Springboard

Optimizing Scooter Utilization

Using Machine Learning to Improve Strategic Placement of Scooters in Austin, TX

Trevor Anand Bhattacharya

Image from: austin.culturalmap.com

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 Austin, TX, I implemented various machine learning strategies to predict optimal fleet distribution.

Importing and Cleaning the data

The city of Austin publishes data for every ride taken on a scooter within the city limits. This data is provided to the city by all of the authorised micromobility service operators: Bird, Jump (Uber), Lime, Lyft, OjO, Spin, and Wheels. 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 the City of Austin: https://data.austintexas.gov/d/7d8e-dm7r
    - Size: 6,848,950 rows x 16 columns (each row represents a trip)
    - Timeframe: April 2018 to September 2019
    - Columns:
        - ID: A unique ID for each trip (string)
        - Device ID: A unique ID for the device used (string)
        - Vehicle Type: Bicycle or Scooter (string)
        - Trip Duration: time length of trip in seconds (float)
        - Trip Distance: distance traveled in meters (float)
        - Start Time: trip start time (datetime)
        - End Time: trip end time (datetime)
        - Modified Date: datetime at which the record was last modified, typically when the data was extracted (datetime)
        - Month: Month when the trip occurred (integer)
        - Day of week: day of the week when the trip occurred, Sunday = 0 (integer)
        - Council District (Start): City council district in which the trip started (string)
        - Council District (End): City council district in which the trip ended (string)
        - Year: Year when trip occurred (integer)

- Census Tract Start: Starting Neighborhood GEOID number from US 2010 Census Tract (string). Note--this is an 11-digit number in which all of the tracts in Austin share the first 7 digits. Throughout this article, the last 4 digits will often be used to denote the tract.
- Census Tract End: Ending Neighborhood GEOID number from US 2010 Census Tract (string). See note above.

2. Removed 132 empty/none rows.

3. Removed 55,000 “OUT OF BOUNDS” rows

4. Removed 590,000 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 50 miles exceed the expected use for these scooters (the best batteries only last about 30 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.

5. Removed Bicycle data, which are out of scope of this analysis.

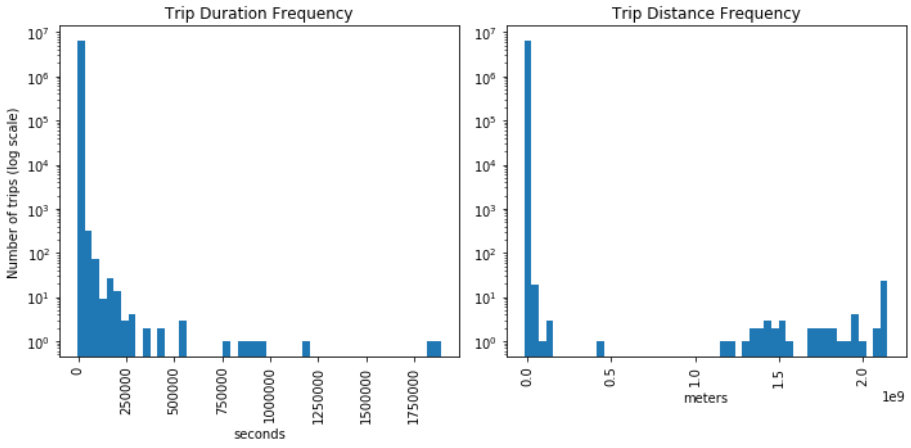


Figure 1--Trip Duration and Trip Distance frequency before removing outliers

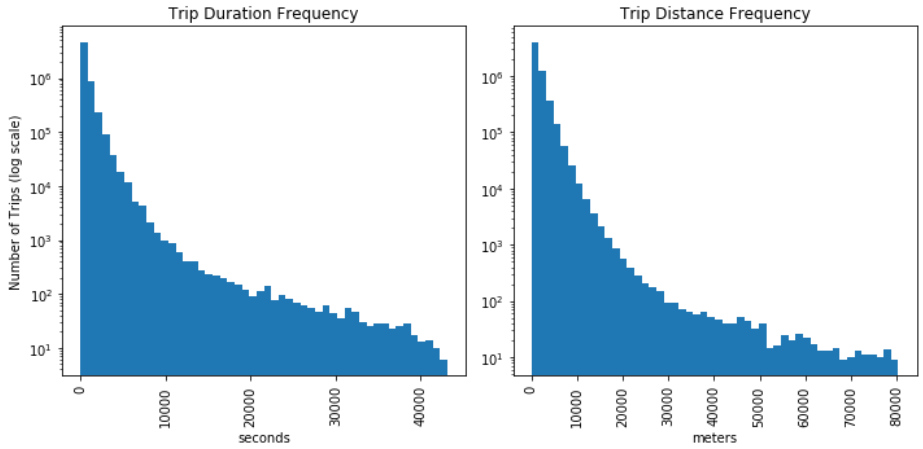


Figure 2--Trip Duration and Trip Distance after removing outliers.

Exploratory Data Analysis

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

Usage visualization

The Jupyter notebook for this section can be found here.

Usage by Location

Of the data's 271 census tracts, usage was heavily centered in certain locations, especially the '1100' census tract in the middle of downtown Austin. In Figure 3, the '1100'. Figure 4 shows a heatmap of census tract usage.

