Analyzing Dockless Find-&-Ride eVehicles Trip Data for Predicting Ridership Demand in Louisville, KY

City of Louisville – Department of Public Works
Commonwealth of Kentucky – Department of Transportation
Author: **Abhiram Muktineni**

Table of Contents

Introduction

Now ubiquitous, the electric scooter cruises through bike lanes and sidewalks of every major US city. In order to stay competitive, operating companies need to ensure their scooters or e-bikes are highly utilized. They must ensure that their fleets are in place to meet demand. Using data provided by the city of Louisville, KY, I implemented various machine learning strategies to analyze historical trip data and predict optimal fleet distribution.

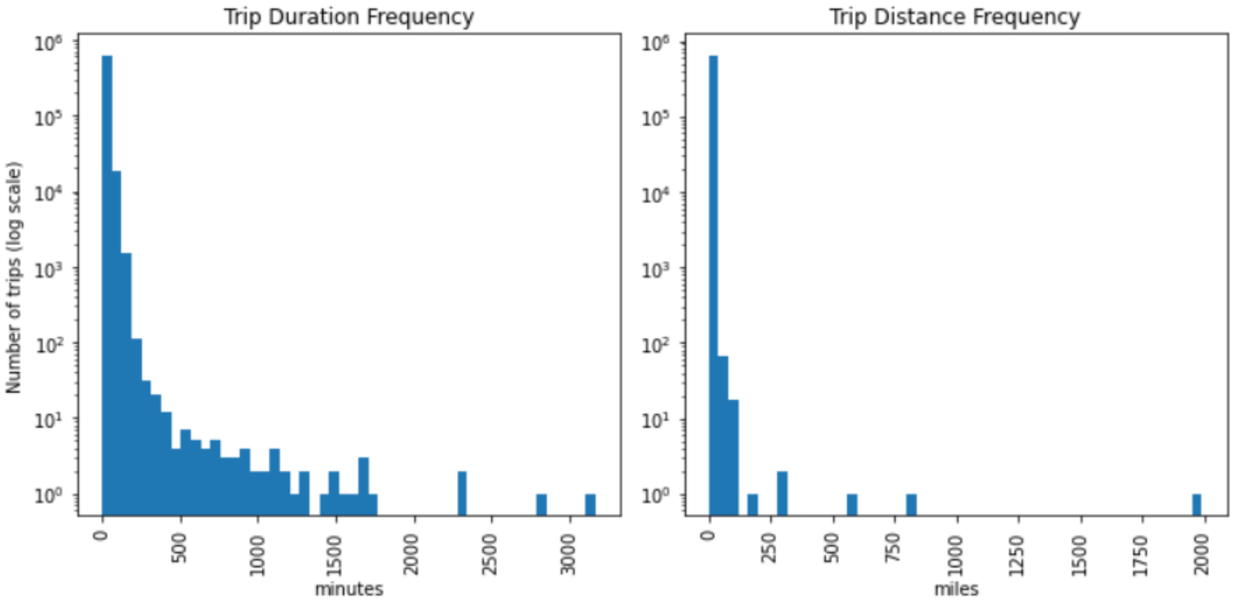
Importing and Cleaning the data

The city of Louisville publishes data for every ride taken on a scooter within the city limits. This data is provided to the city by all of the authorized micromobility service operators: Bird, Bolt, Lime and Spin. Because not all of these providers audit their data before providing it to the city, some cleaning is required to use the data.

Below are the steps I took to import, wrangle, and clean the data. The Jupyter notebook can be found here.

1. Imported the data from the csv file downloaded from Louisville Open Data: https://data.louisvilleky.gov/dataset/dockless-vehicles
    - Size: 641224 rows x 13 columns (each row represents a trip)
    - Timeframe: August 2018 to December 2020
    - Columns:
        - TripID - a unique ID created by Louisville Metro
        - StartDate - in YYYY-MM-DD format
        - StartTime - rounded to the nearest 15 minutes in HH:MM format
        - EndDate - in YYYY-MM-DD format
        - EndTime - rounded to the nearest 15 minutes in HH:MM format
        - TripDuration - duration of the trip minutes
        - TripDistance - distance of trip in miles based on company route data
        - StartLatitude - rounded to nearest 3 decimal places
        - StartLongitude - rounded to nearest 3 decimal places
        - EndLatitude - rounded to nearest 3 decimal places
        - EndLongitude - rounded to nearest 3 decimal places
        - DayOfWeek - 1-7 based on date, 1 = Sunday through 7 = Saturday, useful for analysis
        - HourNum - the hour part of the time from 0-24 of the StartTime, useful for analysis

2. There are 135231 records whose StartTime & EndTime values are equal to NaN. I have dropped these columns from the DataFrame df12 as we already have HourNum column which indicates during which hour the scooter went on a trip.
3. There are 6 rows in the weather dataframe df3 with NaN values under columns HighºF, LowºF, Precip.(inch), Snow(inch), where I had to replace the NaNs with appropriate historical weather data from a different source.
4. Dropped 4 rows with indexes 47061, 142888, 267343, 328402 in df12 dataframe as these rows have no Trip End Date, their Trip Distance and Duration is set to zero. Also their Start/End Longitude/Latitude data is ambiguous. Their Start & End values should remain the same if they haven't travelled anywhere, but that doesn't seem to be the case.
5. Replaced all T values in the Precip & Snow columns in the weather dataframe df3 with 0's.
6. Removed 85,253 excessive Trip distance and Trip Duration rows. The vast majority of the data falls within 'reasonable' boundaries for trip distance and duration. However, there are outliers spread to excessive values. In the 50-bin histograms below, these excessive values tend to only occur a handful of times. It is not possible for a trip to have a negative duration. Also, trips longer than 12 hours or 25 miles exceed the expected use for these scooters (the best batteries only last about 25 mi). I contacted the data owner, and they told me that they are working with the vendors to understand the causes of the junky data. Figures 1 and 2 show the data before and after removing these junky rows.
7. Removed vehicles data which are out of scope of this analysis.


