Taxi Duration Prediction Project

Zijing Gu, Jingwei Li, Wuwei Lin
SCS, ECE, INI @ Carnegie Mellon University

Introduction

Nowadays, ride-sharing apps are playing an increasingly important role in people's lives. It is especially important for customers and drivers to predict travel time in advance. Furthermore, the predicted travel time is also useful for city planning.

The goal of the project is to predict taxi travel time based on data of New York city taxis in 2017. Since the predicting is processed before the travel, only data which is available before the start is used. In the research, several methods and features are used to built the model completely.

Data Exploration

Dataset Summary The dataset we use in this project is approximately 60 million taxi rides recorded by New York City taxicabs in 2017. The starting and ending locations are pre-processed by discretizing the region and reporting the index of the starting and ending sub-regions. Thus we do not have the exact pickup/dropoff locations. The original dataset contains vendor ID, passenger counts, pickup and dropoff datetimes, pickup and dropoff region ID, and payment methods. The dataset contains $\sim$0.5% outliers, where duration is negative or larger than 8 hours. We dropped outliers in the dataset.

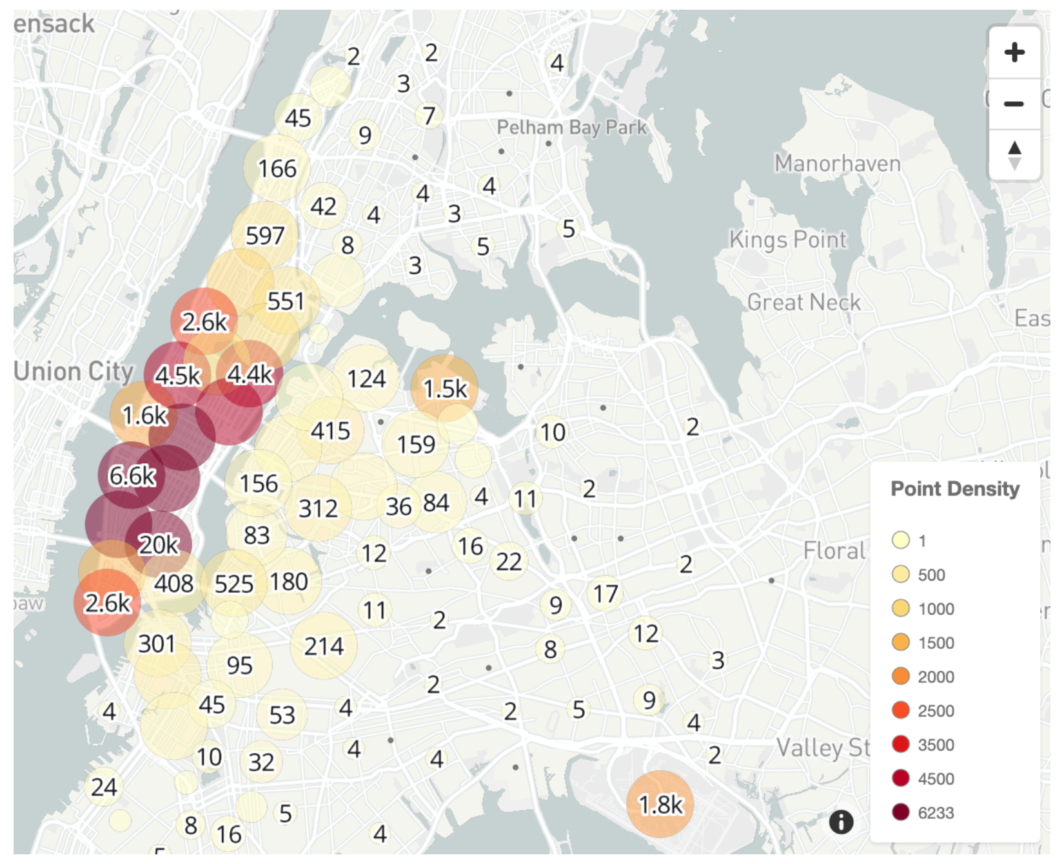