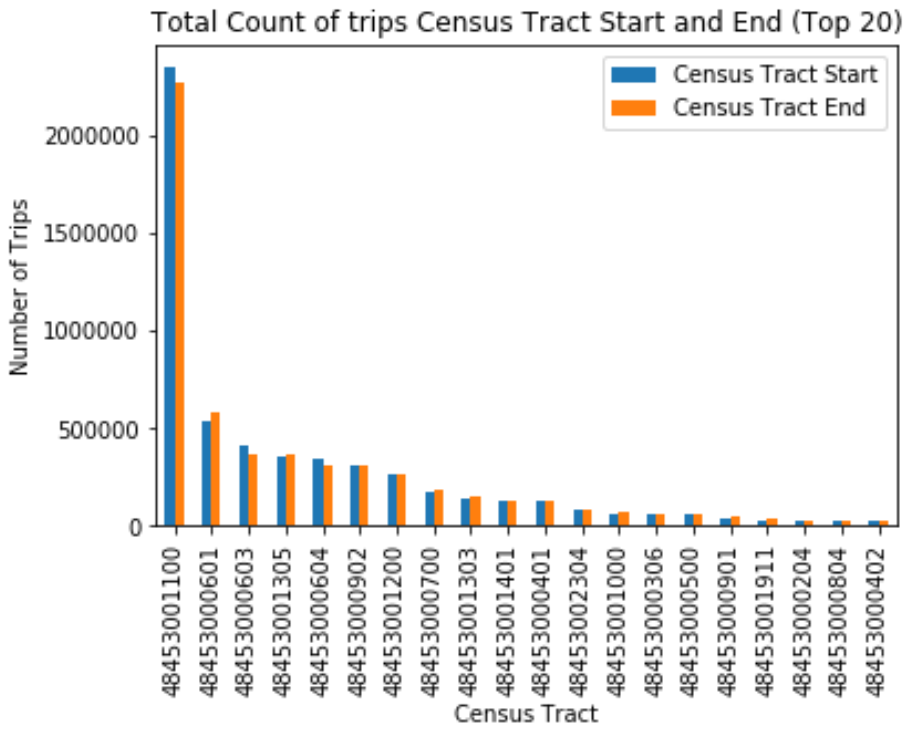


Figure 3 -- Total count of Census Tract Start and End Trips

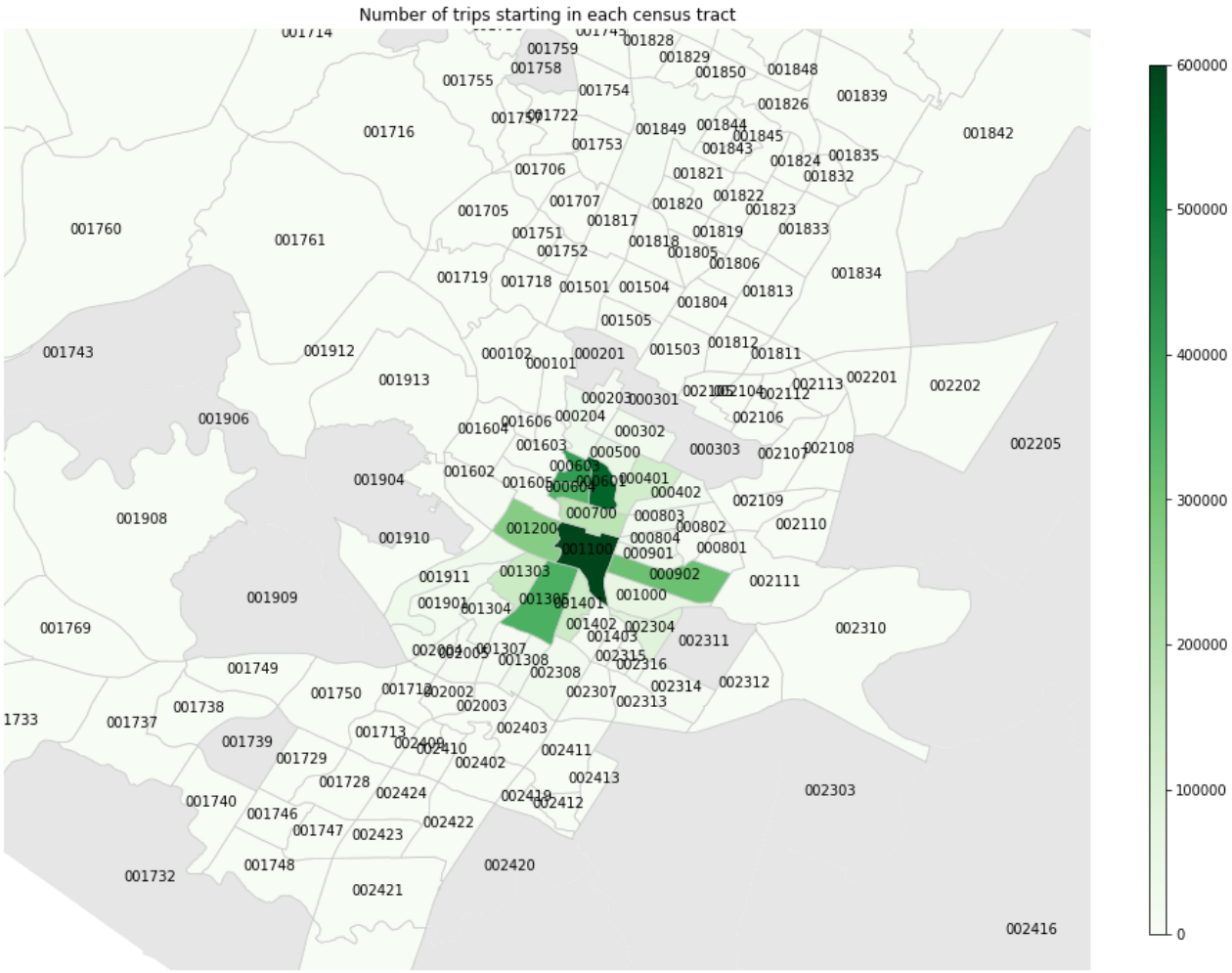


Figure 4--Heatmap of census tract ridership

In Figure 5, trips tend to stay within the central area (the dark blue lines), with a only few exceptions venturing to outer census tract neighborhoods (the light colored lines). This figure shows only trips in '1100', but this pattern is typical of the high-volume downtown census tracts.

