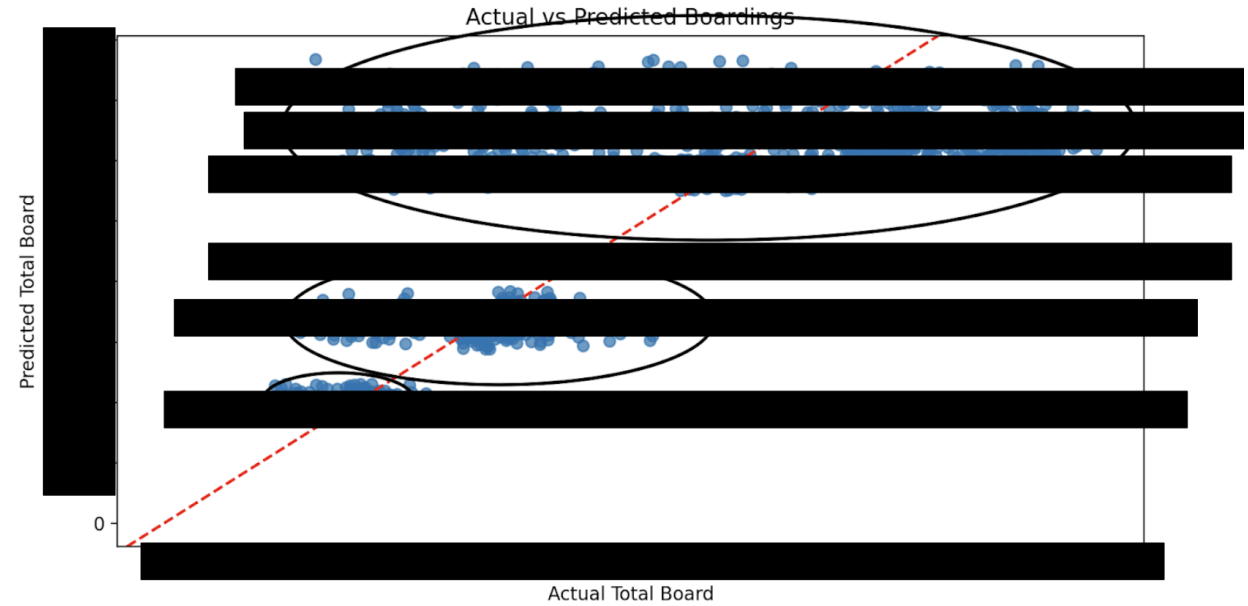
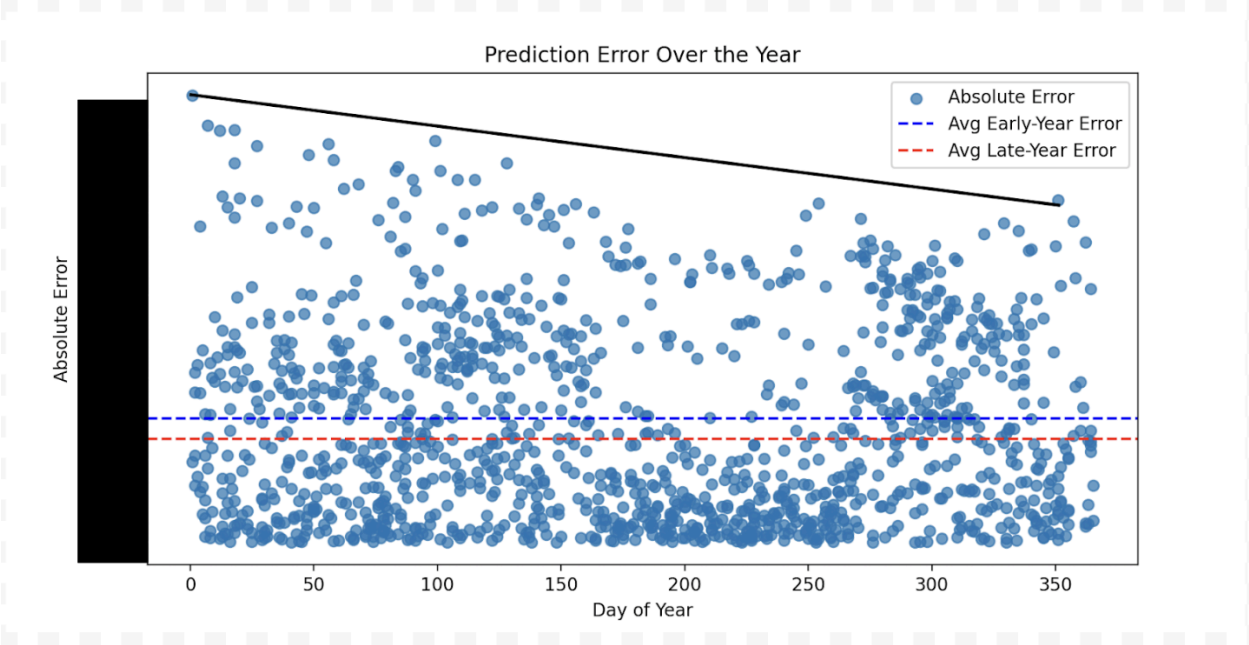
