-24 04:00:00	H007	1 Av (L)	M	omny	19	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)
1	2022-07-05 22:00:00	H007	1 Av (L)	M	omny	229	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)
2	2022-08-30 03:00:00	H007	1 Av (L)	M	omny	5	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)
3	2022-11-05 03:00:00	H007	1 Av (L)	M	omny	78	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)
4	2023-01-19 04:00:00	H007	1 Av (L)	M	omny	5	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)

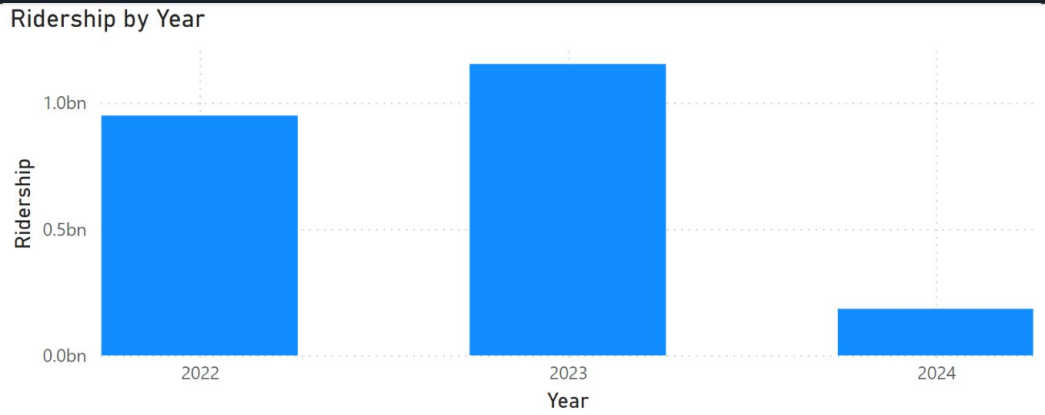
	transit_timestamp	station_complex_id	station_complex	borough	payment_method	ridership	transfers	latitude	longitude	Georeference
0	2022-06-24 04:00:00	H007	1 Av (L)	Manhattan	omny	19	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)
1	2022-07-05 22:00:00	H007	1 Av (L)	Manhattan	omny	229	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)
2	2022-08-30 03:00:00	H007	1 Av (L)	Manhattan	omny	5	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)
3	2022-11-05 03:00:00	H007	1 Av (L)	Manhattan	omny	78	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)
4	2023-01-19 04:00:00	H007	1 Av (L)	Manhattan	omny	5	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)

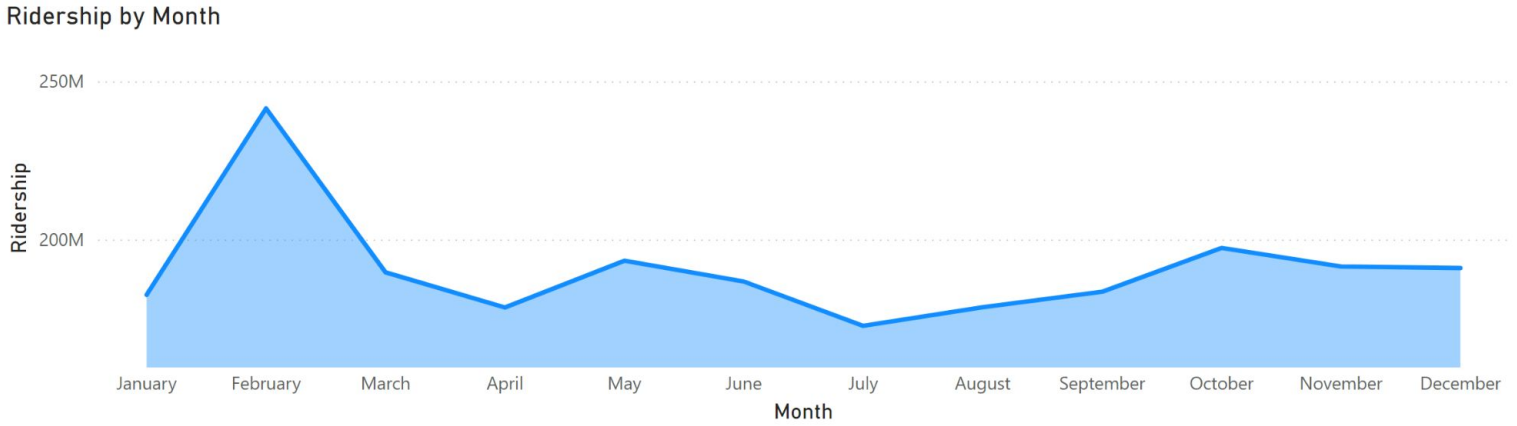
	transit_timestamp	station_complex_id	station_complex	borough	payment_method	ridership	transfers	latitude	longitude	Georeference
0	2022-06-24 04:00:00	H007	1 Av	Manhattan	omny	19	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)
1	2022-07-05 22:00:00	H007	1 Av	Manhattan	omny	229	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)
2	2022-08-30 03:00:00	H007	1 Av	Manhattan	omny	5	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)
3	2022-11-05 03:00:00	H007	1 Av	Manhattan	omny	78	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)
4	2023-01-19 04:00:00	H007	1 Av	Manhattan	omny	5	0	40.730953	-73.981628	POINT (-73.98162841796875 40.730953216552734)

