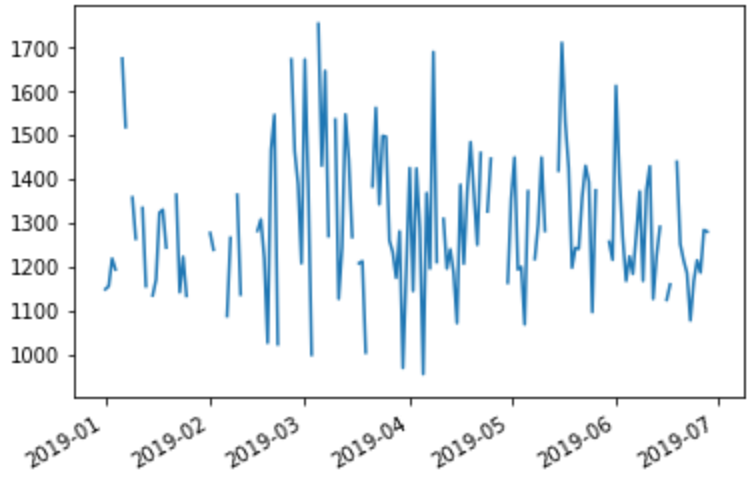
# Uber Travel Data Analysis and Forecast

**Exploring/Cleaning the data:**

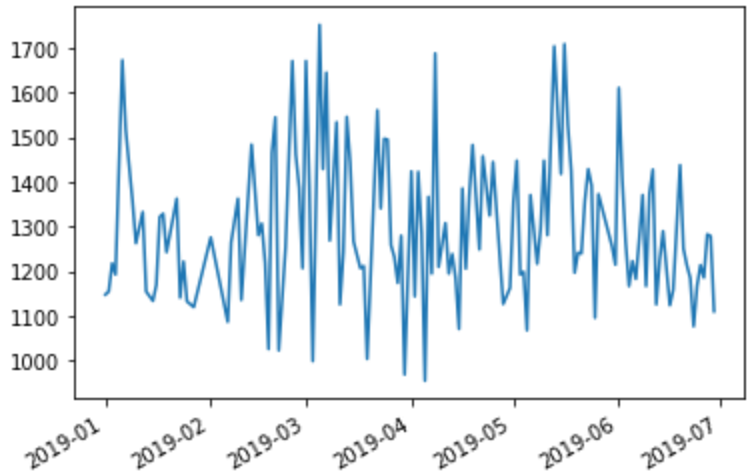
One of the biggest struggles was imputing missing values of the data. One of the most important columns in bart_hotspots.csv was the Daily Mean Travel Time which had 58 missing values. In order to get a better picture of seasonality, we couldn't discard those data points. In addition, because the dataset was a time-series dataset, the normal methods of imputing data may not preserve seasonality and trends. So we tested Last Observation Carried Forward (LOCF), Next Observation Carried Backward (NOCB) and Interpolation. We ended up going with interpolation because LOCF and NOCB introduced bias and the time interval between each data point is 1 day, which works well when doing interpolation for time-series data.

To illustrate the imputation, here is an example:




This is the original traveling time from Powell station to Fisherman's Wharf. Because of lots of null values, we can see many breaking points in the graph. However, the ups and downs

clearly show a trend with respect to the time series. After we used interpolation, the imputed values should look like this:




In this project, we only imputed the Daily Mean Travel Time (Seconds) column, as this is the column that we used most frequently.

