Machine Learning Engineer Nanodegree

Capstone Proposal

Samit Kumar Singh
September 21, 2017

Proposal

Domain Background

The project aims to solve a problem from Kaggle Competition: New York City Taxi Trip Duration. (https://www.kaggle.com/c/nyc-taxi-trip-duration)

Here I will try to build a model that predicts the total ride duration of taxi trips in New York City. An algorithm that can accurately predict the total ride duration will help passengers to better plan their schedule and reach their destination on time. Such an algorithm can also be used by ride hailing services like Uber, Lyft etc. in efficiently utilizing their resources and ultimately providing better service to the passengers. For example knowing when a taxi driver would be ending their current ride, would help these services to identify which driver to assign to each pickup request.

Problem Statement

This is a regression problem which tries to predict the total ride duration of taxi trips in New York City based on pickup time, geo-coordinates, number of passengers, and several other variables.

Datasets and Inputs

Here I am using the train and test data provided by the Kaggle Competition. The train data has the following variables which I think directly relates to the problem:

- Pickup date and time: On some dates the traffic will be more than usual and will affect the ride duration. Also time of travel will can also affect the ride duration of trips.

- Pickup and drop-off location: Ride duration directly correlates to the distance to be travelled. Based on pickup and drop-off coordinates given in the dataset, areas (neighborhoods) can be identified in New York City. Some areas have may have narrow and crowded roads, other may have wide highways. All this will affect the total ride duration.
- Other variable in the train data include id, vendor_id, passenger_count.

  The training set contains 1458644 trip records.
  The testing set contains 625134 trip records.
  I will be using all of training and testing set trip records.

  The dataset can be accessed from here : https://www.kaggle.com/c/nyc-taxi-trip-duration/data

Solution Statement

Based on the training data, using different learning algorithms like random forest, SVM, XGboost etc. for regression I would try to minimize the training loss. I plan to use feature engineering to find out the most insightful features, also if possible related dataset from other public sources that can provide some predictive power of the model can be used. Finally the model will be tested on the test data set that is provide for the competition.

Benchmark Model

A good benchmark model would be one that is solely based on the distance of the trip divided by average speed of taxis in city like New York (not taking into account other variables like traffic, road conditions, location etc.) Here we will assume an average speed of 11.5 mph. (source: http://animalnewyork.com/2014/data-reveals-worst-times-traffic-nyc)

Evaluation Metrics

The evaluation metric for this competition is Root Mean Squared Logarithmic Error.

The RMSLE is calculated as

Where:

$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1) - \log(a_i+1))^2}$$

ϵϵ is the RMSLE value (score)
nn is the total number of observations in the (public/private) data set,
pi is your prediction of trip duration, and
ai is the actual trip duration for ii.
log(x) is the natural logarithm of x

Project Design

I intend to follow the following workflow.

- Exploratory data analysis – Analyzing the given data set, doing feature engineering and coming up with new more relevant features.
- Acquiring additional data – If possible I would explore additional data from public sources which may help the model in better predicting the ride duration.
- Using different regression algorithms – Next step would be to implement and test different algorithms and select the one which will give the best result based on the evaluation metrics.
- Comparing results with benchmark model – The results will then be compared with the benchmark model to see if our model is any good. If not, the process above will be reiterated by coming up with better features.
- Finally, the last step would be to interpret the results.