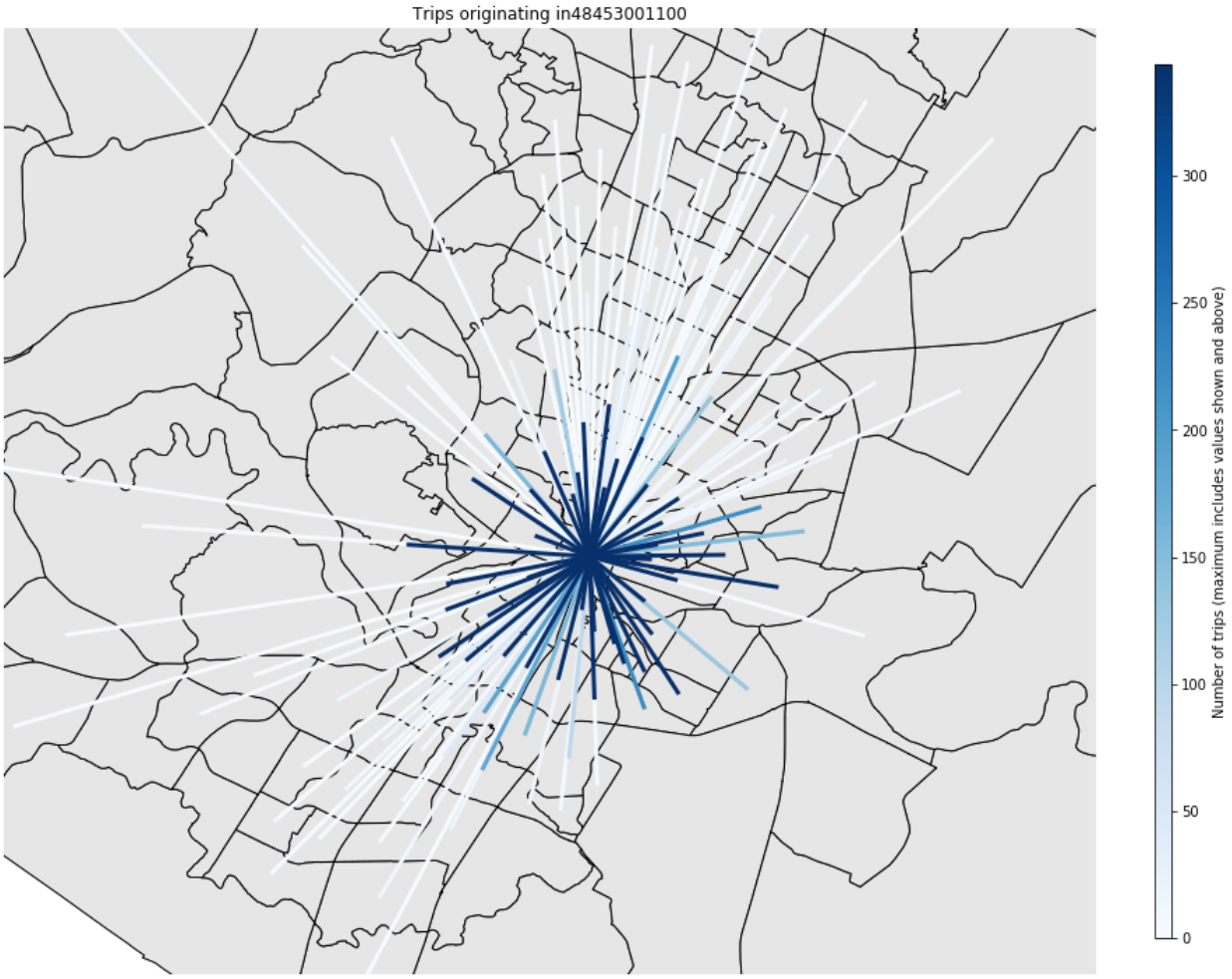


Figure 5--Trip routes originating in census tract 48453001100

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 6, each day of the week has a different sized curve, but each day's usage peaks in the mid afternoon.

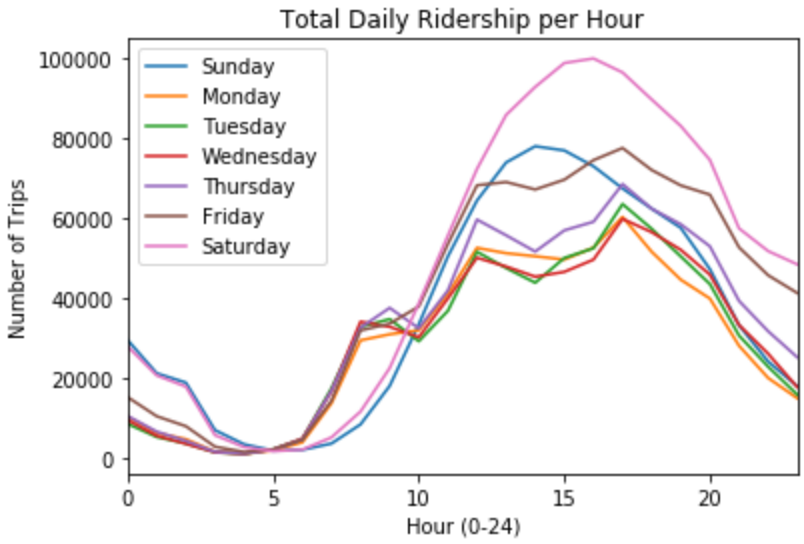


Figure 6--Total Daily Ridership per Hour

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.

Statistical Modeling

The Jupyter notebook for this section can be found here.

To get an idea of how the data is distributed, I created a model to simulate a sample of the data using a Markov Chain Monte Carlo (MCMC) method. A better understanding of the distribution will influence the machine learning approaches in the next section.

Since the data is discrete, I used a Poisson distribution:

$$P(k \text{ events in interval}) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Using PyMC3 to generate posteriors for λ, I set up the distribution and data so that λ will also represent the expected daily number of trips .

It's clear from the aggregated daily trips in Figure 7 that the expected value in a given day changes as time moves forward. The analysis can be set up to attempt to account for changes in trend over time. To account for this, we can attempt to understand when a change occurred. That is, at what point (let's call it τ , the red line in Figure 7) did the behavior change?

$$\lambda = \begin{cases} \lambda_1 & \text{if } t < \tau \\ \lambda_2 & \text{if } t \geq \tau \end{cases}$$