Data Visualization



Data Visualization

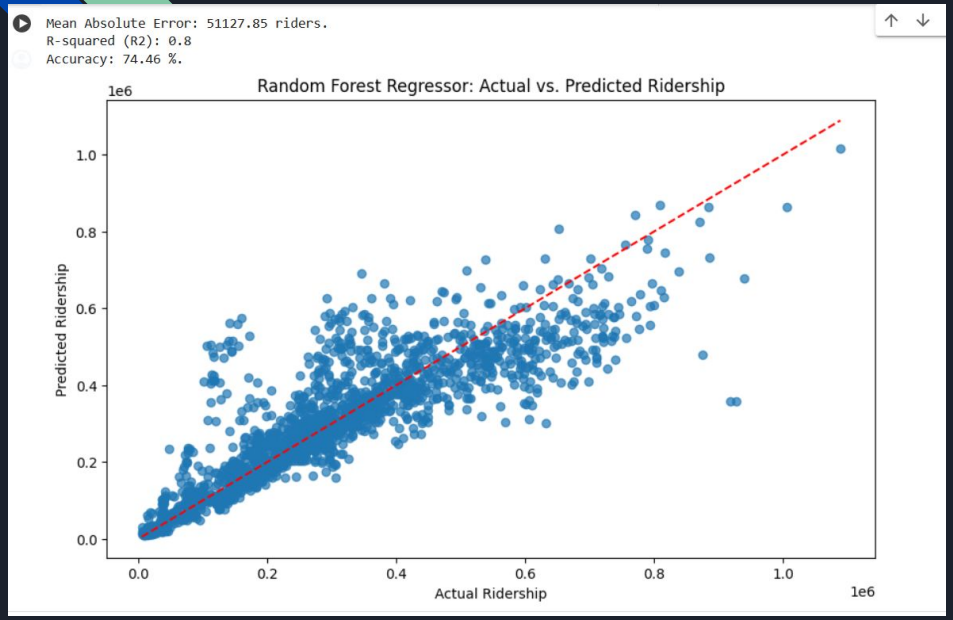


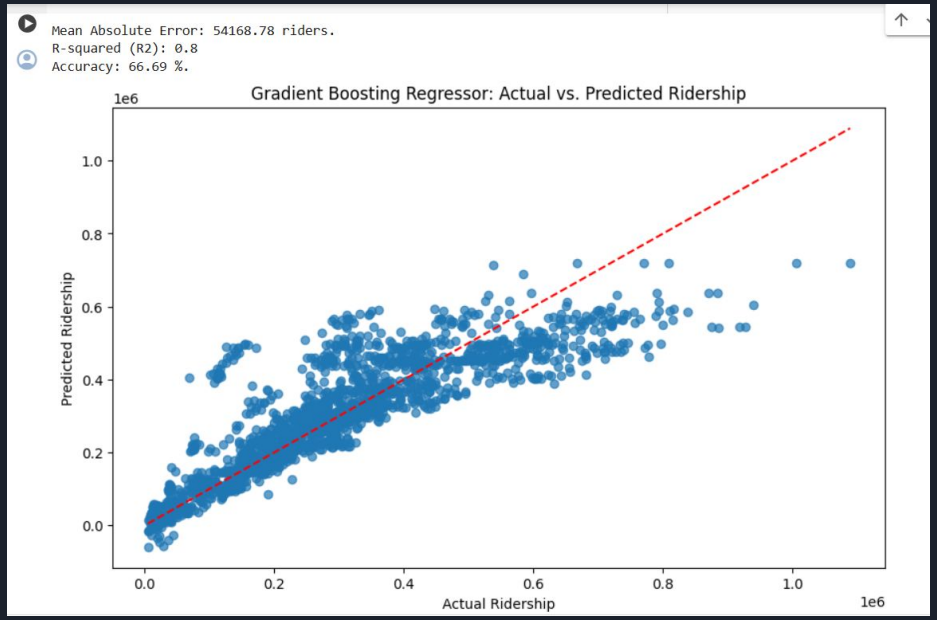


Data Visualization (cont.)

Tableau Data Overview

Random Forest Regressor and Gradient Boosting Regressor



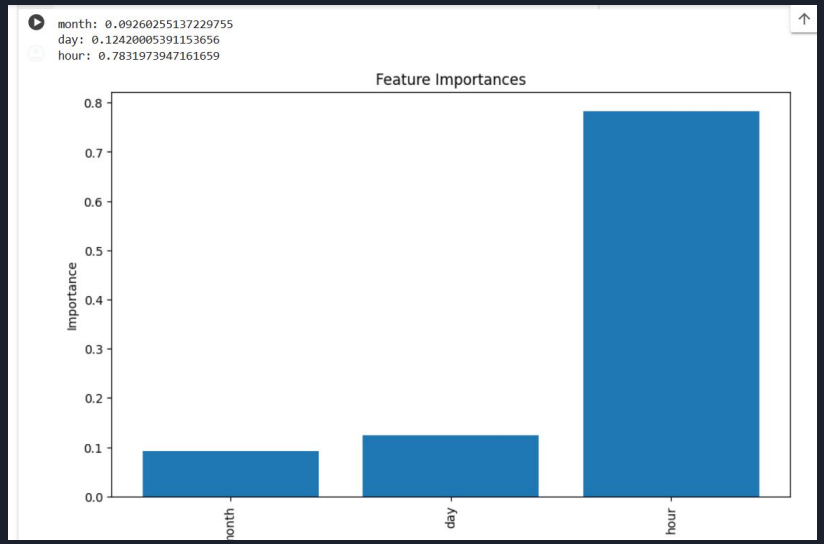


Mean Absolute Error: 51127.85 riders.
R-squared (R2): 0.8
Accuracy: 74.46 %.

Mean Absolute Error: 54168.78 riders.
R-squared (R2): 0.8
Accuracy: 66.69 %.

What are the most important features in our model?

To do this we look at the correlation in the data considering the month, day, and the hour and see their effect on the ridership numbers. **The hour of day is the major indicator on the subways ridership numbers.**




```
[ ]  # Test the function with a specific input
     predict_and_show_ridership(month=5, day=5, hour=12)

     Input: Month=5, Day=5, Hour=12
     Predicted Ridership: 292739.44
     Actual Ridership: 299949

[ ]  predict_and_show_ridership(month=3, day=27, hour=2)

     Input: Month=3, Day=27, Hour=2
     Predicted Ridership: 16202.57
     Actual Ridership: 15305
```

Discussion and Analysis

Schedule maintenance and improve crowd management

This analysis shows we can easily determine the trends for which stations, time of day and day of the week have the most riders. This information can be used to schedule maintenance and improve crowd management. We did this through bar charts, pie charts, and actual New York Map

Ridership by location

It is also clear that by using geovisualization and exploratory data analysis techniques, we can see which boroughs and stations have the most riders.

