# Problem & Context

Taxi trip durations in NYC can vary wildly because of several factors from weather and traffic to simply geography of the pickup and drop-off locations. For instance, New York is abound with one-ways, side streets, and known for the ever-present number of pedestrians. With everyone engaging in a mad rush to get from point A to point B, you'll invariably find yourself late for whatever you need to be on time for. The goal of this analysis is to determine which factors predict the taxi trip's duration.

# Stakeholders & Audience

The main beneficiary will be the Taxi & Limousine Commission of NYC. They’ll better be able to control costs and maintain a level of accuracy relating to ‘appropriate’ travel times for taxi drivers. Another group that will benefit are passengers that will be better able to know how long trips will take, ahead of time.

# Data

The main data stems from the [2016 NYC Yellow Cab](https://cloud.google.com/bigquery/public-data/nyc-tlc-trips) trip record data from Big Query on the Google Cloud Platform.

Two main datasets will be used for this project:

* Taxi\_Train
* Taxi\_Test

**Taxi\_Train**

Each row corresponds to a specific taxi trip with features detailing pickup & drop off times, passenger count and other variables. It consists of 1,458,644 observations and 11 variables, provided via [Kaggle](https://www.kaggle.com/c/nyc-taxi-trip-duration/data?select=train.zip). The target variable is named 'trip\_duration' and it indicates the length of a taxi trip. Data is 81 MB compressed and available as a single zip file.

**Taxi\_Test**

Similar to the Taxi\_Train dataset, each row is associated with a trip. This dataset consists of 625,134 observations and 10 variables. Data is 21 MB compressed and available as a single zip file from [Kaggle](https://www.kaggle.com/c/nyc-taxi-trip-duration/data?select=test.zip). We will be predicting the ‘trip\_duration’ variable for this dataset.

Data Description:

* In the train and test data, features that belong to similar groupings are tagged as such in the feature names (e.g., ind, reg, car, calc).
* Feature names include the suffix ‘bin’ to indicate binary features and ‘cat’ to indicate categorical features.
  + Features without these designations are either continuous or ordinal.
* Values of -1 indicate that the feature was missing from the observation.

# Data Wrangling & Preparation

## Initial Investigation

**Taxi\_Train Data Fields Info**

* *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

A **few** things seem off initially:

* *passenger\_count*:
  + there should probably only be a minimum of 1, unless there're some erroneous trips in the dataset
* *trip\_duration*:
  + the minimum looks like 1, which is definitely not a trip worth analyzing
  + the maximum is 3,526,282 seconds which a back of the envelop calculations means it was a 979.52-hour trip!

Some outliers need to be cleaned up here, in a bit.

## Data Preparation

Here we'll address some of the issues with the data and get it ready for analysis.

* **First** on the docket is formatting the *pickup\_datetime* & *dropoff\_datetime* so that we can summarize it based on days of the week or across months to determine any seasonality.
* **Next** I'll get the trip duration to address the outliers, by excluding information that's 2+ standard deviations away.
* **Also** the latitudes and longitudes need to be verified that they're within NYC's parameters. Specifically, according to NYC's [Open Data Portal](https://data.cityofnewyork.us/City-Government/Borough-Boundaries/tqmj-j8zm):
* longitudinal boundaries of NYC are between -74.03 and -73.75
* latitudinal boundaries of NYC are between 40.63 and 40.85

# Exploratory Data Analysis

## Trip Duration

Chart, histogram

Description automatically generated

It looks like the majority of the trips are between 600 & 1,000 seconds, or about 10 to 15 minutes. There’s a lot of variability so we’ll implement a logarithmic scale to account for the plethora of values.

Chart, histogram

Description automatically generated

## Pickup Dates & Seasonality

Chart, scatter chart

Description automatically generated

Thankfully, the taxi\_train & taxi\_test datasets have a similar trend/shape.

Looks like there's quite the dip in trips in late **January**, with another modest dip in late **May**. My initial guesses are around data errors versus anything season-related (winter in January or Spring in May). Outliers!

## Vendors

Chart, bar chart

Description automatically generated

It looks like there **isn't much of a difference** in the mean trip durations between vendors. However, let's see if there's something else that can help tease apart trip durations.

Chart, bar chart

Description automatically generated

There looks to be **some differentiation** between the Store & Forward Flag and mean Trip Duration! This may indicate inaccuracies in how the trips were recorded given the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server.

## Passenger Counts

Let's look at the number of passengers as this may mean that Trip Duration may be longer for more passengers as they'll need to be dropped off in multiple locations. To test that theory, let's see how passengers looks when bumped up against trip duration.

Chart, bar chart

Description automatically generated

it looks like there's **some consistency/no difference** across the number of passengers (>= 1) in terms of mean travel time. There's might be a data issue relating to the '0' passenger count, but I'll investigate if the test data has the same issue.

## Pickup Locations

Here I’m comparing whether taxi\_train & taxi\_test have any significant differences in their distribution of pickup locations.