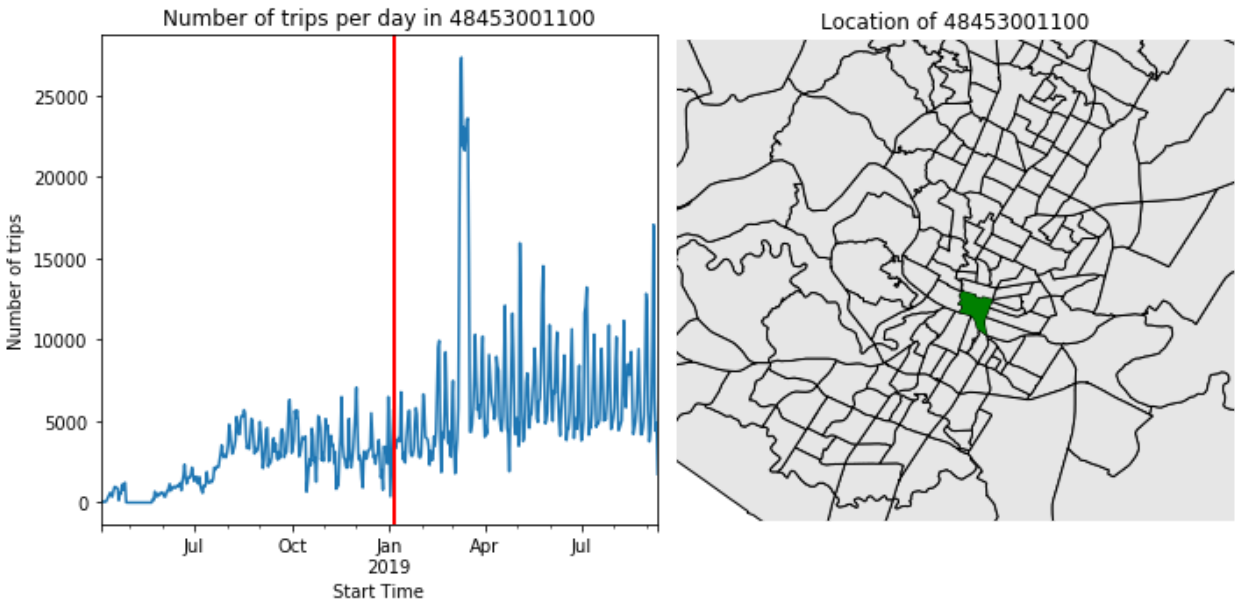


Figure 7--Number of daily trips in census tract 48453001100

Figure 8 shows the results of the posterior for λ_1, λ_2, and τ for census tract 48453001100.

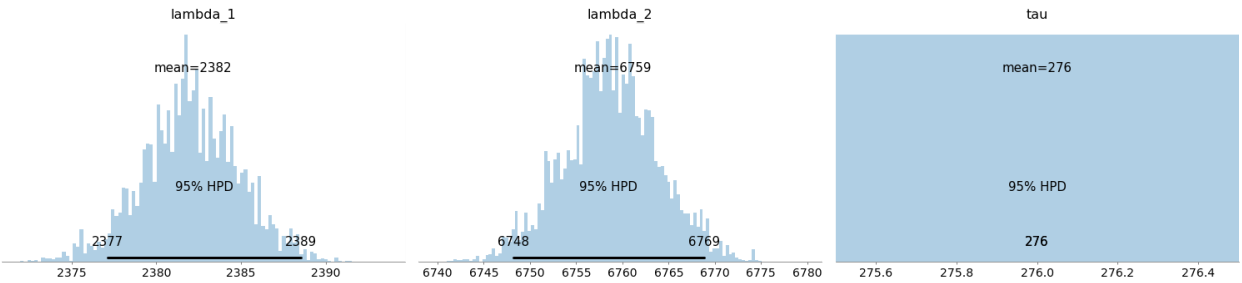


Figure 8--Posteriors for Lambda before and after a changepoint at tau

Tau falls solidly at 276 days, which would be the the beginning of January 2019.

The graphs in figure 9 show lambda_1, lambda_2, and tau for the census tracts for which lambda was greater than 50.

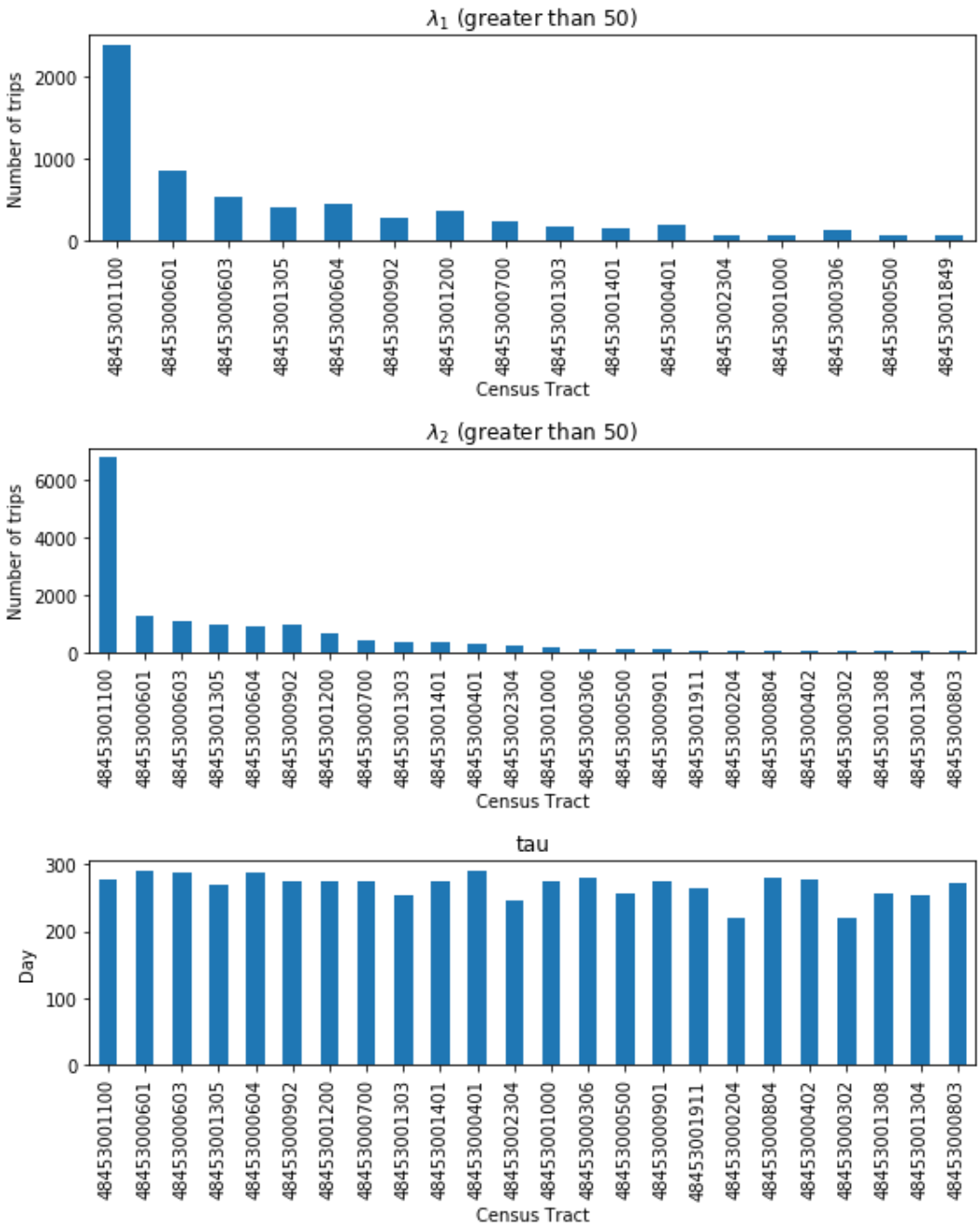


Figure 9--Lambda1, Lambda2, and tau for the busiest census tracts.

