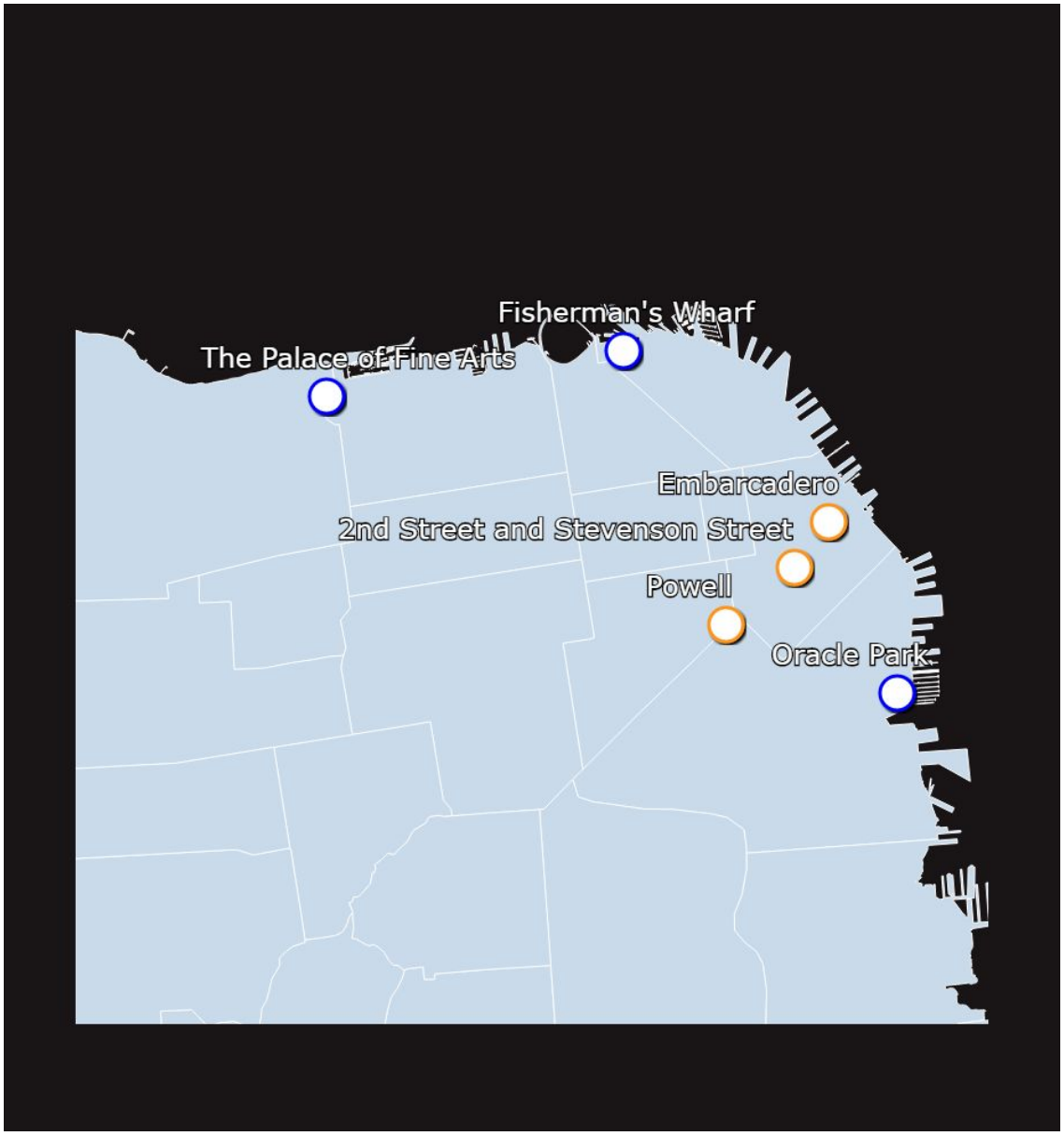
Data Cleaning/Preprocessing

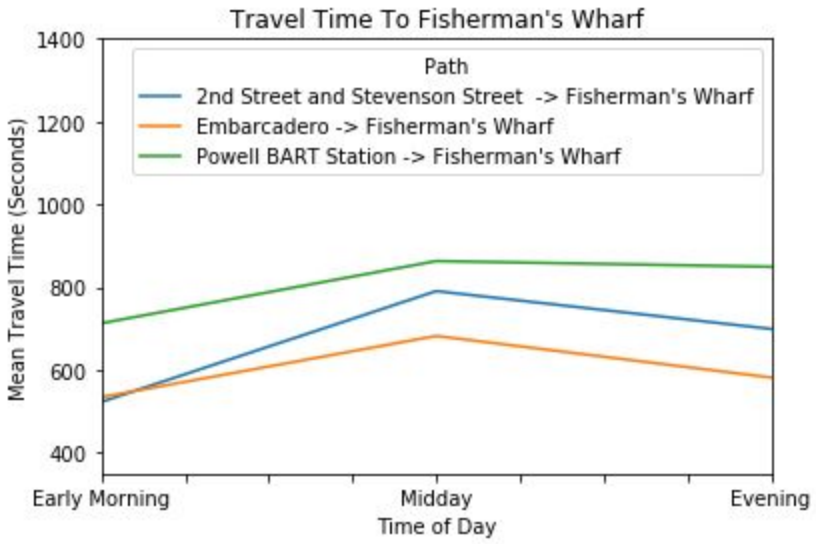
For the machine learning model, we only retained the paths that did not have a nan value. We also dropped AM, and PM as categories as the Early Morning, Midday, and Evening categories had a finer temporal granularity. We also converted the Date column into a Weekday