Trip Duration Frequency

Trip Distance Frequency

Figure 1 – Trip Duration and Trip Distance frequency before removing outliers

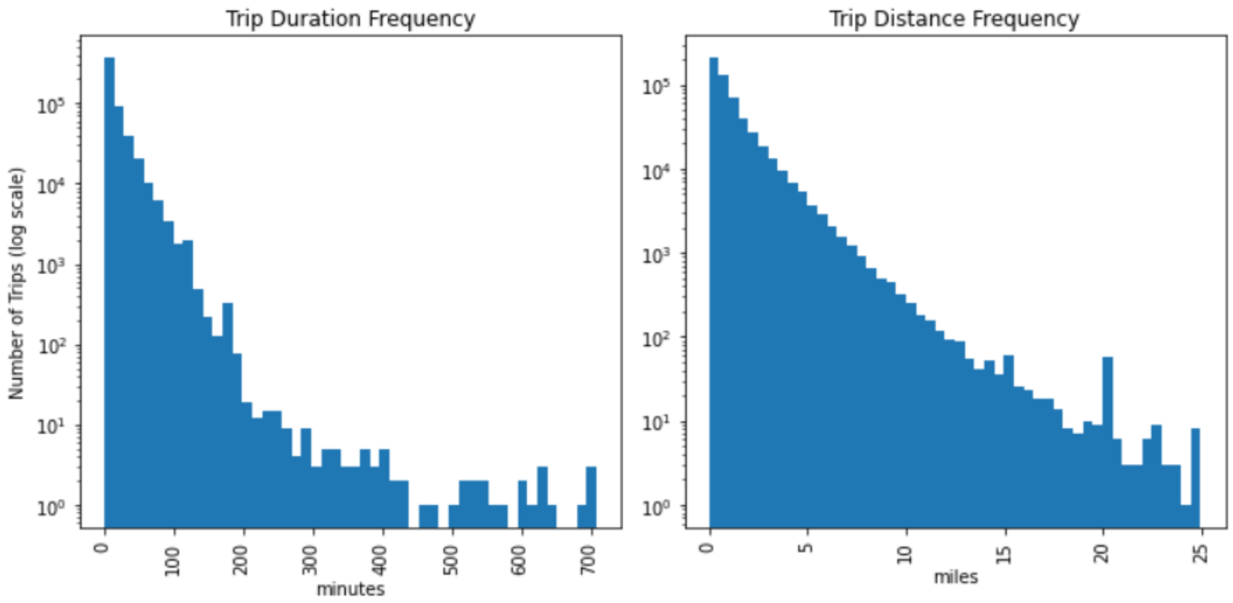


Figure 2 – Trip Duration and Trip Distance frequency after removing outliers

Exploratory Data Analysis

Once I had a clean and usable dataset, I used visualizations and statistical modeling to better understand the data.

Usage Visualization

Usage by Location

Of the data's 22 neighborhoods, usage was heavily centered in certain neighborhoods, especially the top 4 - Downtown, University, Southeast Core and Northeast Core neighborhoods as shown in Figure 3(a). Figure 3(b) shows comparison of usage in each of the top 20 coordinates. Figures 4(a) and 4(b) show the maps plotted with trips with Start Coordinates and End Coordinates respectively.

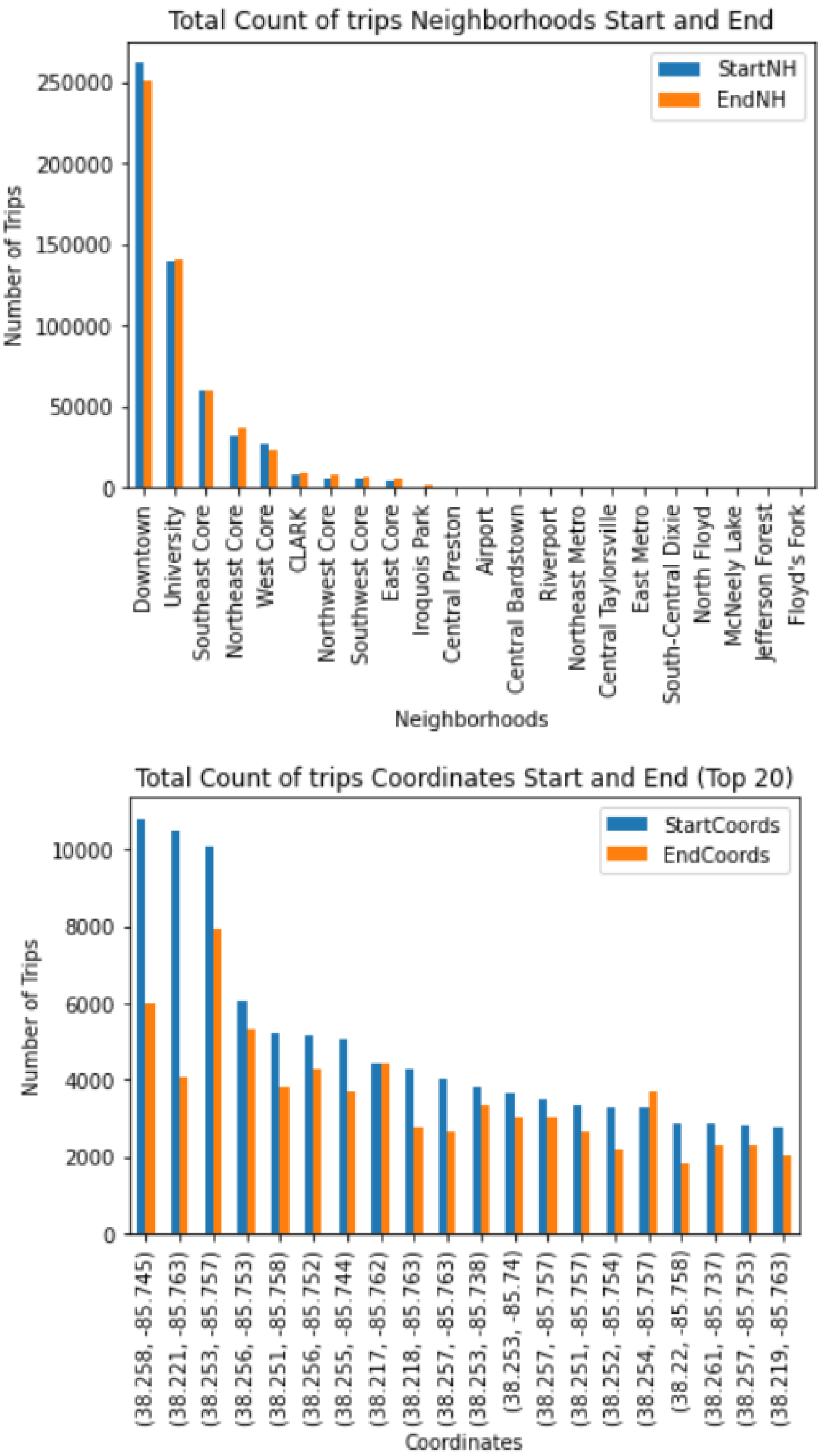


Figure 3(a) – Total count of Neighborhood(NH) and **3(b)** Coordinates Start and End Trips.