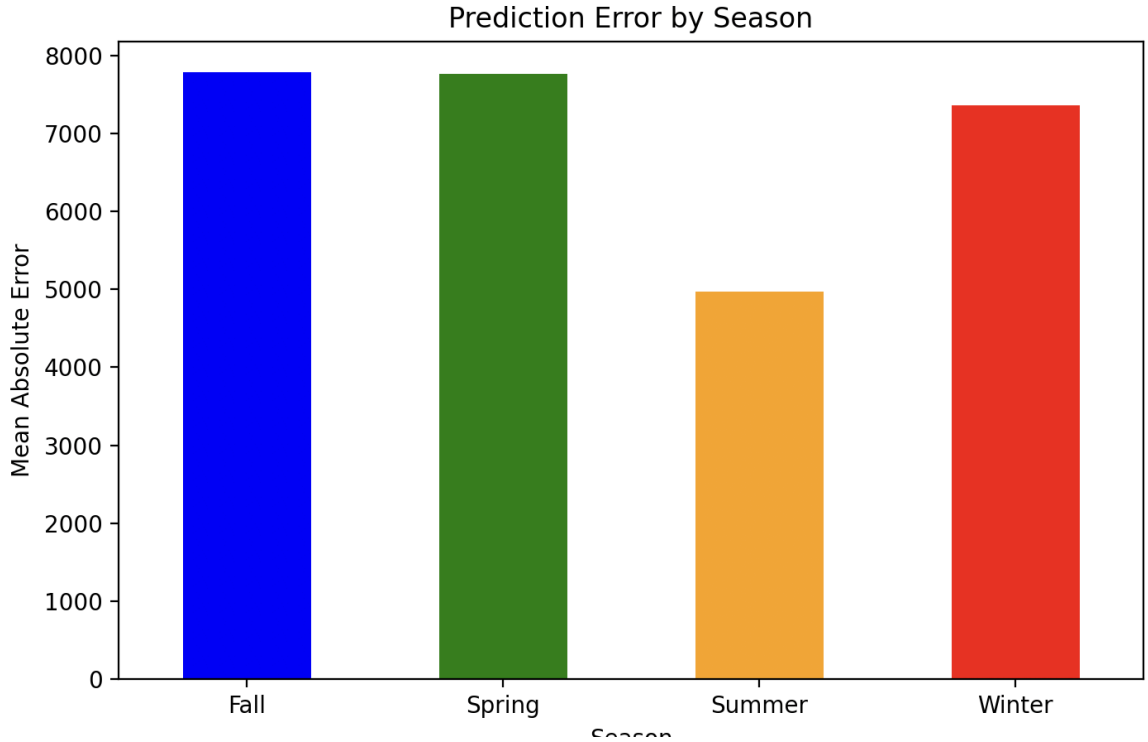
Initial Data Analysis using a ML model