Figure 1: Pickup counts of all the regions

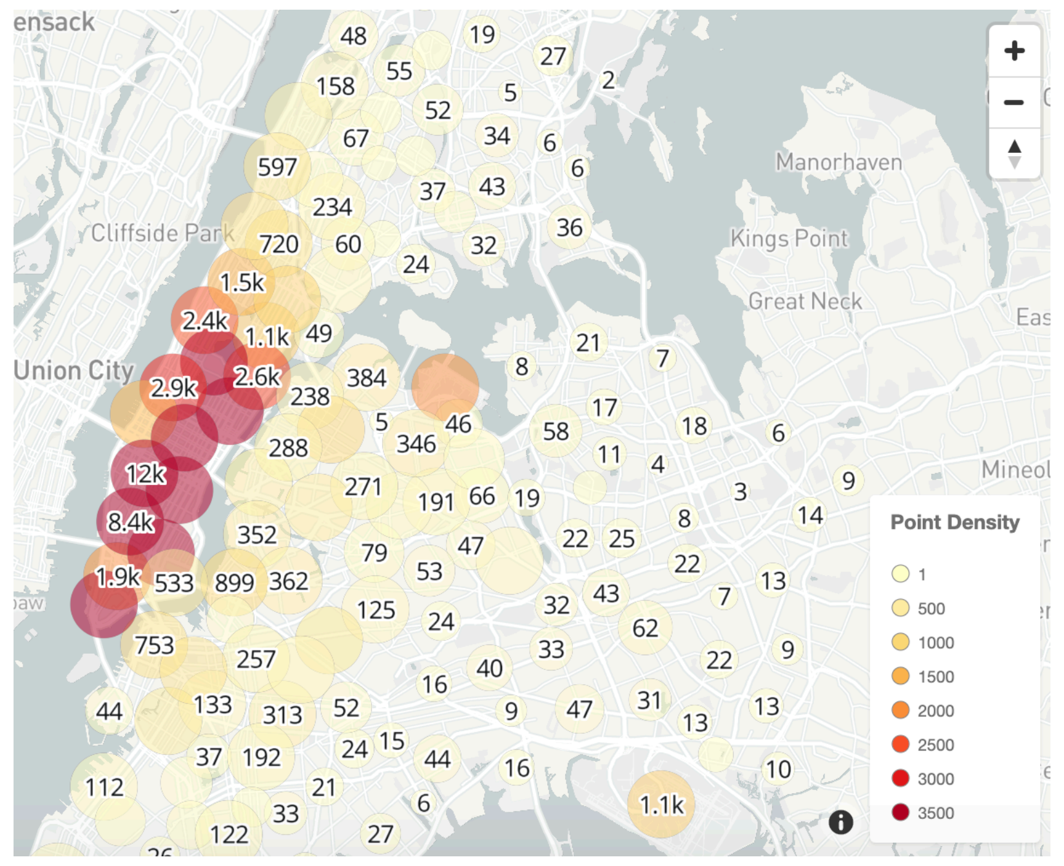

Figure 2: Dropoff counts of all the regions

Data Visulization The above two graphs show the total number of observations for each region during the year of 2017. Figure 1 shows the pickup counts and Figure 2 shows the dropoff ones. We see that most pickup and dropoff occurrences concentrate at midtown and lower Manhattan. We also found that the dropoff locations are more spread out to the east of New York (Brooklyn area). Besides the Manhattan area, other hot spots that stand out, including JFK and LGA, both of which are airports.

Coordinate Mapping We used the centroids' longitude/latitude to represent each area. The average of other areas' coordinates is used for areas with the "unknown" label. The original region IDs are kept as features.

Feature Engineering We extract separate features for year, month, day, weekday, hour and minute from the date and time of each ride, as well as the rain and snow amount per hour using *WorldWeatherOnline* weather API for historical weather data in millimetres. Besides, US holidays are included as a binary feature. Based on location coordinates, we calculate Haversine distances, Dummy Manhattan distances and driving directions. To utilize global information, we compute pick up / drop off frequency of each region.