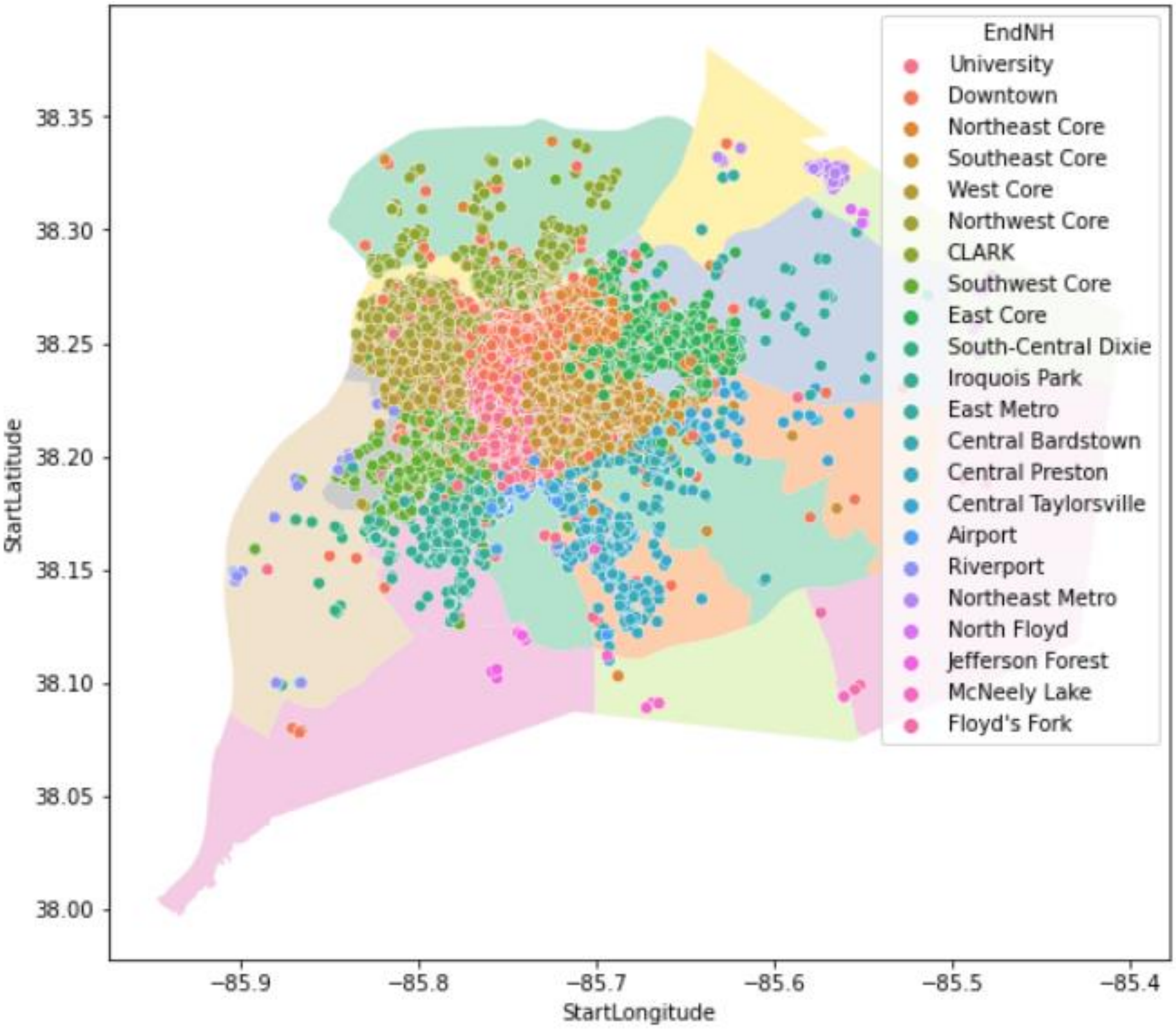


Figure 4(a) – Map of Louisville Metro plotted with the Starting locations of trips

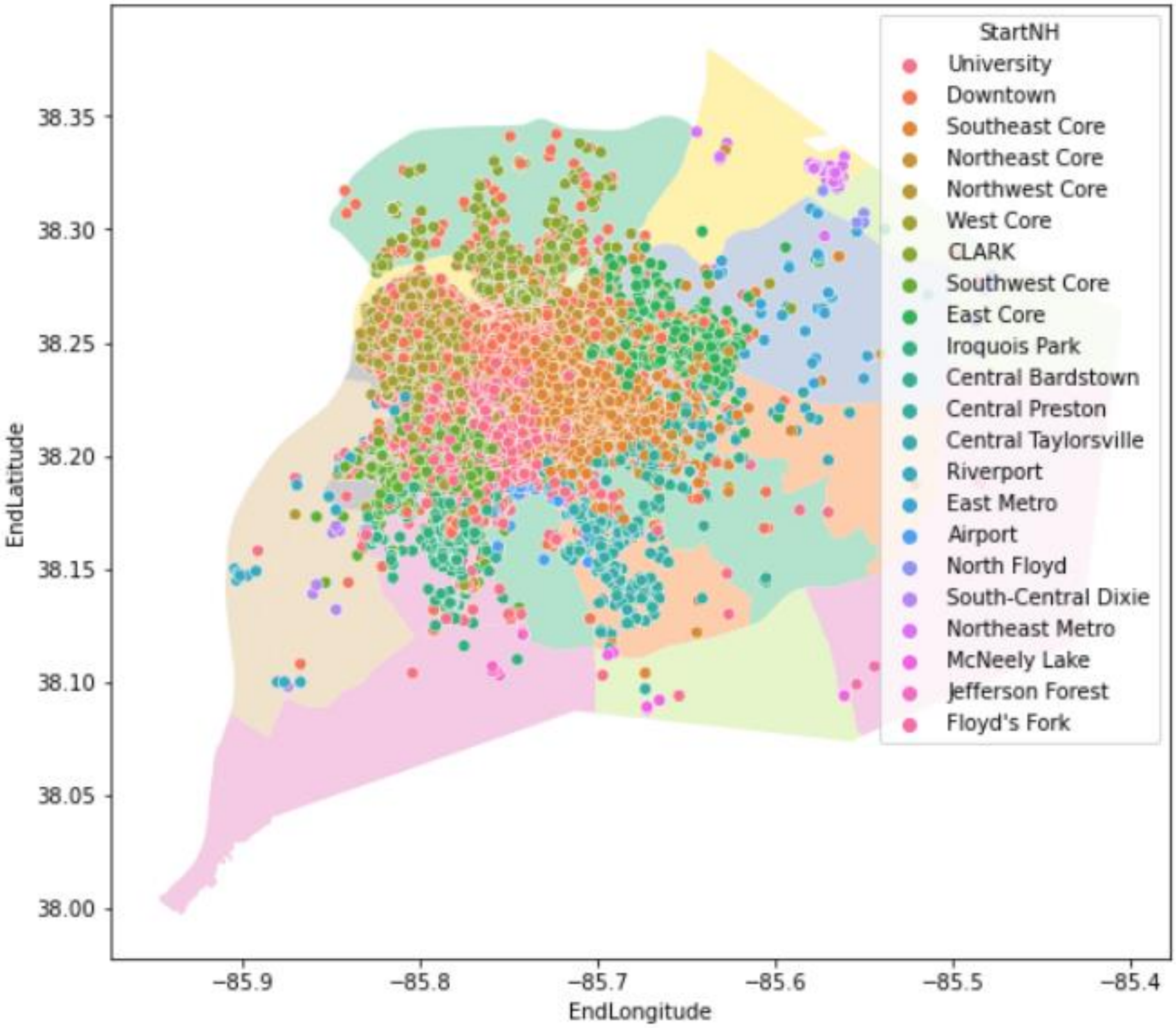


Figure 4(b) – Map of Louisville Metro plotted with the Ending locations of trips

Usage by Time

Unsurprisingly, the number of rides in a certain time period will vary depending on time of day and day of the week. In Figure 5, each day of the week has a different sized curve, but each day's usage peaks in the mid afternoon.

