Capstone Project 1 - Milestone Report

**Problem:**

The aim of this Capstone 1 project is to predict the duration of taxi trips in New York City. Accurate estimates of taxi trip durations can improve the taxi utilization and the satisfaction of drivers and passengers. If a taxi dispatching system knew approximately when a taxi driver would be ending their current ride, it would help identifying which drivers should be assigned to which pickup locations. Additionally, the predictive model could be used for finding optimal routes for different kinds of trips.

The data has been taken from Kaggle which provides a starting point dataset consisting of the records of ~1.5 million taxi trips that took place in 2016. I’ve tried to go through the process of understanding the individual variables in the data by presenting beautiful, clear, and interactive data visualizations along with some approaches to their interpretation.

**Client:**

The prospective client would be any one of the taxi vendors operating rides in New York City.

**Data Set:**

**Test dataset:**

|  |  |
| --- | --- |
| **Variable name** | **Variable description** |
| id | A unique identifier for each trip |
| vendor\_id | A code indicating the provider associated with the trip record |
| pickup\_datetime | Date and time when the meter was engaged |
| passenger\_count | The number of passengers in the vehicle (driver entered value) |
| pickup\_longitude | The longitude where the meter was engaged |
| pickup\_latitude | The latitude where the meter was engaged |
| dropoff\_longitude | The longitude where the meter was disengaged |
| dropoff\_latitude | The latitude where the meter was disengaged |
| store\_and\_fwd\_flag | This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server. |
| Y=store and forward; N=not a store and forward trip |

**Train dataset:**

|  |  |
| --- | --- |
| **Variable name** | **Variable description** |
| id | A unique identifier for each trip |
| vendor\_id | A code indicating the provider associated with the trip record |
| pickup\_datetime | Date and time when the meter was engaged |
| dropoff\_datetime | Date and time when the meter was disengaged |
| passenger\_count | The number of passengers in the vehicle (driver entered value) |
| pickup\_longitude | The longitude where the meter was engaged |
| pickup\_latitude | The latitude where the meter was engaged |
| dropoff\_longitude | The longitude where the meter was disengaged |
| dropoff\_latitude | The latitude where the meter was disengaged |
| store\_and\_fwd\_flag | This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server. |
| Y=store and forward; N=not a store and forward trip |
| trip\_duration | Duration of the trip in seconds |

**Data Wrangling / Cleaning:**

* **Cleaning up trip duration.**

As we noted earlier there are some outliers associated with the trip\_duration variable, specifically a 980 hour maximum trip duration and a minimum of 1 second trip duration. I've decided to exclude data that lies outside 2 standard deviations from the mean. It might be worthwhile looking into what effect excluding up to 4 standard deviations would have on the end-results.

*m = np.mean(train['trip\_duration'])*

*s = np.std(train['trip\_duration'])*

*train = train[train['trip\_duration'] <= m + 2\*s]*

*train = train[train['trip\_duration'] >= m - 2\*s]*

* **Latitude and Longitude cleanup**

Looking into it, the borders of NY City, in coordinates comes out to be:

city\_long\_border = (-74.03, -73.75) city\_lat\_border = (40.63, 40.85)

Comparing this to our train.describe() output we see that there are some coordinate points (pick ups/drop offs) that fall outside these borders. So I cleaned up the data to limit our area of investigation to within the NY City borders.

*train = train[train['pickup\_longitude'] <= -73.75]*

*train = train[train['pickup\_longitude'] >= -74.03]*

*train = train[train['pickup\_latitude'] <= 40.85]*

*train = train[train['pickup\_latitude'] >= 40.63]*

*train = train[train['dropoff\_longitude'] <= -73.75]*

*train = train[train['dropoff\_longitude'] >= -74.03]*

*train = train[train['dropoff\_latitude'] <= 40.85]*

*train = train[train['dropoff\_latitude'] >= 40.63]*

* **Formatting datetime for better and easy extraction of these fields.**

*train['pickup\_datetime'] = pd.to\_datetime(train.pickup\_datetime)*

*train.loc[:, 'pick\_month'] = train['pickup\_datetime'].dt.month*

*train.loc[:, 'hour'] = train['pickup\_datetime'].dt.hour*

*train.loc[:, 'week\_of\_year'] = train['pickup\_datetime'].dt.weekofyear*

*train.loc[:, 'day\_of\_year'] = train['pickup\_datetime'].dt.dayofyear*

*train.loc[:, 'day\_of\_week'] = train['pickup\_datetime'].dt.dayofweek*

**Other potential data sets:**

There are other data sets that can be used to have additional features added to the existing dataset. This has been taken from <http://project-osrm.org/>

**Accidents\_2016.csv**

**Fastest\_roustes\_test.csv**

**Fastest\_routes\_train\_part1.csv**

**Fastest\_routes\_train\_part2.csv**

**Second\_fastest\_route\_test.csv**

**Second\_fastest\_route\_train.csv**

All of the above data have id field that can be used to merge with existing train/test data set. Some of those features include **`dropoff\_datetime`, `avg\_speed\_m`, `avg\_speed\_h`, `pickup\_lat\_bin`, and `pickup\_long\_bin`.**

**Initial findings:**

* ~70 % of the time people travel alone. The most popular way of sharing a taxi is to share it with a single friend. The second most popular way of sharing a taxi is to travel in a group of five people.
* There are two vendors in the data, with the second vendor representing the majority.
* Based on the trip duration in train data, vendor 1 is taking more time than vendor 2 on all days of the week.
* Trip duration graph on hourly basis for all days of the week shows that the trip duration is very less on Saturday and Sunday than all the weekdays between 5 AM - 3 PM.
* The highest number of pickups are around 6pm and 7pm in the evening which makes sense as many people are on their way to home from office.