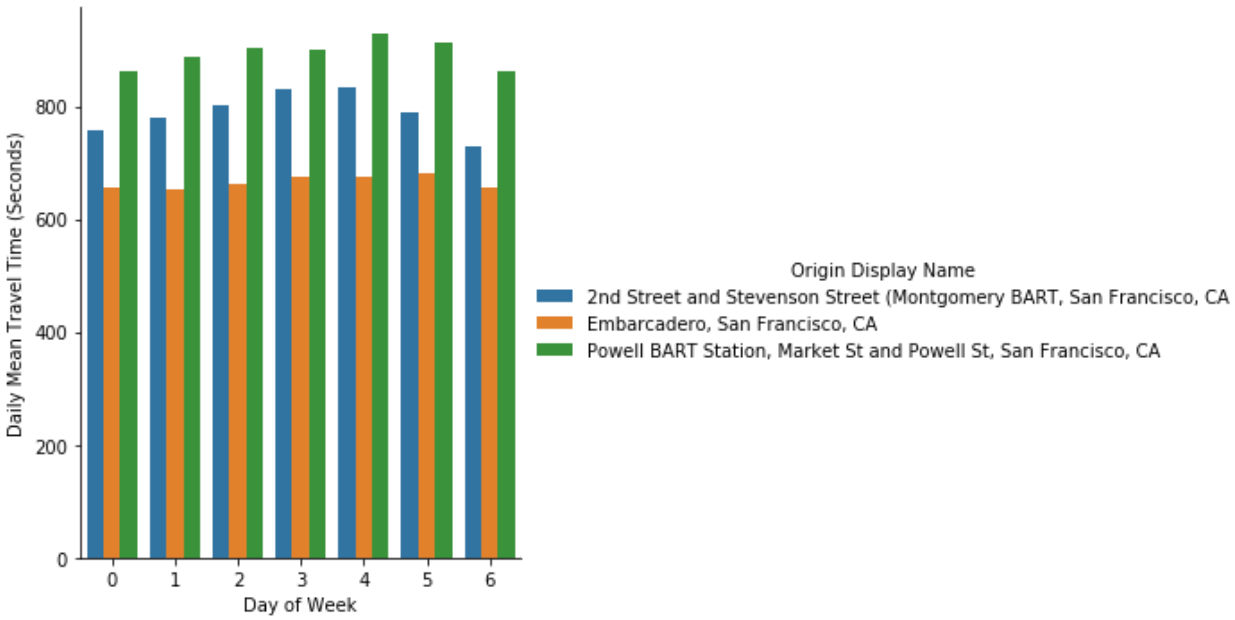
**Recommendation on how to decrease the amount of traveling time:**

Recommending the best BART stations to get off in order to go to specific hotspots is one of the most popular services amongst customers. Therefore, if Uber could provide this recommendation, it would largely increase the loyalty of its customers.

We extracted data points with origin from BART stations from the barts_hotspots dataset and gave each data point a week of day attribute. In order to know which BART station is the best to get off at to get to specific hotspots, we grouped the data according to the day of the week and calculated the average traveling time for each.

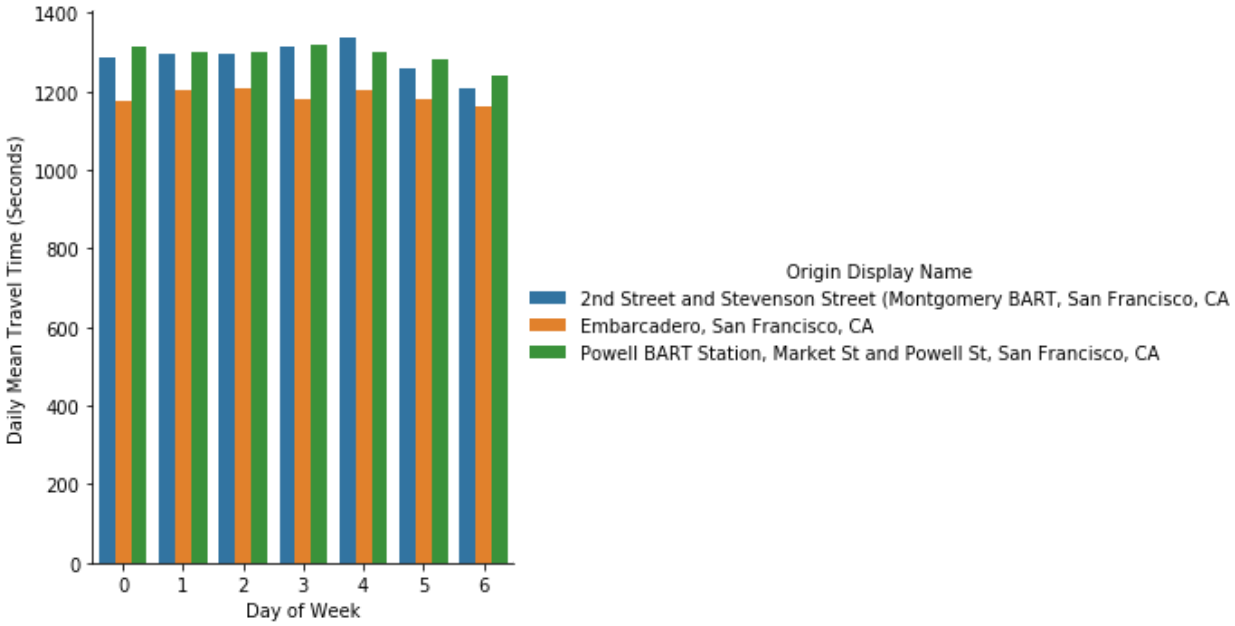
Here are the visualizations for traveling time. The x-axis shows the day of the week and the y-axis shows the travel time and different colors correlate to different BART stations.

