Machine Learning Engineer Nanodegree

Capstone Proposal

Wenzhe(Emma) Ding

September 12th, 2017

Proposal

Domain Background

There are currently over 13,000 licensed taxicabs and over 50,000 taxicab drivers providing transportation for passengers in New York City via street hails(http://www.nyc.gov/html/tlc/html/industry/yellow_taxi.shtml). The New York City Taxi & Limousine Commission (TLC) has released data with detailed information of each taxi trip from January 2016 through July 2016. To understand taxi trip patterns and answer questions such as what is the rush hours of NYC taxi trips, what features are relevant to predict taxi trip duration, what are most popular taxi regions at different time of the day and etc. we need to investigate such data sets. In fact, many data scientists as well as researchers have done analysis using such public taxi data, some are listed below. * http://toddwschneider.com/posts/analyzing-1-1-billion-nyc-taxi-and-uber-trips-with-a-vengeance/ uses an older dataset to analyze travelling patterns * https://www.kaggle.com/headsortails/nyc-taxi-eda-update-the-fast-the-curious * https://www.kaggle.com/gaborfodor/from-eda-to-the-top-lb-0-367

Additionally, being a transportation engineering major, I am especially interested in studying people's travelling behavior by investigating public transportation data.

Problem Statement

The New York City Taxi & Limousine Commission hosts a kaggle competition (https://www.kaggle.com/c/nyc-taxi-trip-duration) to challenge data scientists to build a model that predicts the total ride duration of taxi trips in New York City, which also serves as the major goal of this project. Available features in the data include pickup time, geo-coordinates, number of passengers, and several others. Root Mean Squared Logarithmic Error is used to compare the performances among models.

Datasets and Inputs

Both the train data and test data are provided by the competition, which is also under the folder "data" in this repo and title as "train.csv" and "test.csv" respectively. The time range of both data set is from January 2016 to July 2016. The training set contains 1458644 trip records and the testing set contains 625134 trip records. Detailed explanation of data fields are shown below, which is also available at the "Data" section of the competition

(https://www.kaggle.com/c/nyc-taxi-trip-duration/data).

- 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

Basically, the train data would be split into train-validation set and we will use only train data for model training. Validation set will be used for parameter selection and to avoid overfitting of the model being built. Test data will be used to evaluation the performance of our model.

Solution Statement

One solution to this problem is gradient boosted trees regressor using XGBoost (a package in python http://xgboost.readthedocs.io/en/latest/), which is one of ensemble methods to deal with supervised learning problems. The main principle behind this method is that a group of “weak learners” can come together to form a “strong learner”. hey typically less prone to overfitting and make the model more robust,unlikely to be influenced by small changes in the training data. The input of this algorithm is a set of numerical features and the output is a number, which is also the prediction of the algorithm.

Benchmark Model

A benchmark model is decision tree regression model, which is widely used for supervised learning problems. It uses a tree like structure to specify a series of conditions that are tested to determine the value for a sample.

Evaluation Metrics

The evaluation metric for this project is Root Mean Squared Logarithmic Error (RMSLE), which is defined here (https://www.kaggle.com/wiki/RootMeanSquaredLogarithmicError). RMSLE penalizes an under-predicted estimate greater than an over-predicted estimate.

Project Design

Firstly, we will conduct comprehensive Exploratory Data Analysis (EDA) before building models for prediction as the insights can be both valuable for our model building as well as the community. A

few analysis of the data needs to be done in this process: * dealing with missing values if needed * remove outliers * extract useful features for model * feature engineering if needed Afterwards, we will implement algorithms which will includes split the training data into train-validation set, train XGBregressor, analyze feature importance, and score test set. Finally, we'll compared the performance of the best performed decision tree model and gradient boosted tree model.