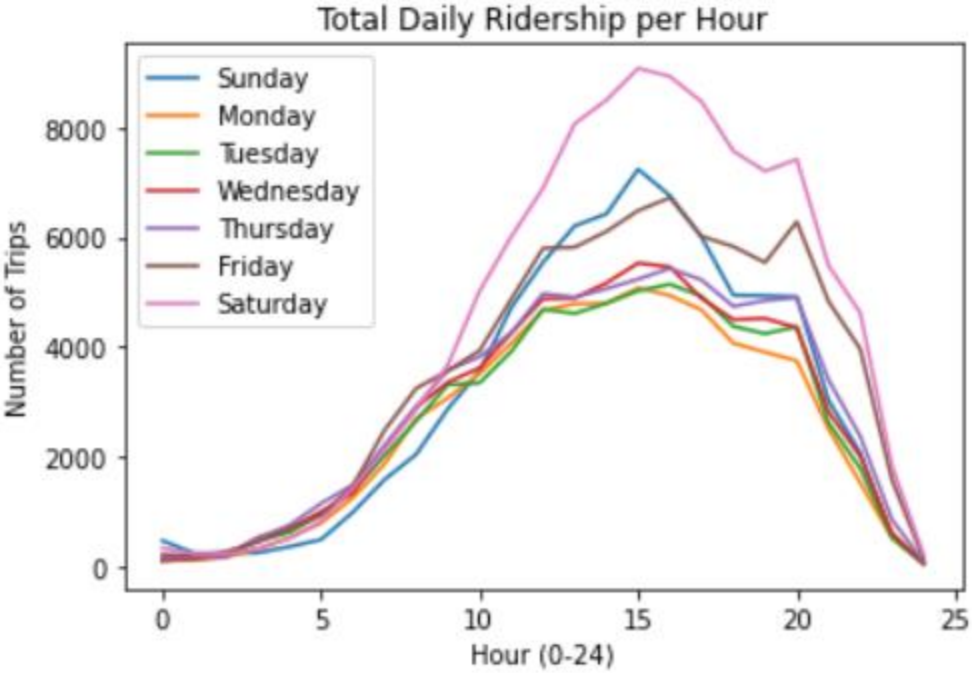


Figure 5 – Total counts of daily ridership per hour

Most scooter operators require/encourage their chargers to drop off their scooters in the early morning, during the low usage times, so this study will focus on the daily resolution.

Usage by Month

In Figure 6 , there appears to be more ridership in the warm summer months through mid fall, with peak usage in July through October.