column. One irregularity we found in the data was 2 duplicate columns for Fisherman's Wharf that had slightly differently written addresses. The values were similar enough, so we decided to drop one of them from the visualization.

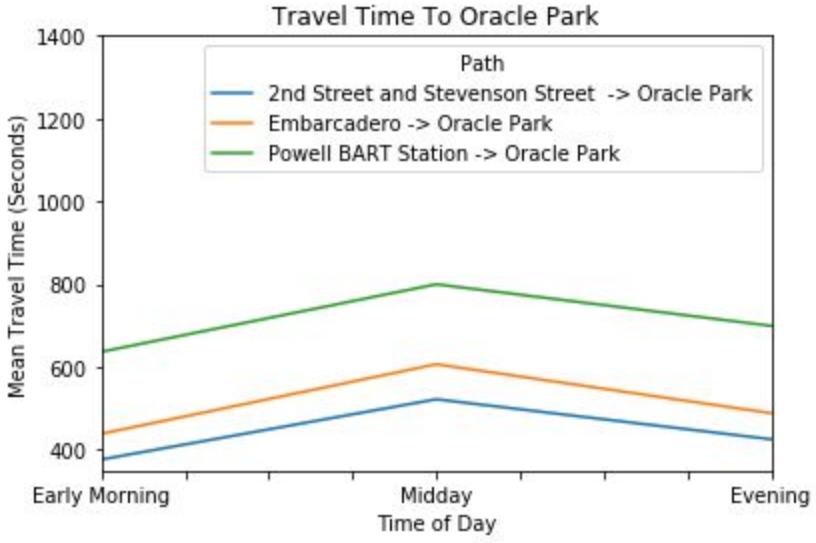
Data Visualization

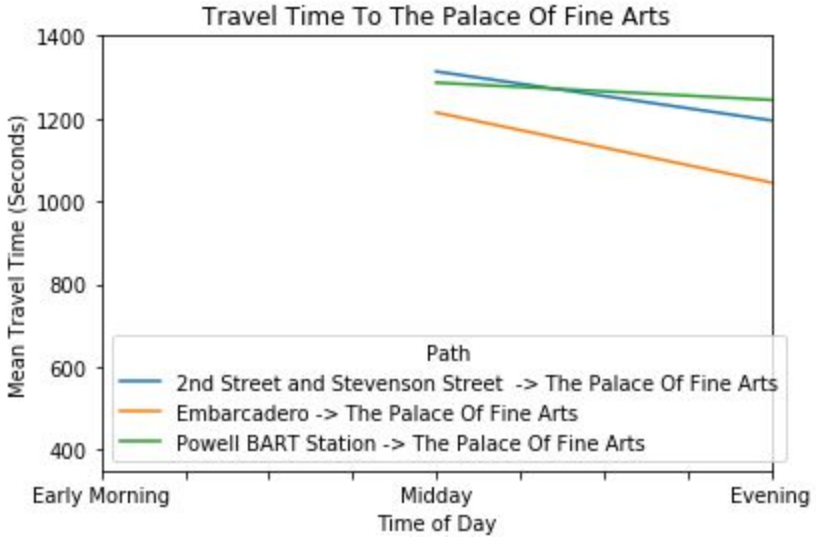


Map created using d3 to render geoJSON.