(1)



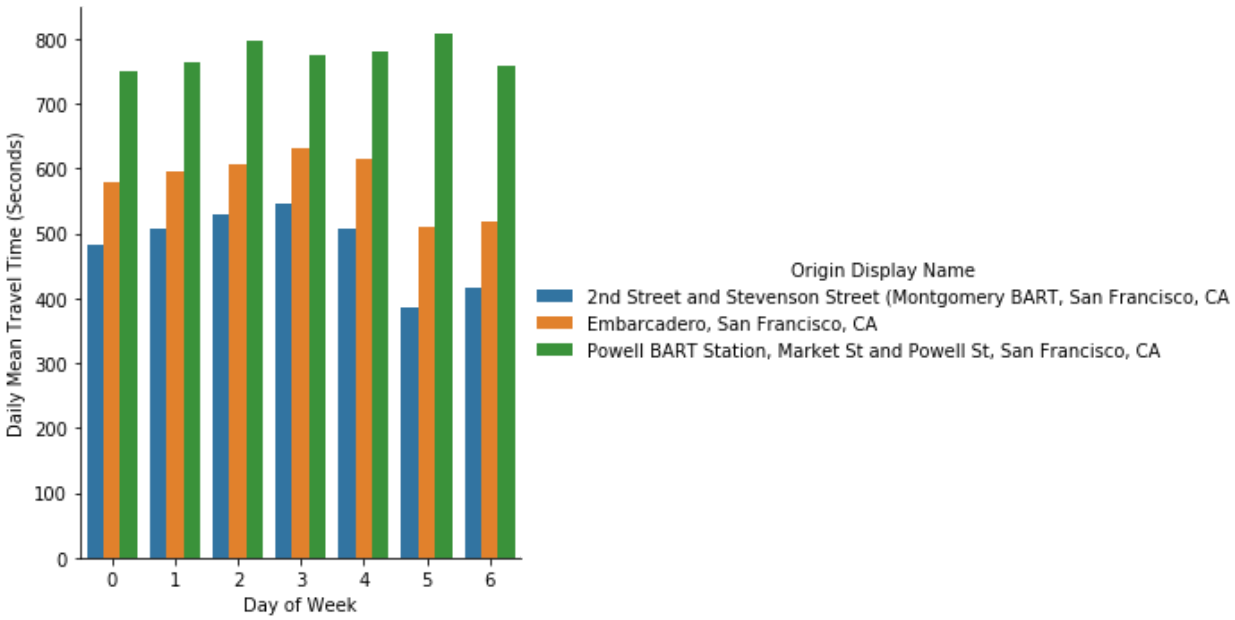

From the graph, we conclude that if we want to go to Fisherman's Wharf, the best BART station to get off at is Embarcadero for any day of the week.

(2)



Moreover, if we want to go to The Palace of Fine Arts, Embarcadero is also our best choice, as it will cost us less time.

(3)




On the other hand, if we want to go to Oracle Park, we'd better go from the Montgomery BART station.

Note: When we did splitting and grouping on the barts_hotspots datasets to get trips from one BART station to a hotspot on a specific day of the week, we essentially decreased the amount of data we can use. On average, for each mean we got, we only had about 25 data points. This limited the accuracy of our results.