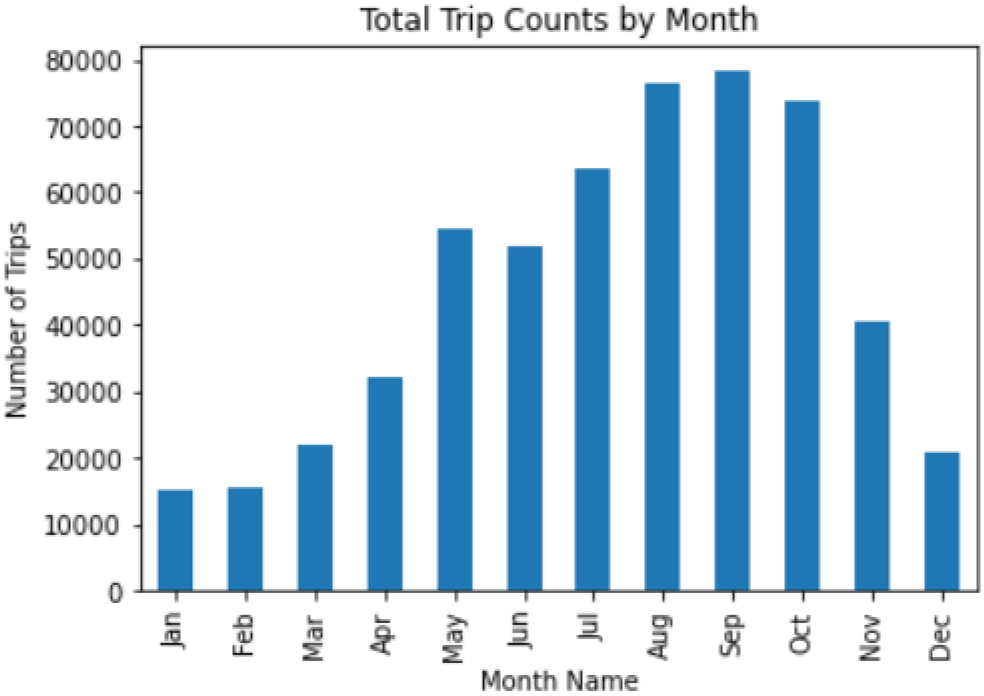


Figure 6 – Total counts of monthly ridership

Overall Usage Overtime

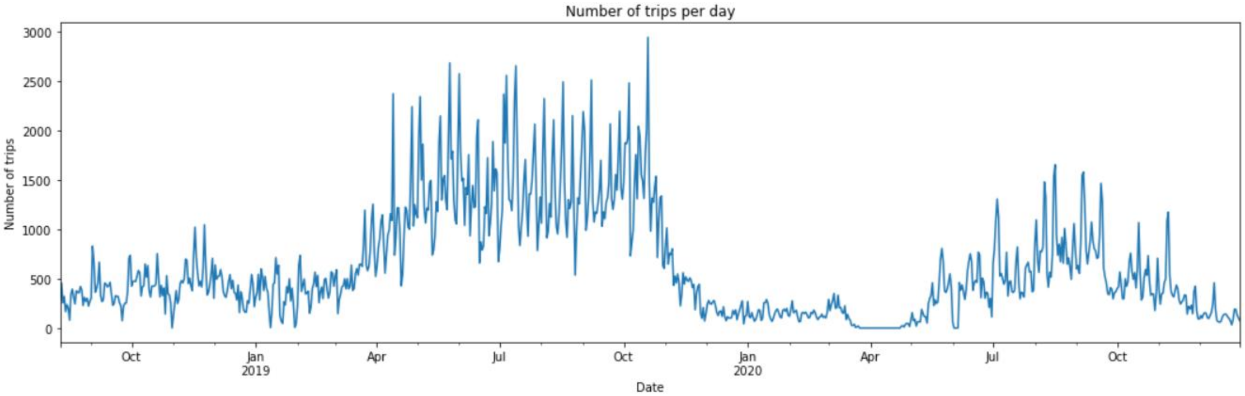


Figure 7 – Ridership overtime

In mid-October of 2019, there is a peak that coincides with Louisville's HalloScream, Jack-O-Lantern and Haunted Park after Dark geocatching Halloween events. The regularly spaced spikes in the data appear to fall on weekends. The graph also shows a drastic decrease in March/April 2020 as scooters lost popularity across the US due to increasing Covid cases during the early pandemic days.

Machine Learning Analysis

After getting a better understanding of the data, I applied some machine learning methods: Linear Regression, Ridge Regression, Lasso Regression, some Ensemble methods like Random Forest Regression and Gradient Boosting Regression.

Choice of Models

The goal being to predict future ridership demand, we treated this as a regression problem and chose models accordingly. We built the following models for the below reasons:

- **Linear Regression** – It's simple model with an interpretable equation and the coefficients give us the direction of influence on the dependent variable.
- **Ridge Regression** – Ridge regression is a way to create a parsimonious model when the number of predictor variables in a set exceeds the number of observations, or when a data set has multicollinearity.
- **Lasso Regression**– Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models.
- **Random Forest Regression** – This tree model gives more accurate results as it is an ensemble of many individual models. The random selection of features to build each model makes the trees built less correlated and so this could improve the results. Also, it gives the important features by using information gain.
- **Gradient Boosting Regression** – Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.

Choice of Metric

The metric considered to evaluate the performance of the model was RMSE (Root Mean Squared Error). It is defined as,

$$RMSE = \sqrt{\overline{(f-o)^2}}$$

where, f is forecasted values or expected values and o is observed values. It is essentially the standard deviation of the residuals. It is a numerical measure of how spread out are these residuals. This score is high when our predicted value is not close to actual values and vice versa. Lower score shows a more accurate model, excluding cases that deem over fitting.

Models

Linear Regression Model

The first model that was run was a standard linear regression model. This model is basically a linear approach to modelling the relationship between a dependent variable and one or more independent variables which hold some explanatory power.

The linear regression model gave a **root mean square error of 274** when ran against the test set.

A visualization of the fit for each individual test record can be seen below.

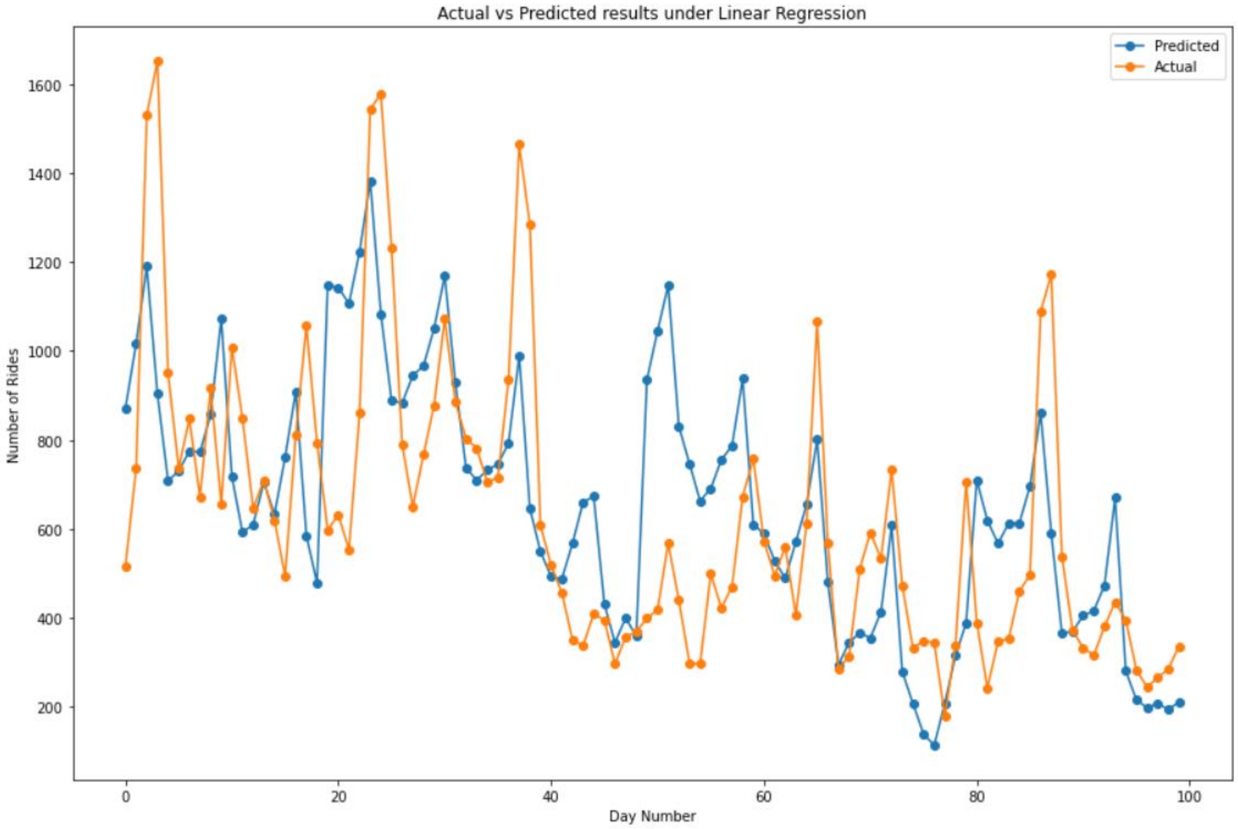


Figure 8 – Linear Regression Actual vs Predicted results

The predicted vs test values are shown in Figure 9, with a dashed line to represent where prediction and test would be equal.

Ridge Regression Model

Ridge Regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of coefficients. The ridge coefficients minimize a penalized residual sum of squares.

The ridge regression model gave a **root mean square error of 258.58** when ran against the test set.

A visualization of the fit for each individual test record can be seen below.