Diagram, engineering drawing

Description automatically generated

A picture containing chart

Description automatically generated

It looks like they're **near identical** regarding the pickup locations!

## Dates

Currently, the *pickup\_datetime* and *dropoff\_datetime* aren't granular enough to work with, and definitely not ready for modeling in its categorical state. After conversion, let’s explore the mean trip durations by dates.

Chart, bar chart

Description automatically generated

Nothing really stands out here, across the first 6 months of the year. Let’s look at day of the momth.

Chart, background pattern

Description automatically generated

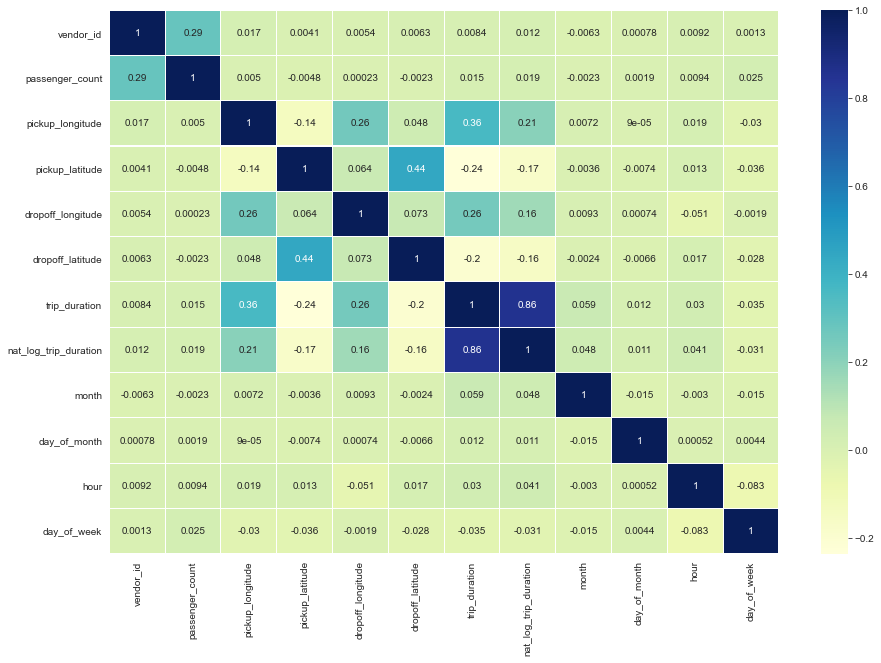
Again, pretty consistent. Now let’s look at even more granularity with hour of the day.

Chart, bar chart

Description automatically generated

It's pretty interesting **how much consistency** there is in mean *trip\_duration* amongst all the date variables, **except for hour of the day**.

It makes sense that *trip\_*duration during typical working hours of 8a - 5p are longer. The assumption is that there're more people on the road, commuting and generally creating traffic.



I'm not seeing anything that has a strong correlation with *trip\_duration*, but there's a little indication between *pickup\_longitude/latitude* & *dropoff\_longitude/latitude*. This, of course, makes sense that a closer drop off location means a shorter trip, in general. Otherwise, nothing really gained from the heatmap.

# Feature Engineering & Dummy Coding

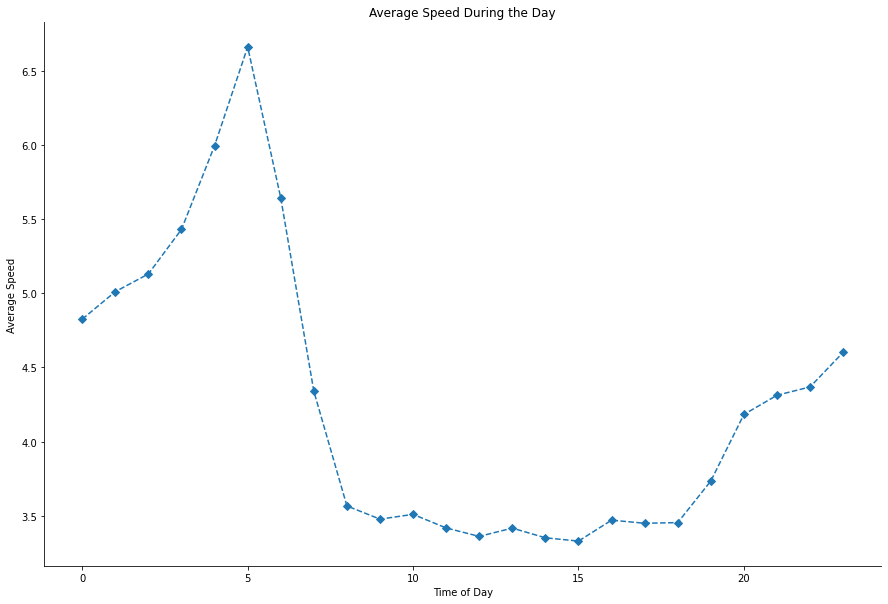
One potential avenue of analysis will be to look at the speed the particular taxi is travelling at while traversing NYC. In order to calculate speed we need to determine distance in order to compare the time it takes to traverse that distance.