First, I tested how good a NN model would be at predicting ridership based on other variables. As a sum up, it was pretty bad. So that was the first thing I was set to fix, the discrepancy of predicted vs actual and why the gaps were appearing.



Turns out the day of the year had only a little bit of a significant effect overall on error of the model, but for the error extremes, it trended a lot higher at the beginning of the year than the end of the year. You'll also notice that there is a clear gap between x=160 and 280.



If we break this down furthermore into seasons, it seems like summer is the most accurate, while the others are a lot less accurate. This tells us either that summer is a lot more predictable or there is more/less data for summer to train on than the other seasons.

Overall, this model is still very bad.

Data Prep & Analysis Strategy

Since we still have some work to do with null values and data cleaning, I want to find out what's important and what's not. You can do this a few ways, but first I'm going to try a manual analysis of the categories and their percent null. Below are all the columns with >1% null. We can drop these columns or we can conduct an analysis with just the non-null columns. We could try to fill the non-null data with estimates, but none of these columns seem applicable. I might conduct an analysis on [REDACTED] for just [REDACTED] transit.

table	column	total_rows	null_count	null_percentage
my_table				100.000000
my_table				100.000000
my_table				100.000000
my_table				99.901435
my_table				95.044265
my_table				81.682055
my_table				81.682055
my_table				76.772636
my_table				76.772636
my_table				55.539990

Below are the columns with 1%<x<0% null. These columns have enough data to be considered for estimating nulls, but again, it depends on how important the column is.

Below are all the fully non-null columns. These are perfect for our ML model to train on, depending on their importance and reliability. [Ridership Column] is the most important obviously, with it directly signifying ridership count.

my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000
my_table	[redacted]	0	0.000000

Column Meanings and Data Types

I did an analysis on the column meanings and what one row of data looks like. This is obviously not able to be shown in the redacted version due to the NDA. However, this is still shown to show where my analysis went next.

Second Initial Data Analysis using a ML model

I tried one more time to run a NN model and this one included 64 neurons in the first layer and 32 neurons in the second layer with one neuron for the output (the total ridership prediction). Turns out you might want to use a GPU for this type of model because my program was running for so long that my system killed it. Doing a little bit of research, I learned that this process can take hours.

Filling null values using KNN & Simple Imputer

Instead of manually assessing the null values, I wanted to try using the SimpleImputer data tool for categorical columns and KNN imputer for numeric columns and see if it got data as accurate as Brayden and Jacob's.

Unfortunately, when I started to get these systems to work, my computer would kill the process after around a minute of running it. Output below:

```
zach@10-197-152-207 ltdtest % python3 duckfileml.py
zsh: killed     python3 duckfileml.py
zach@10-197-152-207 ltdtest % python3 duckfileml.py
zsh: killed     python3 duckfileml.py
```

I then had the idea of clearing the nulls on a smaller set of my data, so that maybe in the future I could use a stronger computer or a VM to process that data without it being killed.

So I started in that process, but I had the following errors and fixes until my program finally worked:

Error: SimpleImputer was not working with certain columns that were categorical

Solution: I used mode imputation for the categories that weren't working.

Error: Completely null columns were causing the program errors because they cannot have a strategy used on them

Solution: Dropped the completely null columns

Error: I created the table with flawed data and after that it didn't work

Solution: Created an override feature to replace the existing table

Error: Upon being ambitious and trying to fill the null values of the data in batches, it could only fill one batch