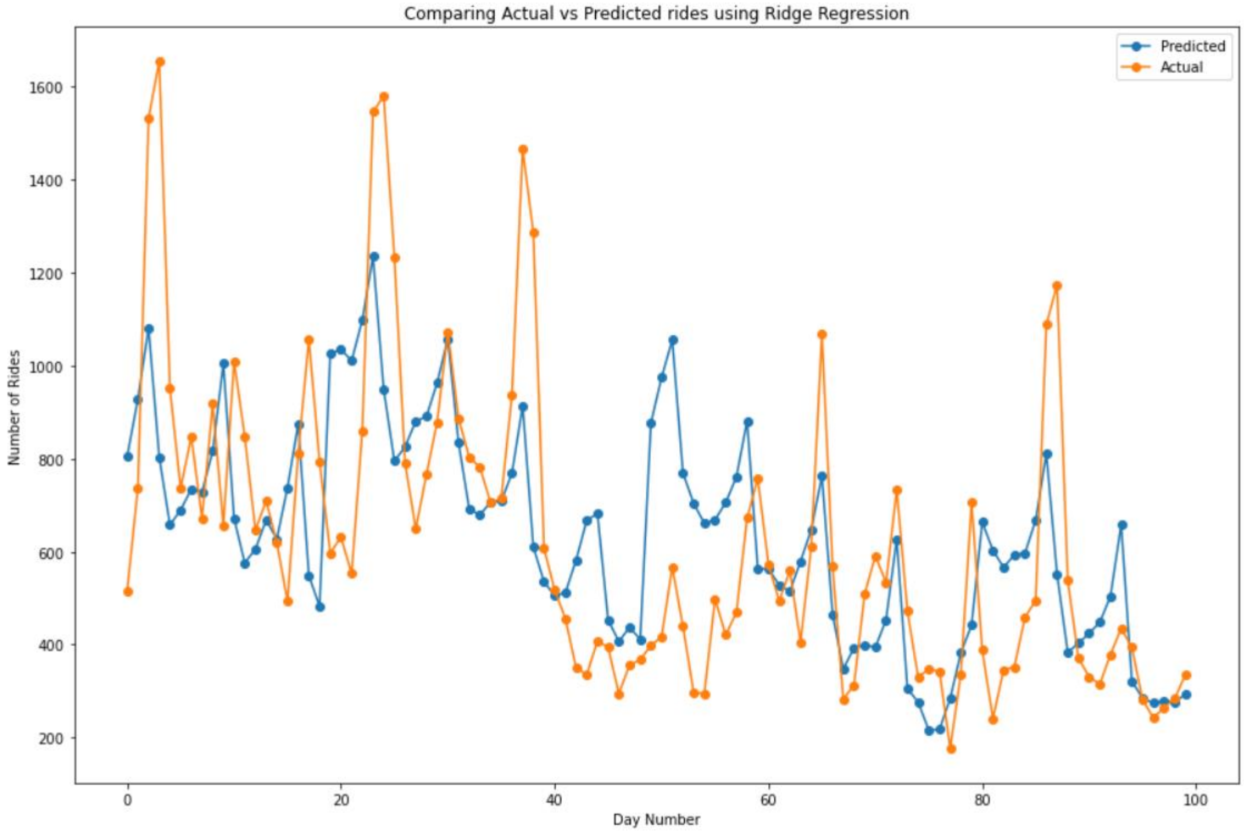


Figure 9 – Comparing Ridge Regression Actual vs Predicted results

Lasso Regression Model

Lasso Regression estimates sparse coefficients. It is useful in some contexts due to its tendency to prefer solutions with fewer parameter values, effectively reducing the number of variables upon which the given solution is dependent.

The ridge regression model gave a **root mean square error of 272.84** when ran against the test set.

A visualization of the fit for each individual test record can be seen below.

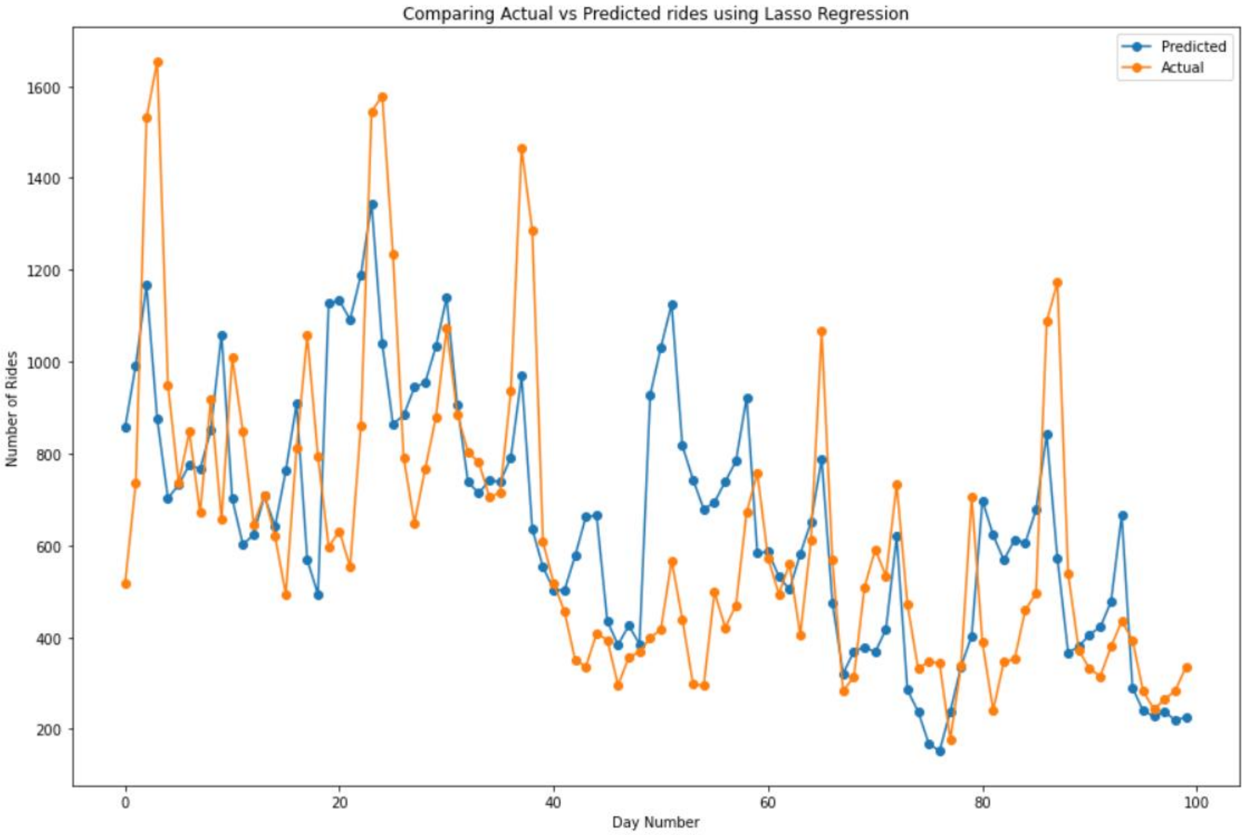


Figure 10 – Comparing Lasso Regression Actual vs Predicted Results

Random Forest Regression Model

Random Forest is an ensemble tree-based algorithm which creates multiple decision trees from randomly selected subsets of data from the training set. These decision trees are ultimately aggregated on the basis of votes from the different trees to finalize the best tree.