The graph of lambda_2 above creates a predicted daily count of trips for the most popular census tracts (not yet taking into account any seasonality) .

Tau is in the upper 200s for most of the census tracts, which may coincide with the steady increase in scooter popularity in 2019. However, if the increase is indeed steady and not

abrupt, choosing a single τ may not be completely accurate. The next section will cover this in more depth.

Machine Learning Analysis

The Jupyter notebook for this section can be found here.

After getting a better understanding of the data, I applied some machine learning methods: Linear Ridge Regression, Batch Gradient Descent, and finally Facebook Prophet for time series.

Linear Approaches

Ridge Regression

First, I attempted to create a linear model without using time-series. To try to capture seasonality in the data, I encoded the date column into 374 binary columns to represent year, day of year, day of the month, and day of the week.

On the 48453001100 census tract, the ridge regression had an poor r2 score (0.355) and a very large mean absolute percent error. The predicted vs test values are shown in Figure 10, with a dashed line to represent where prediction and test would be equal.

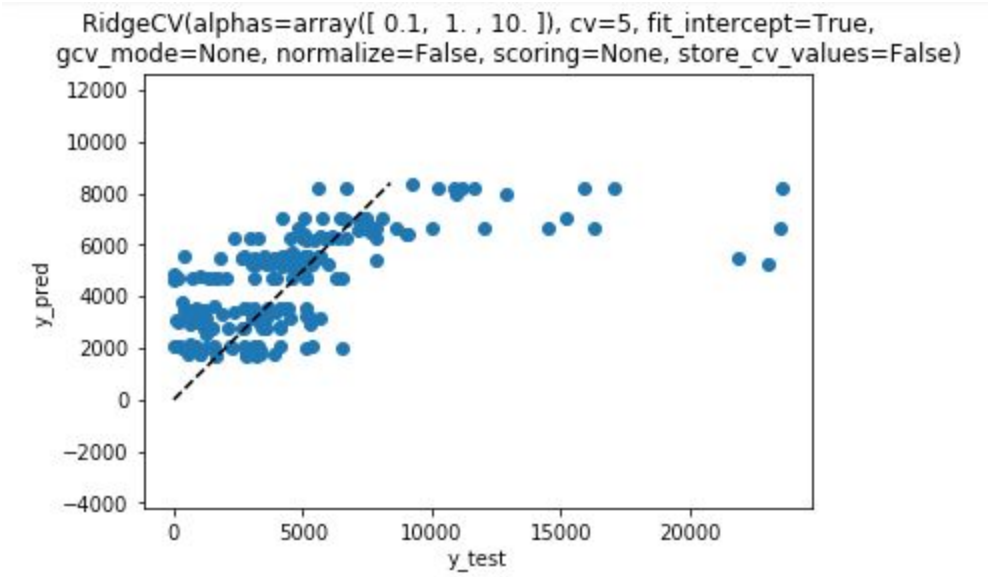


Figure 10--Ridge Regression Results

Batch Gradient Descent

Using the same encoded data, I attempted a batch gradient descent method with learning rate and number of iterations as hyperparameters. These hyperparameters might give me better 'handles' to tune the model.

Even with the quickest convergence, shown in Figure 11, the r2 score was a dismal 0.192, and still resulted in a very large mean absolute error.

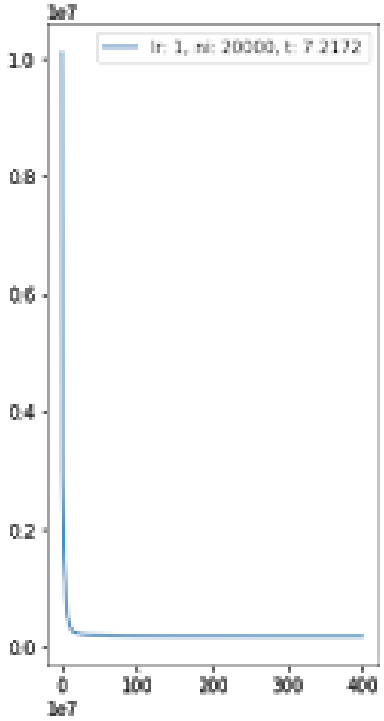


Figure 11: Convergence of the batch gradient descent model

Time Series and Facebook Prophet

Linear regression having failed me, I turned to time-series.

Facebook Prophet is a forecasting procedure that makes predictions on time series data. The major tunable hyperparameters of the Facebook Prophet model are trend, holidays, and seasonality (Letham 7).

Trend and Changepoints

The first component is Trend. Facebook Prophet accounts for the way that trends change over time using changepoints in a more sophisticated way than the MCMC methods above. Instead

of picking a τ for a single changepoint, Facebook Prophet defines a trend slope and searches for places where that slope changes.

In Figure 13, the changes in the slope of the trend line correspond to changepoints in the general trend of the daily use data. These changepoints are highlighted in Figure 12.

