Udacity's Machine Learning Engineer Nanodegreee Program
New York City Taxi Trip Duration
Capstone Proposal

Rodrigo S. Veiga

May 13, 2018

This proposal is based on the Kaggle's homonymous challenge.

1 Domain Background

The challenge is to build a model that predicts the total ride duration of taxi trips in New York City. The primary dataset is one released by the NYC Taxi and Limousine Commission, which includes pickup time, geo-coordinates, number of passengers, and several other variables.

2 Problem Statement

The competition dataset is based on the 2016 NYC Yellow Cab trip record data made available in Big Query on Google Cloud Platform. The data was originally published by the NYC Taxi and Limousine Commission (TLC). It was sampled and cleaned for the purposes of the playground competition. Based on individual trip attributes, we should predict the duration of each trip in the test set.

3 Datasets and Inputs

The data is composed by two data files:

- `train.csv` - the training set (contains 1458644 trip records)
- `test.csv` - the testing set (contains 625134 trip records)

There are eleven features in the training data set.

- `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

Based on individual trip attributes, we should predict the duration of each trip in the test set, which means the last feature, `trip_duration`, is the target variable.

Both training and test sets files are going to be converted into pandas data frames. Naturally, taking advantages of pandas, data pre processing and feature selection will be applied in order to improve performance.

4 Solution Statement

At least initially, ensamble regression methods will be applied with sckit-learn. The optimization will be made by the minimization of the root mean square logarithmic error described below using grid-search and cross-validation.

5 Evaluation Metrics

Following the challenge by Kaggle the models will be evaluated using the root mean square logarithmic error (RMSLE) ϵ, which is given by

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

where n is the total number of observations in the test set, p_i is the i-th prediction of trip duration, and a_i is the i-th actual trip duration.

The RMSLE is similar to square root of the mean square error, which is the simplest evaluation metric and it has a straightforward interpretation; it estimates the variance between the predicted instances and correct ones. The lower this variance, the better the prediction. Mean square logarithmic error is basically the same measure but in a logarithmic scale. It is interesting if one does not want to penalize too much huge differences between predictions and actual values.

6 Benchmark Model

The winner team, composed by Francesco Palma, Aldo Podestá, TomBoy and W.R. Lemes de Oliveira obtained the score 0.28976 on the private leaderboard, which is calculated with approximately 70% of the test data. Naturally, their solution is a benchmark model. However, since it would be quite ambitious to take such a value as goal and our purpose now is learning, not competing, we define our goal as reaching a score less than 0.5 on the private learderboard.

7 Project Design

It seems reasonable to treat this problem with ensamble regression methods, since many of the previous features will have to be split, resulting in a high dimensional data set.

Initially, data exploration and data preprocessing will be carried out before the implementation with sckit-learn. After training the proposed algorithms will be evaluated on test data using the evaluation metrics mentioned above.