The random forest regression model gave a **root mean square error of 291.66** when ran against the test set.

A visualization of the fit for each individual test record can be seen below.

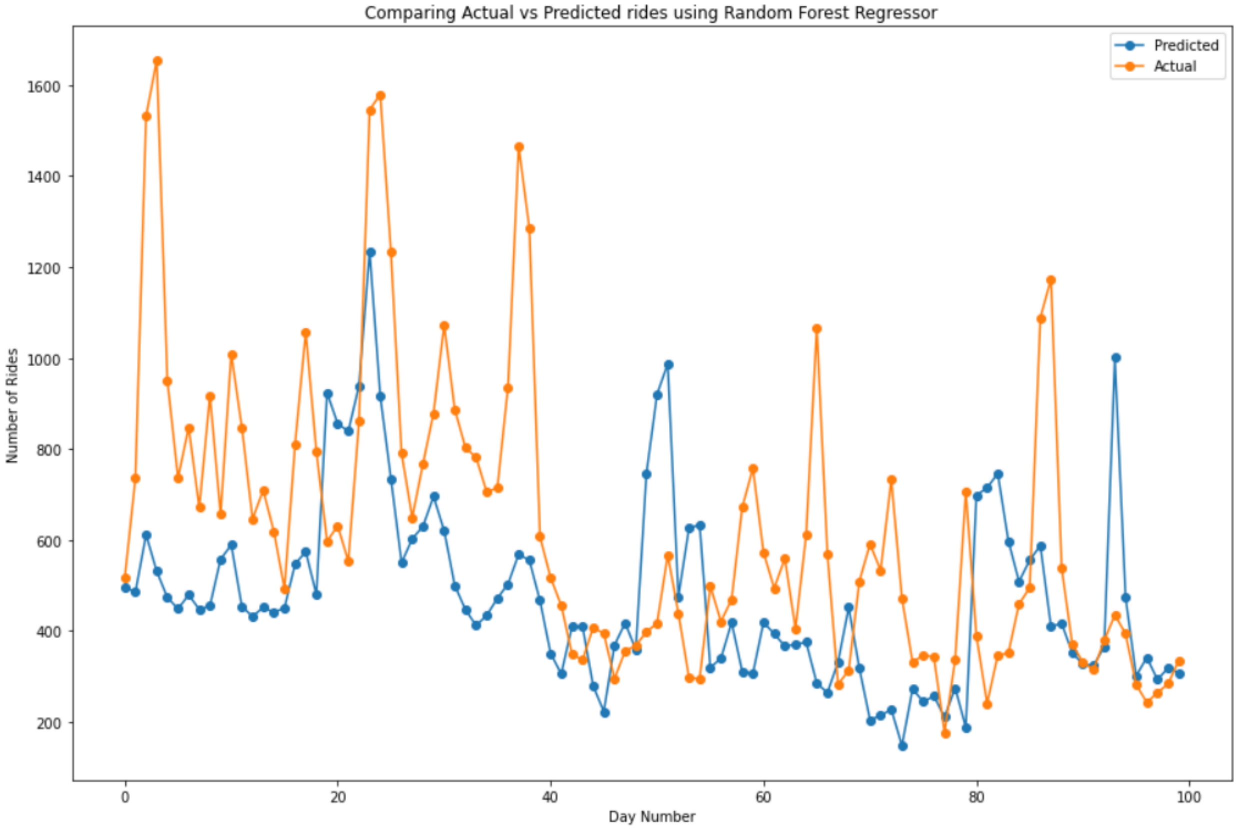


Figure 11 – Comparing Random Forest Regressor Actual vs Predicted Results

Gradient Boosting Regression Model

Gradient Boosting learns and computes new residuals at every step based on predictions made on previous steps. These will be used as leaves for the next tree. This process goes on until iterations and estimators match. The final residual is the mean of all the residuals at every step. For this model, we have given 100 estimators which mean 100 iterations take place. Gradient Boosting uses depth greater than 1.

The gradient boost regression model gave a **root mean square error of 281.72** when ran against the test set.

A visualization of the fit for each individual test record can be seen below.