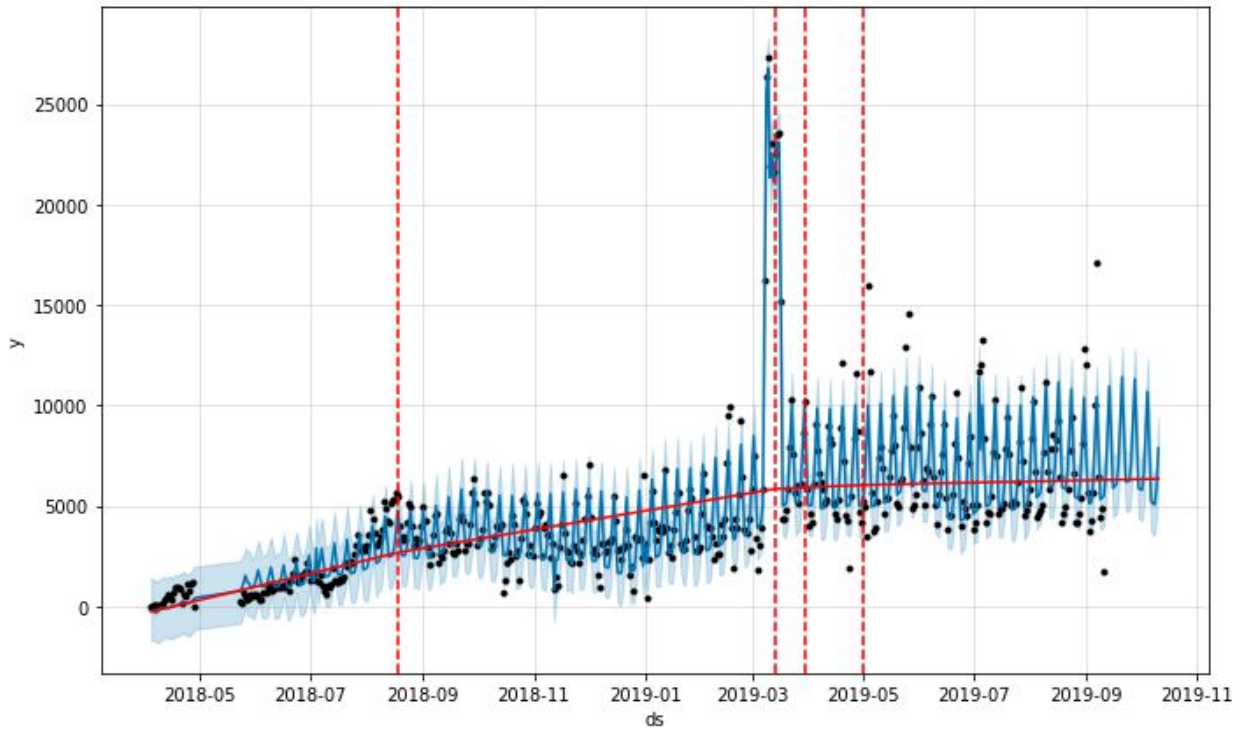


Figure 12: Regression line and confidence interval of predictions by Facebook Prophet for 48453001100

Holidays and South by Southwest

Facebook Prophet then accounts for holidays. At the beginning of March the city of Austin sees a major influx of visitors attending the popular South by Southwest conference. Facebook Prophet takes a custom dataframe of major events and US holidays and measures their effect on the prediction. Their effects can be seen in the holidays graph of Figure 12, notice the largest spike at the beginning of March.

Seasonality

Third, Seasonality can be specified yearly, monthly, weekly, and daily. In this model, weekly and yearly seasonality effects are shown in figure 13.

Not surprisingly, the weekly effect surges on weekend, and there is a general increase in the warm summer months.

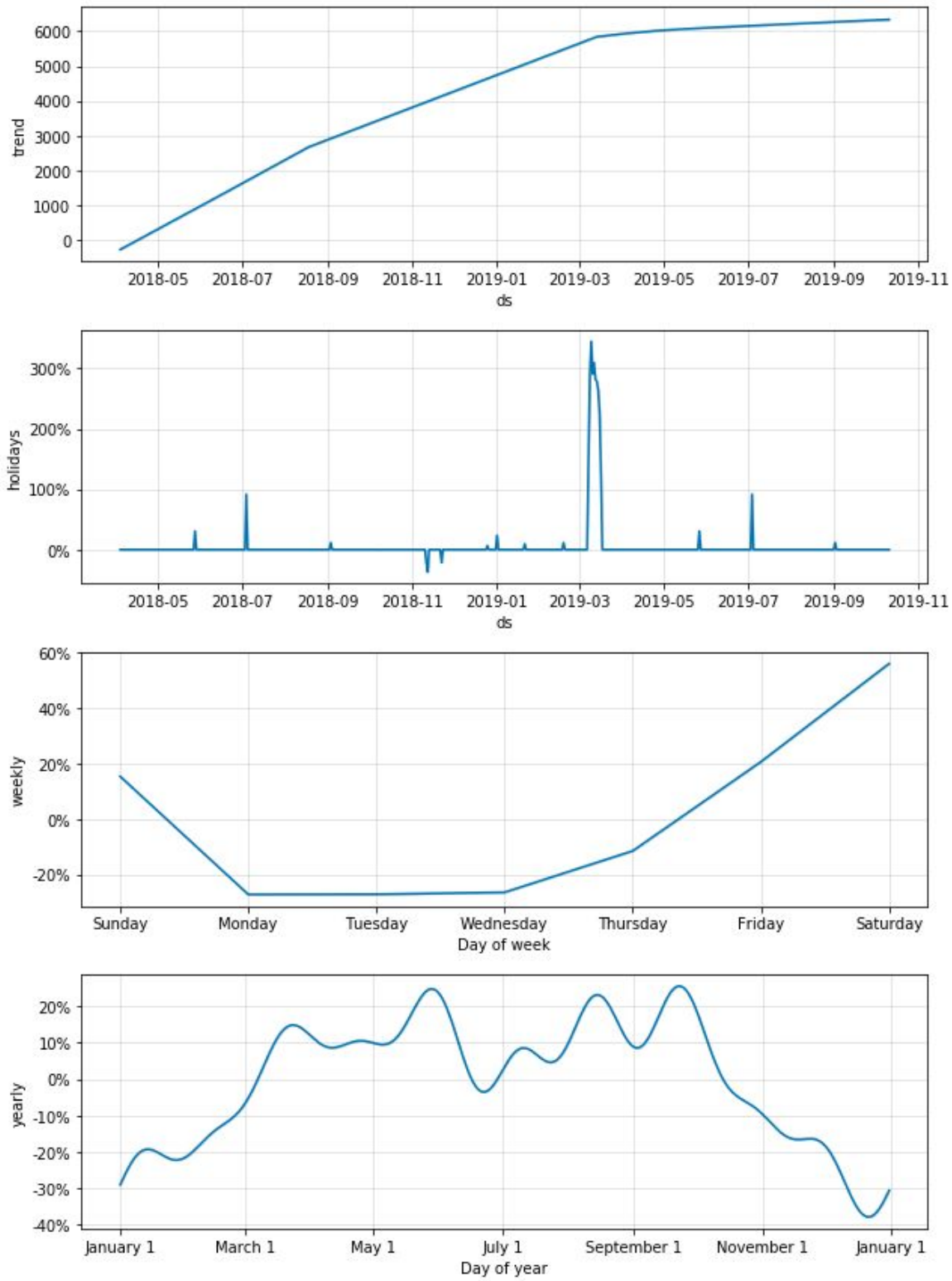


Figure 13-- Trend, Holidays, and Weekly and Yearly seasonality

Business Impact

Up to this point, the examples have focused on the ‘1100’ census tract. In reality, micromobility operators will need to understand how to distribute their fleet across multiple census tracts.