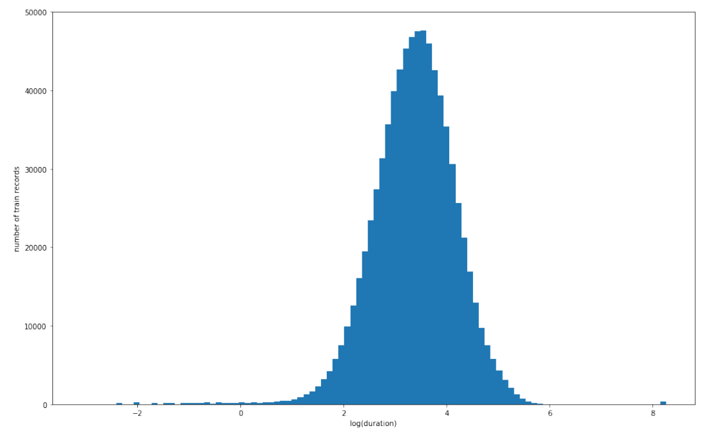

Figure 3: Distribution of logged duration based on date time

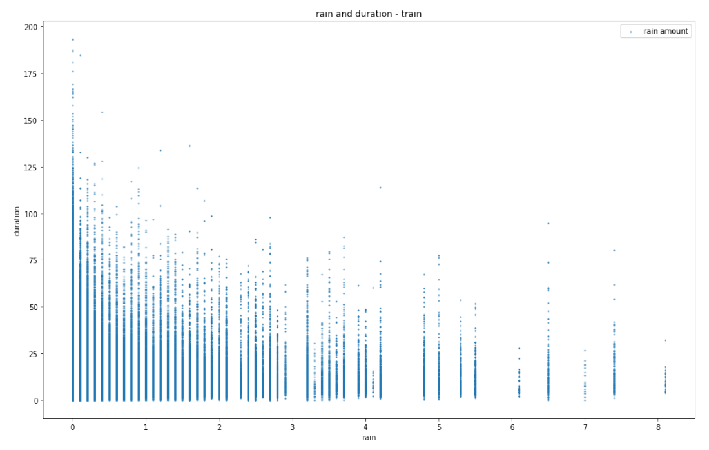

Figure 4: Correlation between duration < 200min and rain

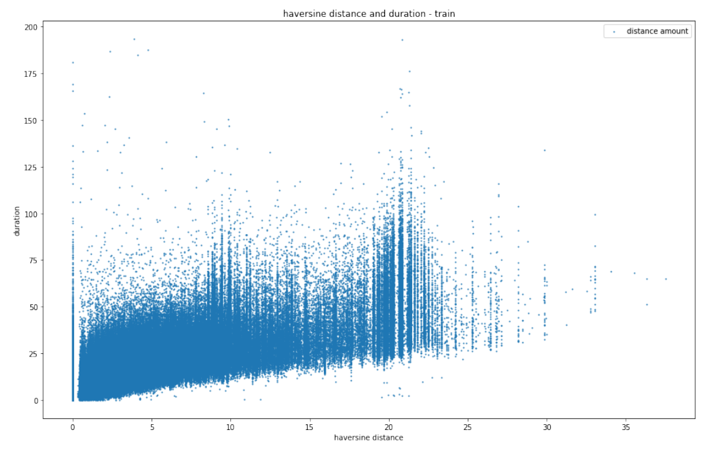

Figure 5: Correlation between duration < 200min and distance

Methodology

Gradient Boosted Tree We train a gradient boosted tree model using XGBoost[1]. Since the dataset is huge ($\sim$12GB after pre-processing in memory), we use external memory mode, which caches the processed feature matrix in the disk. We use approximate tree splitting mode to speed up training. To prevent overfitting, we restrict the maximum depth of the decision tree to 15. We choose squared error as the objective function.

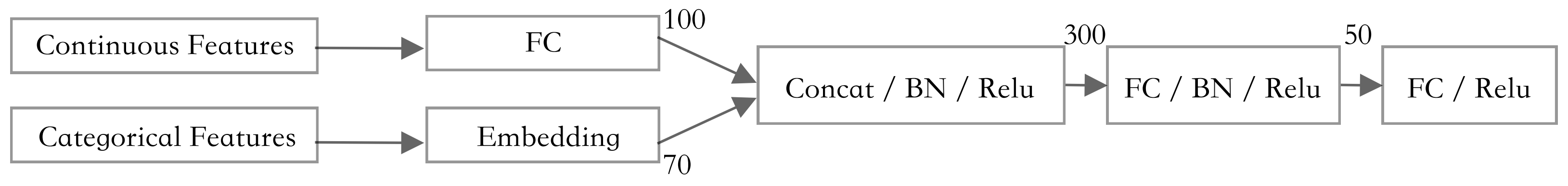

Figure 6:Network Architecture

Neural Network As shown in Figure 6, the network takes two sets of features: continuous features, and categorical features. The continuous features are fed into a fully connected layer. Embedding layers are used for categorical features. The embedding layer is a simple lookup table that maps category id to parametric vector representation. Continuous features and categorical features are concatenated after the first layer and then fed into blocks of fully connected, batch normalization and ReLU layers. Due to the dataset size and the limited number of features, this model is difficult to optimize using gradient descent. Experiments show that this model does not achieve competitive performance.

Random Forest The random forest model is another meta estimator that fits a number of decision trees. Without constraints of the tree depth, the model is likely to overfit. We empirically choose the maximum tree depth to 15. The model consists of 32 estimators. Due to the memory limit, we are unable to train larger random forest model.

Model Ensemble To aggregate predictions of different models, we train another linear regression model on top of individual predictions using mean square error objective function. The output is taken as the final prediction.

Experiments

K-fold cross validation result on the validation set is reported, where $k = 5$ in our experiment. Two metrics are used: mean absolute error (MAE) and root mean squared error (RMSE) in seconds as suggested by [2]. A linear regression model is trained as baseline. Table 1 shows the prediction error of each models. The single best model is gradient boosted tree.

To verify the effects of the external features, we also train the gradient boosted tree model with different features. Table 2 shows than compared with original features, adding external features boost model performance.

Model	MAE	RMSE
Linear Regression	431.12	632.87
Random Forest	260.74	400.39
Gradient Boosted Tree	202.29	339.63
Neural Network	623.35	827.28
Ensemble	202.24	339.56

Table 1: Prediction Error of Different Models

Features	MAE	RMSE
Original	203.03	340.01
+ Coordinates & Distance	203.03	340.00
+ Holiday	202.59	339.64
+ Weather	203.01	339.99
+ Region Frequency	202.56	339.71
+ All External Features	202.29	339.63

Table 2: Ablation Study on Feature Engineering

Summary

We explored feature engineering and different models for the taxi duration prediction task. The result showed that the gradient boosted tree method has the best performance. Future work includes engineering more external features and using graphical models to utilize road link information.

Reference

[1] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *KDD '16*.

[2] H. Wang, Y.-H. Kuo, D. Kifer, and Z. Li, "A simple baseline for travel time estimation using large-scale trip data," in *SIGSPACIAL '16*.

10-701 December 6, 2019

zijingg, jingwei5, wuweil @andrew.cmu.edu