Average Duration of Travel by Time of Day


Travel Time To Fisherman's Wharf
Path
2nd Street and Stevenson Street -> Fisherman's Wharf
Embarcadero -> Fisherman's Wharf
Powell BART Station -> Fisherman's Wharf
Mean Travel Time (Seconds)
1400
1200
1000
800
600
400
Early Morning
Midday
Evening
Time of Day

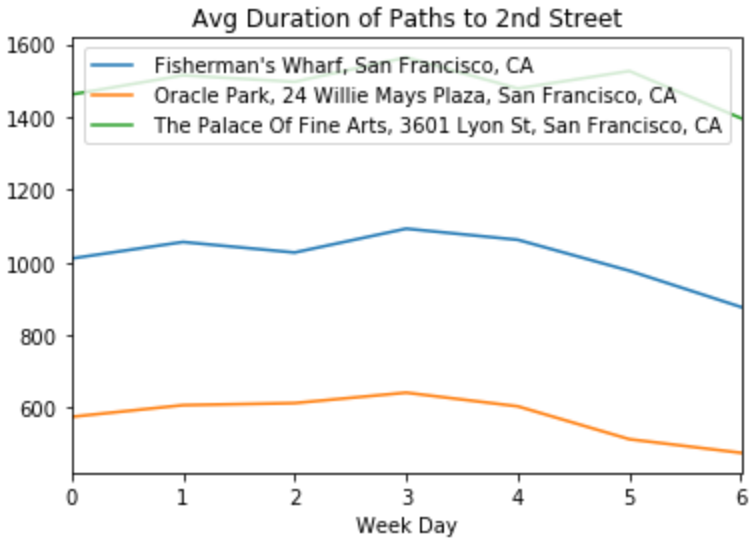

Travel Time To Oracle Park
Path
2nd Street and Stevenson Street -> Oracle Park
Embarcadero -> Oracle Park
Powell BART Station -> Oracle Park
Mean Travel Time (Seconds)
1400
1200
1000
800
600
400
Early Morning
Midday
Evening
Time of Day

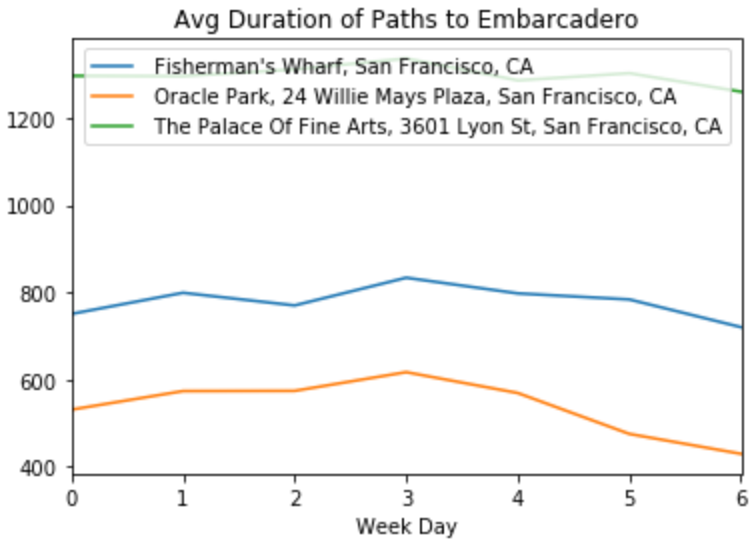

Travel Time To The Palace Of Fine Arts
Path
2nd Street and Stevenson Street -> The Palace Of Fine Arts
Embarcadero -> The Palace Of Fine Arts
Powell BART Station -> The Palace Of Fine Arts
Mean Travel Time (Seconds)
1400
1200
1000
800
600
400
Early Morning
Midday
Evening
Time of Day

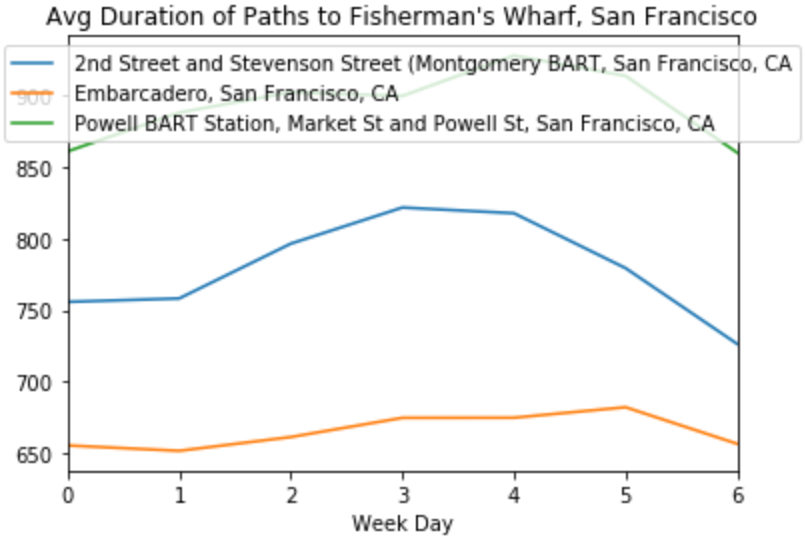
These plots visualize average travel time to each hotspot destination. They simply and effectively communicate that Early Morning and Evening are the best times of day for travel, and tell us which Bart stations provide the quickest path on average.