**Average Travel Time Forecasting/ Machine Learning**

Forecasting the travel times from BART stations to hotspots and from hotspots to BART stations is important because these routes usually have heavy traffic, so predicting the travel time

of these routes can help determine how much fare should Uber charge riders. Furthermore, it can also help predict how much revenue Uber can earn from these popular routes each day, and estimate the profit from it.

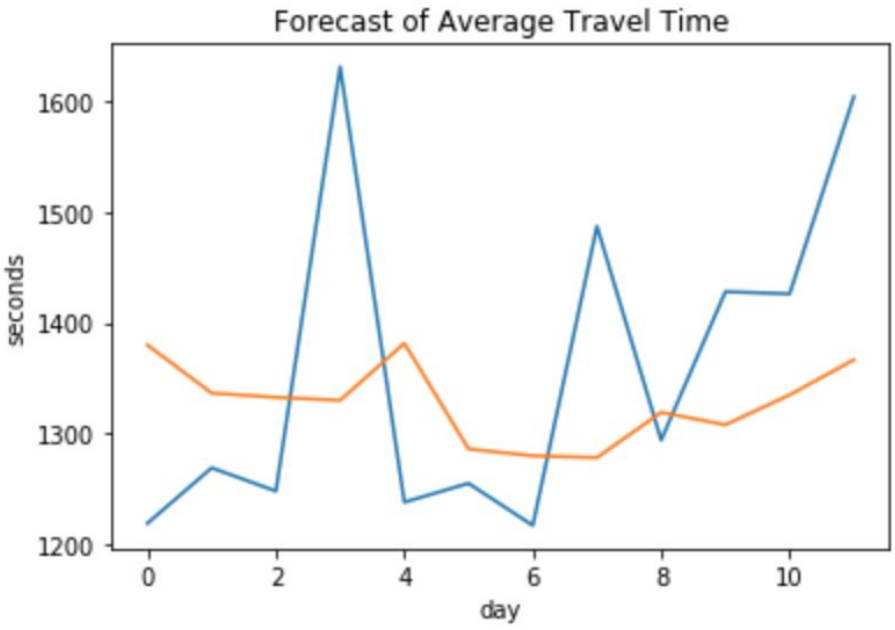
To forecast the average travel time, we built a model using Long Short-Term Memory Network (LSTM). The benefit of LSTM is that it can memorize previous data, and saves that data for later when predicting how future data points would behave. Therefore, LSTM is great for processing a sequence or a temporal series of data points, where the value of each data point depends heavily on previous data points. In this case, it would be perfect for forecasting future travel time of a certain route, since we know that past average travel time is a good predictor for future travel time.

Since many data entries for AM/PM and Morning/Midday/Evening average travel time are not available to us, we build the machine learning mainly based on the "Daily Mean Travel Time" column in the bart_hotspots dataset. From the previous data analysis, we know that there are 3 BART stations and 3 hotspots locations, so there are in total 18 different routes considering travel directions. For demonstration purposes, we only considered the route from The Palace Of Fine Arts to Embarcadero when building our model.

Before we built our model, we first transformed the time-series data into a supervised learning problem, where input and output are clearly defined. In this case, we used travel time from the day before as the input, and current day's travel time as output. Next, we transformed the data into stationary. This is done by differencing current travel time with previous travel time. Finally, we normalize the data for stable results by scaling all data into -1 to 1 range.

To train the model, we split the dataset into train and test data set with a ratio of 9:1, because we don't have a lot of data to process, and we want to make sure our model is trained based on sufficient amounts of data to ensure accuracy. The model is built using Keras library, which contains two layers, an LSTM layer and a density connected layer, the percentage error is calculated through mean square error. After training, our test result for the forecast is as following the image, where the blue line indicates real values, and yellow line indicates prediction values:

```
Test RMSE: 151.957
```




The mean square error shows that the model performed fairly poorly, with Root Mean Square Error of 152. From the graph we can also see that the prediction values roughly matches the real values. We believe this is because the we only have a little over 100 entries for this dataset, if we have more data, we believe the model would perform much better.