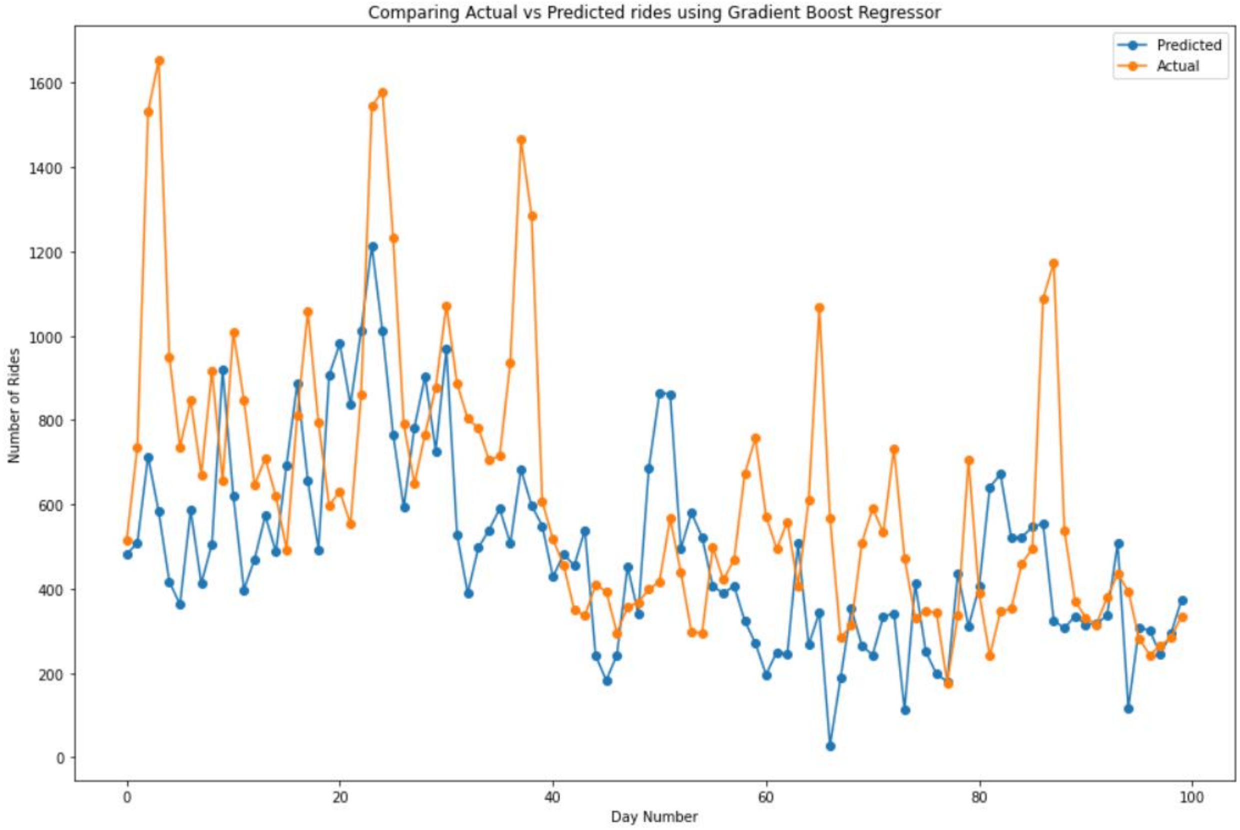


Figure 12 – Comparing Gradient Boost Regressor Actual vs Predicted Results

Model Comparison

Comparing the scores on the validation set for various models, we see that Random Forest and LSTM have the lower RMSE. LSTM has lower RMSE than that of Random Forest. So, we select the LSTM model to base our recommendations on.

Model	MAE	MSE	RMSE
Linear Regression	216.74	75410.23	274.60
Ridge Regression	198.76	66867.16	258.58
Lasso Regression	213.94	74443.99	272.84
Random Forest	212.08	89163.24	298.60
Gradient Boost	201.15	78962.08	281

A visualization of the metrics for each model can be seen below:

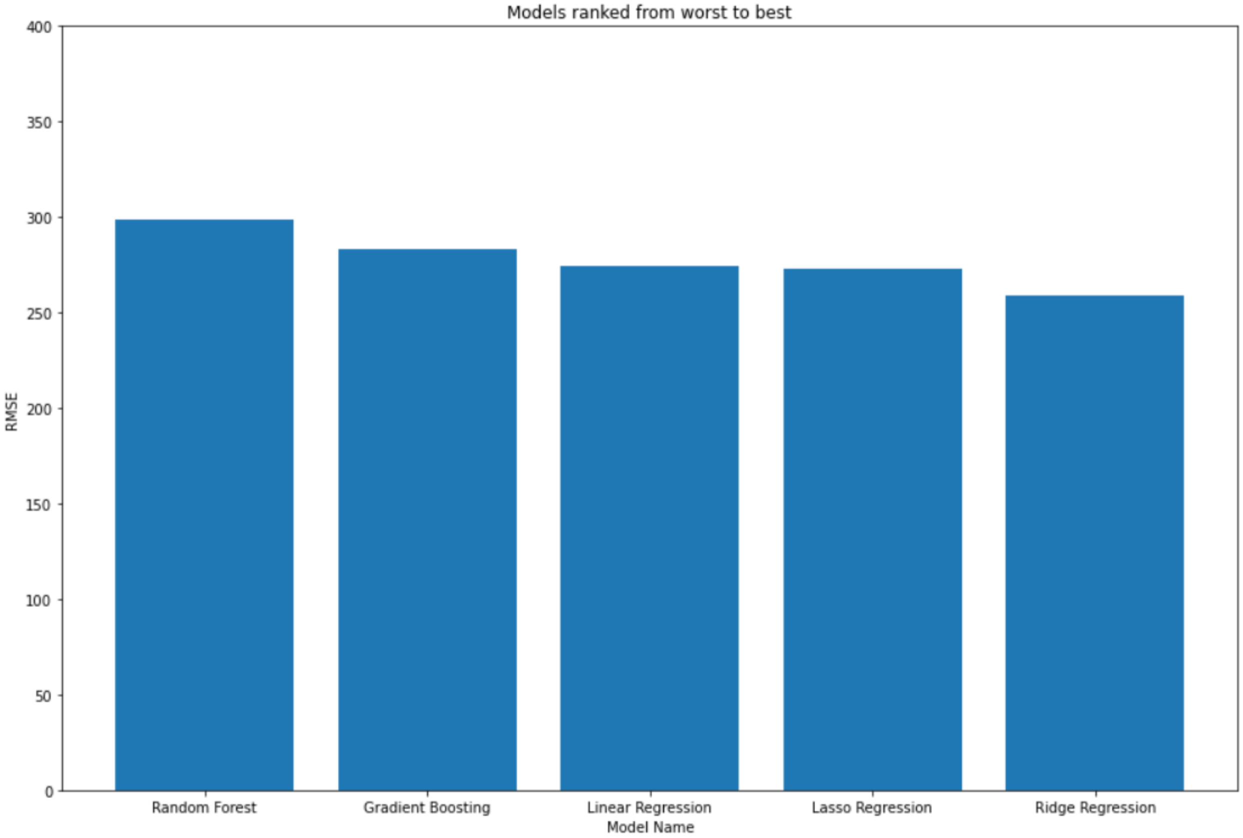


Figure 13 – Comparing Models

Recommendations

If you observe the model comparisons made in the above bar plot, it turns out that Ridge Regression model has been the best performing model over other models. Hence, we recommend Louisville Department of Public Works to deploy the Ridge Regression model in order to forecast the total daily ridership demand of electric find and ride vehicles in Louisville.

Limitations

- Up to this point, I have focused only on total number of daily rides. In reality, micro mobility operators will need to understand how to distribute their fleet across multiple neighborhoods or locations on a daily basis.
- Though the models are quite accurate, they do not exactly predict the daily rides differently for different neighborhoods. Each neighborhood has a different number of rides taking place which isn't captured in these models.

Future Research and Conclusion

- In order to harness the nature of the data in a better way, it would be wise to run other time-series models as well. One of the most popular of such models is the Facebook Prophet which is a forecasting procedure that makes predictions on time series data.
- Along with this, another possibility would be to run a mixed effect model which will be able to differentiate between the different neighborhoods and predict the daily rides differently for each neighborhood. This would be particularly useful as each neighborhood has a distinct demand which drastically affects its ridership.
- In order to make the modelling process more efficient, some modifications can also be done to the data in the future. This includes, splitting the date feature into three columns - days, month, year.
- Additionally, the dataset could have included details about special occasions such as Kentucky Derby, Christmas, pre-Christmas, Black Friday, Labor Day or any other special events happening in the city. People obviously tend to ride more or less on these days than usual. Adding these as a new feature to data will also improve accuracy to a large extent.
- An hourly model could be used if scooter providers were interested in providing a more dynamically changing fleet distribution.
- The data on which these forecasts are based is the sum total of all scooter operators in Louisville. We cannot assume that if we predict 1,000 rides in a neighborhood, that one operating company could see 1,000 of their scooters used. However, these numbers can be used to determine where to place a percentage of the fleet.