Fleet Usage

The data on which these forecasts are based is the sum total of all scooter operators in Austin. We cannot assume that if we predict 1,000 rides in census tract X, 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. In Figure 14 I generated an area plot of the forecasted percentage distribution in the top 10 census tracts.

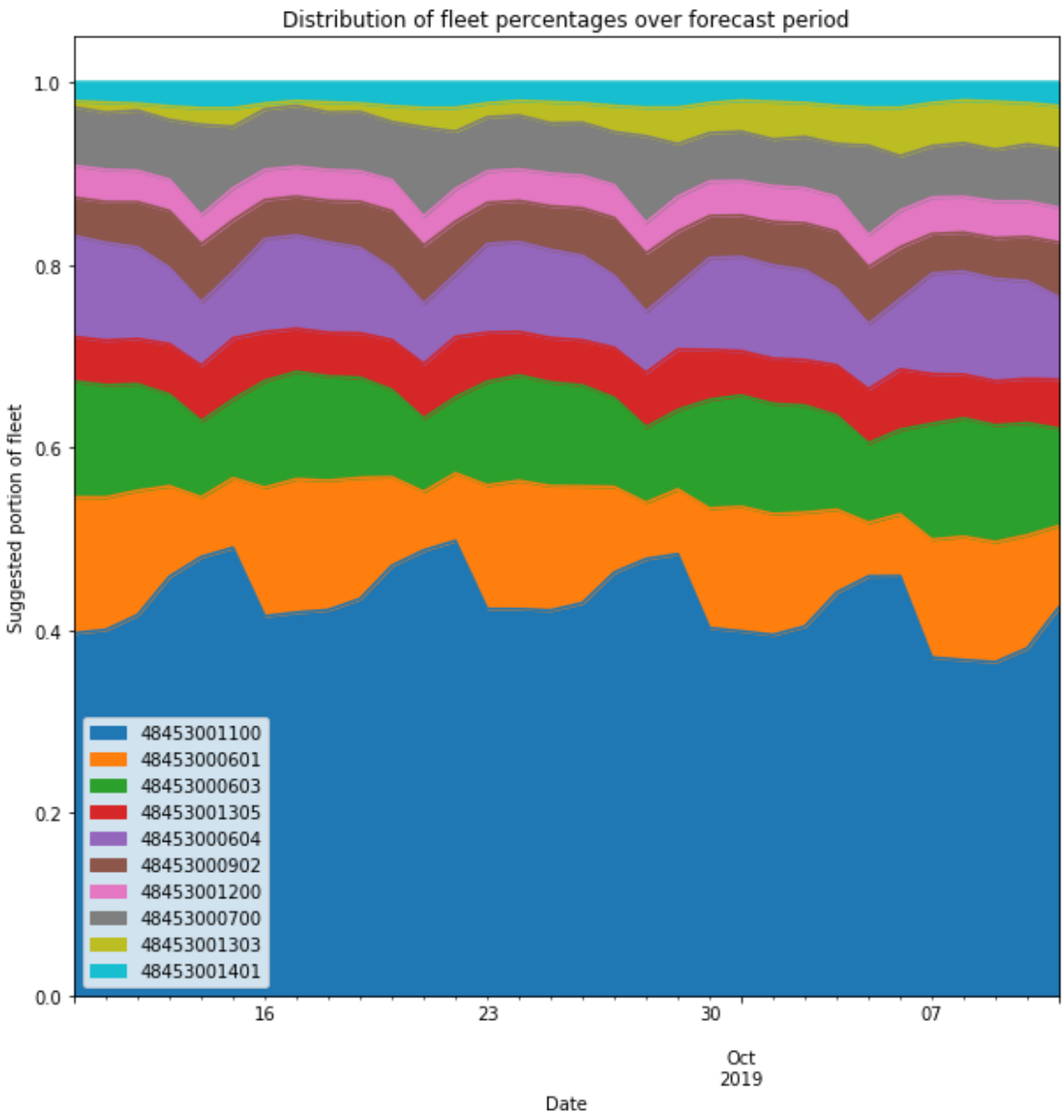


Figure 14--Recommended fleet distribution.

Suppose a provider wanted to place scooters more aggressively to increase their share of the market. By default, Facebook Prophet predicts with an 80% uncertainty margin. The lower and upper bounds of the uncertainty could be used to calculate the percentage of fleet to use depending on how aggressive they wanted to be. For example, setting 48453001100 and 48453000601 as aggressive yields a slightly different composition in Figure 15 than the composition in Figure 14. The orange and blue sections are thicker than before. Note that it

would not be meaningful to set all census tracts as either ‘aggressive’ or ‘conservative, because the terms are relative to the other census tracts.