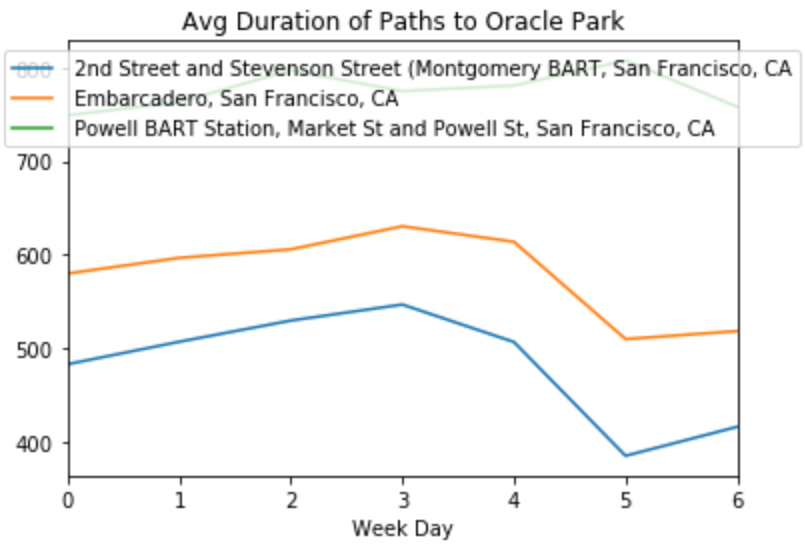
Interesting things to note: looking at the map, it may seem the shortest path is always the best, but an example where it's not is shown in the duration of paths to the Palace of Fine Arts, where Embarcadero, which looks to be the farthest away, has the shortest travel time to the hotspot.