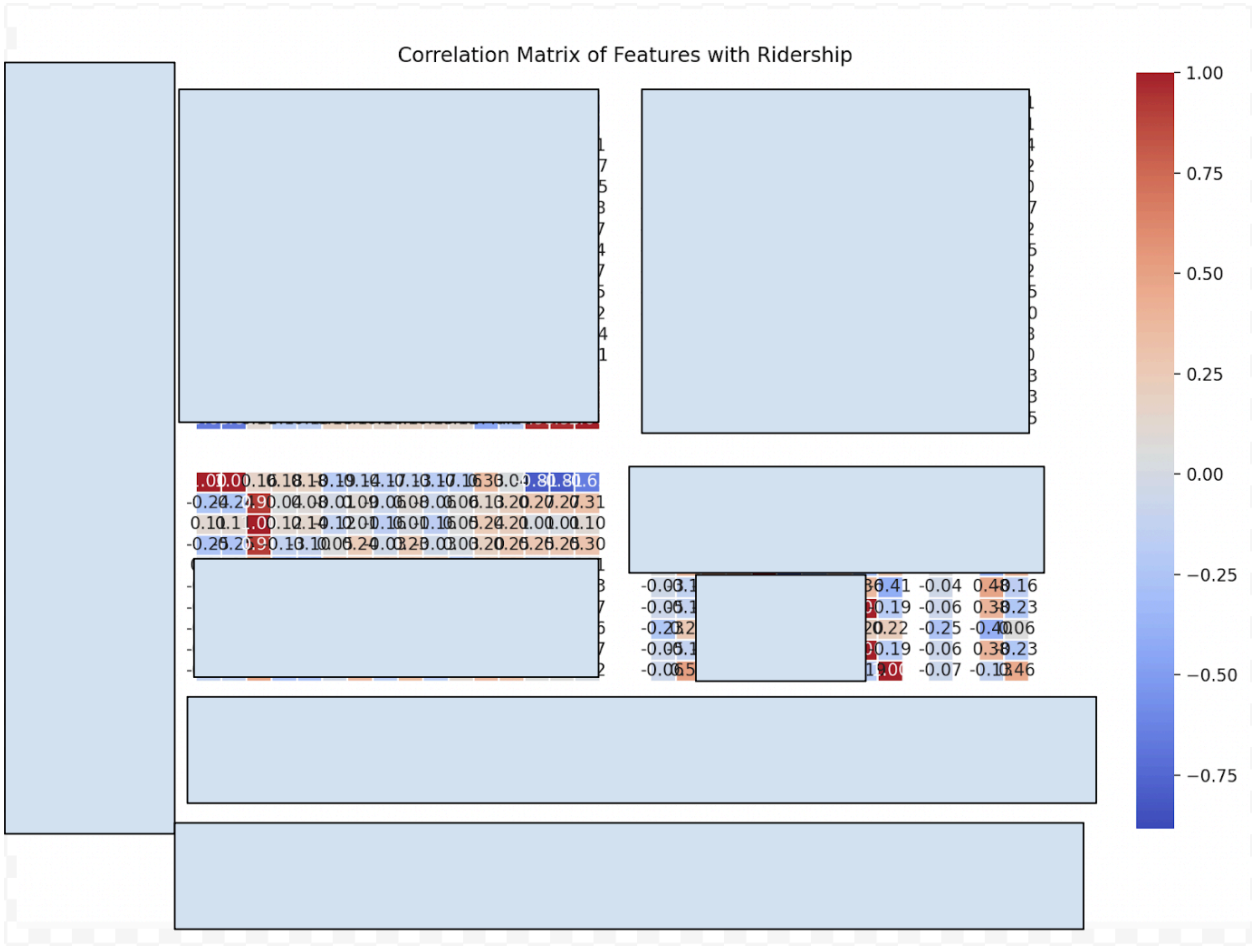
Solution: Will try to solve this problem later

Conclusion: I will test ML models on the smaller data and see how it fares in accuracy. I will come up with analysis of that smaller data and then work on moving to the bigger data if my model is accurate and efficient.

TLDR; Works with the smaller data, doesn't with the bigger data.

Training on that data and finding correlations

Obviously the main goal is finding correlations with ridership and other factors, but I included a square list of all the columns correlating with all the rest of the columns. This graph was not the easiest to read, but it does the job and shows feature importance.



Findings

From what we can see on the graph, the ridership has to do a lot with [REDACTED CATEGORY]. The [REDACTED CATEGORY] doesn't show much correlation, but this could be

how it's formatted in the system. We should use a [REDACTED CATEGORY] indicator vs. [REDACTED CATEGORY]. [REDACTED CATEGORY] has a significant negative correlation, which makes sense and reinforces that our model is at least somewhat accurate (more [REDACTED CATEGORY] should mean less passengers on board and less ridership). Two interesting values were that the [REDACTED CATEGORY] and the [REDACTED CATEGORY] had significant negative correlations on ridership. My estimate for why this is, is that the transportation system has gotten more popular, thus newer [REDACTED CATEGORY] have a higher amount of passengers. This is a good sign for LTD.