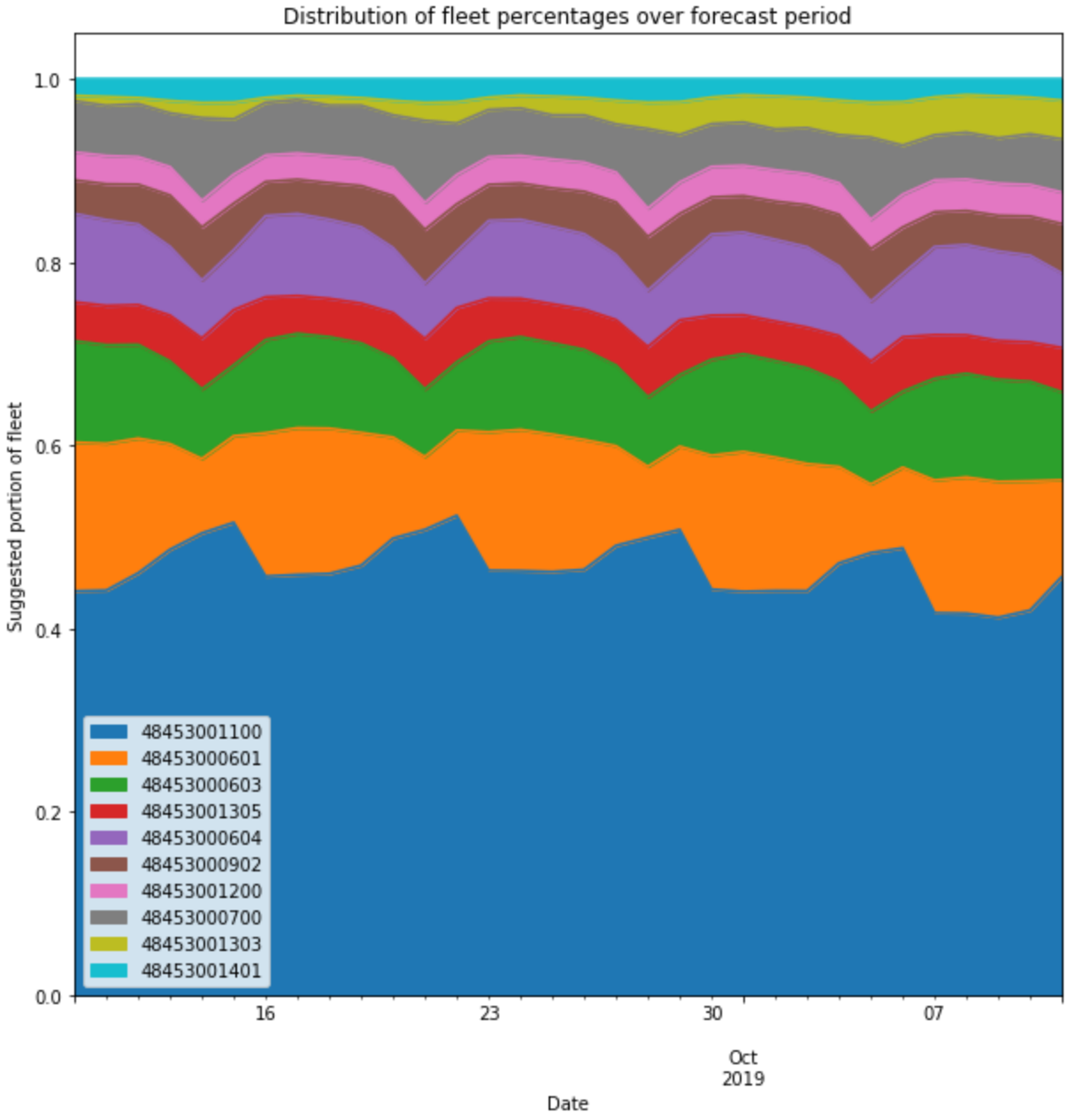


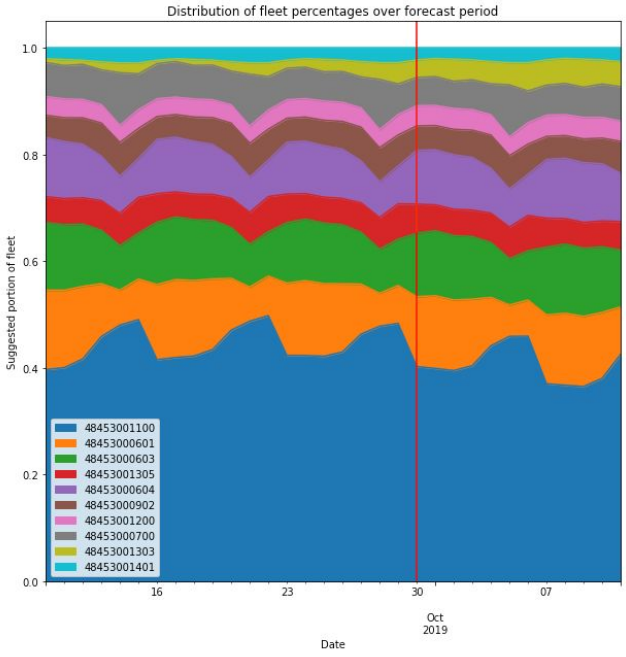
Figure 15--Recommended fleet distribution with two ‘aggressive’ estimates

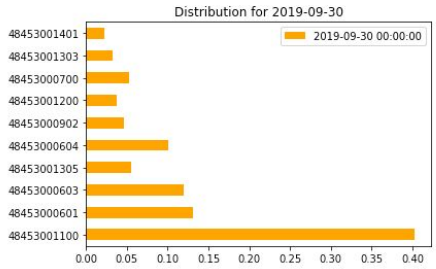
Daily Dashboard

Ideally, this system would be integrated with the scooter operator's mobile app for the employees and contractors who place the scooters. However, to illustrate the business

application of the model, I made the distribution into a daily dashboard that shows where in the prediction period the forecast was made, the percent distribution and the number of scooters to deploy for a given fleet size. I arbitrarily chose 1200 as the fleet size and made a prediction for September 30, 2019 to generate the dashboard for that day.

```
1 dashboard('2019-09-30', fleet_size=1200)
```





	Percent of Fleet	Number of Scooters to Deploy
48453001100	0.402735	483
48453000601	0.130978	157
48453000603	0.118969	142
48453001305	0.054598	65
48453000604	0.100748	120
48453000902	0.045842	55
48453001200	0.037861	45
48453000700	0.053066	63
48453001303	0.032824	39
48453001401	0.022378	26

Figure 16 -- Example Dashboard

Future Enhancements

Below are some ideas for future enhancements to the model.

1. An hourly model could be used if scooter providers were interested in providing a more dynamically changing fleet distribution.
2. Weather data might improve the predictions of the model. Weather forecasts could be combined with historical weather data to influence the model as a type of 'holdiay' seasonality.

Resources

P. Bazin, "Linear Regression: Implementation, Hyperparameters, Comparison - Pavel Bazin: Software Engineering, Machine Learning," *Linear Regression: Implementation, Hyperparameters, Comparison*, 26-Jan-2018. [Online]. Available: http://pavelbazin.com/post/linear-regression-hyperparameters/. [Accessed: Dec-2019].

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Taylor SJ, Letham B. 2017. Forecasting at scale. PeerJ Preprints 5:e3190v2 https://doi.org/10.7287/peerj.preprints.3190v2

C. Davidson-Pilon, "Probabilistic-Programming-and-Bayesian-Methods-for-Hackers," *GitHub*. [Online]. Available: https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers. [Accessed: Dec-2019].