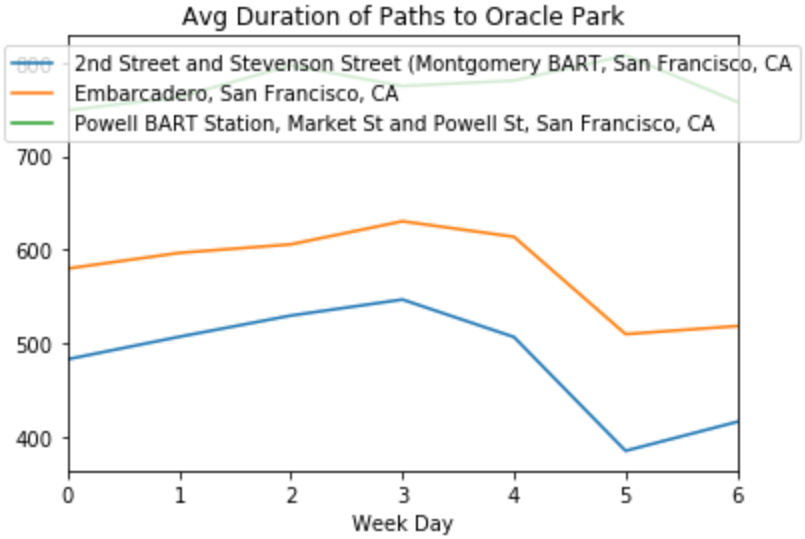
Average Duration of Travel by Week Day

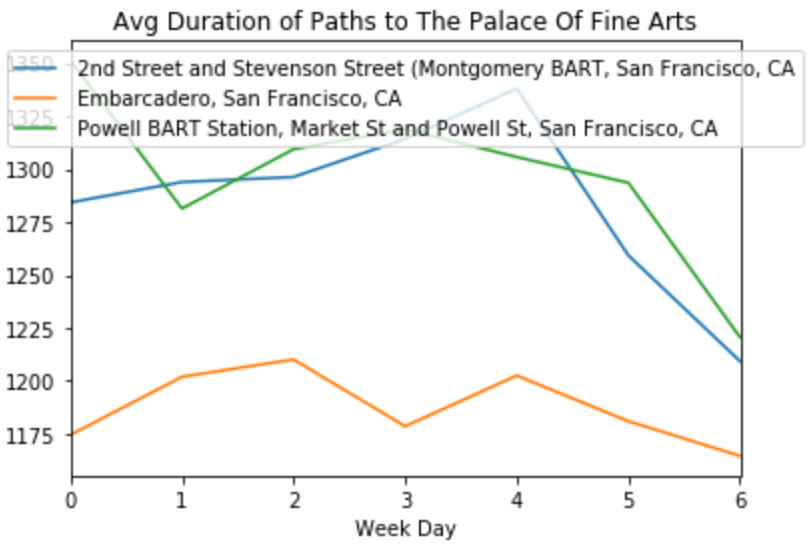



Avg Duration of Paths to Embarcadero
Fisherman's Wharf, San Francisco, CA
Oracle Park, 24 Willie Mays Plaza, San Francisco, CA
The Palace Of Fine Arts, 3601 Lyon St, San Francisco, CA
1200
1000
800
600
400
0
1
2
3
4
5
6
Week Day


Avg Duration of Paths to Fisherman's Wharf, San Francisco
2nd Street and Stevenson Street (Montgomery BART, San Francisco, CA
Embarcadero, San Francisco, CA
Powell BART Station, Market St and Powell St, San Francisco, CA
900
850
800
750
700
650
0
1
2
3
4
5
6
Week Day


Avg Duration of Paths to Oracle Park
2nd Street and Stevenson Street (Montgomery BART, San Francisco, CA
Embarcadero, San Francisco, CA
Powell BART Station, Market St and Powell St, San Francisco, CA
700
600
500
400
0
1
2
3
4
5
6
Week Day


Avg Duration of Paths to Oracle Park
2nd Street and Stevenson Street (Montgomery BART, San Francisco, CA
Embarcadero, San Francisco, CA
Powell BART Station, Market St and Powell St, San Francisco, CA
700
600
500
400
0
1
2
3
4
5
6
Week Day



These visualizations show us that the best paths to hotspots stay consistent throughout the week. The best times to travel are during the weekend (Friday, Saturday, Sunday).

To produce the summary travel time graphs, we used pandas' built in plot functions after restructuring the data into pivot tables.

Forecasting/Machine Learning

The first step in our pipeline was converting the proper feature columns in our cleaned dataset to binarized categorical values using OneHotEncoding. We then trained the model using the linear model, Lasso, on training sample of 0.33 of the original

dataset. Lasso is a regression technique optimized for predicting, and was better than more complex options because there were few samples. Since Lasso regularizes the coefficients, it allows the models created to better generalize to a new dataset. One thing to note is that between quarters, there was a 0.06% increase in mean travel duration, a 1.03% increase in the mean lower bound of travel time, and a 0.94% decrease in the mean upper bound of travel time, however we have not yet tested the significance of these values. Including a feature that looks at the Time of Year could be helpful in determining the best time of year to travel, as well as to predict travel times during different seasons given more data.

In the following code, we split train and test into ⅓ train and ⅔ test. We could have included a validation set, but we felt the number of samples were too small to warrant splitting the data further. Although one option to consider is resampling the data with replacement in order to increase the number of samples. We decided to keep it simple however.

```
# test if model works

X = pd.get_dummies(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)

# Lasso

clf = linear_model.Lasso(alpha=0.1)
clf.fit(X_train, y_train)

print("Score: " + str(clf.score(X_test, y_test)))
print("RMSE: " + str(np.mean((clf.predict(X_test) - y_test)**2)))
```

```
Score: 0.9331117707456638
RMSE: 4489.519142439949
```

The score is the coefficient of determination, or R-squared. 1.0 is the best possible value.

We then created a function that predicts the average amount of time in seconds that a particular path would take, given: a time of day, origin bart station, destination bart station, day of the week.

For example, calling

```
get_prediction('Early Morning', "Embarcadero, San Francisco, CA",
               "The Palace Of Fine Arts, 3601 Lyon St, San Francisco, CA",
               6)
```

Would return

```
Predicting travel time for
{
        time_of_day:    Early Morning,
        path:           Embarcadero -> The Palace Of Fine Arts,
        day_of_week:    6
}

1010.486098937441
```

Resulting in a prediction of about 1010 seconds for traveling from Embarcadero to The Palace of Fine Arts in the early morning on Saturday. One important thing to note from this model is that it can predict the average amount of seconds the travel would take for paths that are missing data (Average travel times to The Palace of Fine Arts in the Early Morning).

Conclusion

Uber can benefit from these findings by implementing a travel time recommender

or reward system for users that reserve transport vehicles ahead of time. By recommending users to travel during times of low traffic, overall travel time for Uber customers and regular commuters would decrease.