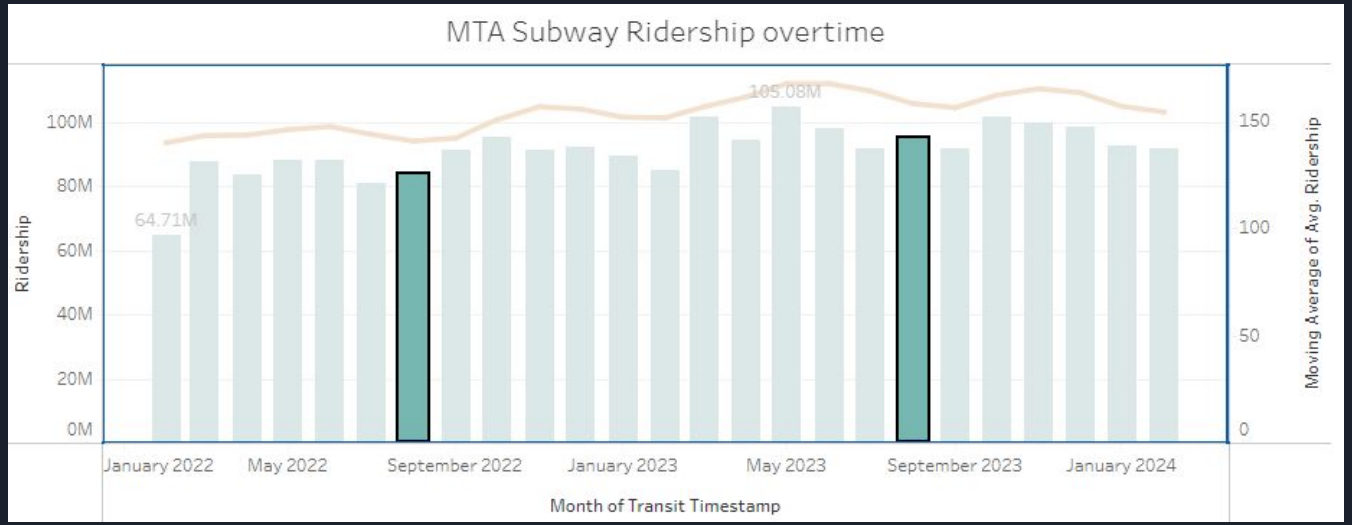
Predicting Ridership

We were able to predict ridership with a high degree of accuracy. This shows that there is potential to build a model to predict ridership. We believe that given the right data, this model could be improved and reliably predict how many riders will be at a given station at a given time.

Conclusion

What were the results of the fare increase?

- The fare increase in August 2023, had little impact on the ridership numbers in the long term, but in the short term (following months) the subway had a decline in ridership. This is emphasized more so by the figure below.



The figure above Highlights in specific August 2022 and August 2023 to further analyze the impact of the subway fare increase

Future Work

This sort of prediction model allows future works to be able to update the dataset and see of any future possible ridership numbers given previous data. This can potentially be included to also take into account the brief drop that occurs directly after any future fare increase if there is enough focus on it.

Overall this provides further abilities to predict how to prepare for future commuters on any given day and allow extra data to work with for railroad maintenance in the department of transportation.

References

Authority, Metropolitan Transportation. “MTA Daily Ridership Data: Beginning 2020: State of New York.” MTA Daily Ridership Data: Beginning 2020 | State of New York, 6 May 2024, data.ny.gov/Transportation/MTA-Daily-Ridership-Data-Beginning-2020/vxuj-8kew/about_data

Authority, Metropolitan Transportation. “MTA Subway Hourly Ridership: Beginning February 2022: State of New York.” MTA Subway Hourly Ridership: Beginning February 2022 | State of New York, 1 May 2024, data.ny.gov/Transportation/MTA-Subway-Hourly-Ridership-Beginning-February-202/wujg-7c2s/about_data.