Thankfully, in the dataset we have longitudinal and latitudinal data. Using information from a pre-existing [Kernel](https://www.kaggle.com/gaborfodor/from-eda-to-the-top-lb-0-367) on Kaggle, the following functions calculate distance of a specific trip based on the pickup and drop off latitudinal & longitudinal data.

Three functions created to calculate distance:

* **haversine\_array**: Haversine distance is great-circle distance between two points on a sphere using latitude & longitude
* **dummy\_manhattan\_distance**: Calculates the summed distance traveled in Manhattan
* **bearing\_array**: Calculates the bearing of the distance traveled.

With the distance figured out, we can start looking at how to calculate speed (distance / time). Here, we'll zero-in on the hour of the day, the day of the week, and the month compared against average speed.



Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

**Initial Findings**

* During the day: typical work hours of 8a - 5p have the lowest average speed of taxi trip
* During the week: Monday - Thursday have the lowest average speed of taxi trip
* During the year: warmer months have lower average speed

This makes sense given the weather can heavily factor into how much time one is outdoors, where the colder, earlier hours of the day have less people, and in turn less traffic. Less traffic, intuitively, means there's more of an opportunity to travel faster, speed-wise.

## Dummy Coding

Here we want to be able to handle the object/non-numeric variables that can trip up our model.

taxi\_train:

* vendor\_train
* month\_train
* passenger\_count\_train
* dom\_train (day of month)
* store\_and\_fwd\_flag\_train
* hour\_train
* dow\_train (day of week)

taxi\_test:

* vendor\_test
* month\_test
* passenger\_count\_test
* dom\_test
* store\_and\_fwd\_flag\_test
* hour\_test
* dow\_test

The *taxi\_train* dataset should have one more column than the *taxi\_test* dataset because it contains the target variable, *nat\_log\_trip\_duration*, which the test dataset shouldn't.

# Approach

Given the built-in time & geographic nature of the features, the first order of business is to determine which features, or groups of features, have the most impact (correlation) with the trip duration (target). In order to take this on, a significant number of visualizations/plots will be employed during the Exploratory Data Analysis portion of this capstone. In particular, an alluvial plot to determine multi-feature interactions.

Once there’s been sufficient exploration of the connections within the current features, I’ll use these insights to build new features. I will be using XGBoost to be able to predict how long the taxi trips will last.

# Modeling

Here I'll be splitting *taxi\_train* into a sub-training and sub-testing set. This is so I can tweak model parameters to increase accuracy (e.g. decrease root mean square error [RSME] value) without creating a situation of overfitting on the actual *taxi\_test* data.

I need a validation set to use with XGBoost algorithm. It'll take three datasets: a training set, a test set, and a validation set. The validation dataset evaluates the accuracy of the training model. The training model, of course, is used to make predictions against the test dataset.

Splitting out the training set into a separate train and test datasets gives us a test sample where we know the outcome variables!

Using the following parameters, I achieved RMSE of 0.49373 by running the model 10 times.

* 'min\_child\_weight': 1
* 'eta': 0.5
* 'colsample\_bytree': 0.9
* 'max\_depth': 6
* 'subsample': 0.9
* 'lambda': 1
* 'nthread': -1
* 'booster' : 'gbtree'
* 'silent': 1
* 'eval\_metric': 'rmse'
* 'objective': 'reg:squarederror'

The RMSE is below 0.5! Based on a rule of thumb, it can be said that RMSE values between 0.2 and 0.5 show that the model can relatively predict the data accurately.

Let’s plot to determine what feature influences *nat\_log\_trip\_duration* the most:

A picture containing chart

Description automatically generated

From the top-down, we're seeing which features have the greatest effect on trip duration. It makes sense that (haversine) distance has the greatest effect since the further you're going, the longer it'll take to get there.

# Recommendation

Between the hours of 8a & 5p, prioritizing nearby drop off locations, will result in shorter trips and better trip duration times. It would behoove the taxi & limousine company to determine drop off location ahead of time via either an app where users enter their destination or via a dispatcher that’s organizing a driver’s pickup spots.

# Future Analyses

There’s a machine learning algorithm called [Open Source Routing Machine](http://project-osrm.org/) (OSRM) that’s determined the best route between two points. There’s already a Kaggle [Kernel](https://www.kaggle.com/oscarleo/new-york-city-taxi-with-osrm) with these routes determined for this dataset.

There’re 2 reasons why I’d look at this for future analyses:

* Would incorporating that kind of ML knowledge into how a driver gets around the city impact the mean trip duration?
* If a ‘good’ route had been chosen during the typical rush hour, would it impact trip duration?