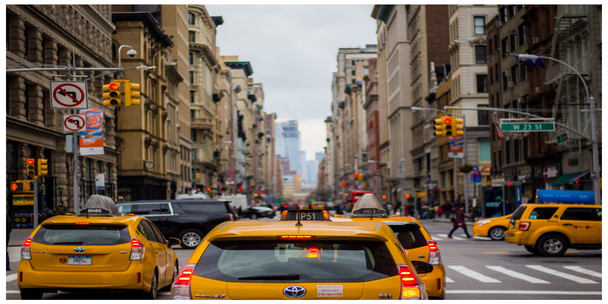


New York City Taxi Trip Duration

Bingxin Lu

Intro to Machine Learning Applications: MGMT 6560

Rensselaer | LALLY SCHOOL OF MANAGEMENT

Summary

In the Kaggle competition New York City Taxi Trip Duration, the challenge is to predict the duration of each taxi trips in the test dataset using the model built on features related to taxi trip. The dataset of this competition comes from the 2016 New York City Yellow Cab trip record data published by the New York City Taxi and Limousine Commission (TLC).

After benchmark other solutions, we find that for feature approach, the common ways include generate new features and remove outliers based on the distribution of variables. Because the dependent variable is trip duration, from the given latitude and longitude information, we can generate new features such as distance and direction, from the given date and time of pickup and dropoff, we can calculate trip duration and hence the corresponding speed. And then use correlations and feature importances to decide those features that can help improve the performance of our model. After further analysis, the results show that new created features “distance” and “direction” are very helpful in improving the accuracy of our predicting models. On the other hand, the feature “store_and_fwd_flag” can be ignored, as it did not improve our models.

For our problem, the goal is to predict trip duration, so we should build a regression model. Linear regression, random forest, gradient boosting, XGBoost and LightGBM are all advanced algorithms solving regression problems. In the modeling section of the paper, I compared the performance of three different models: random forest, gradient boosting and LightGBM. The results show that overall, LightGBM performs best in predicting the trip duration in testing dataset.

Datasets

The train dataset contains almost 1.5 million trip records (1458644 to be exact) and has 10 different attributes. The attributes can be divided into three categories. First, basic description of every trip, including who is the provider of that trip, the number of passengers in the taxi, whether the taxi had a connection to the server or not, and duration of every trip, measured in seconds. Second, date and time of pickup and dropoff. Third, location information, including pickup longitude and latitude, dropoff longitude and latitude.

The test dataset is smaller, contains 625134 trip records. The test dataset is removed two attributes from the train dataset, date and time of drop-off, and duration of the trip, for us to use the trained data to predict the duration, because the duration of every trip is calculated by dropoff time minus pickup time.

The method of accuracy evaluation is Root Mean Squared Logarithmic Error (RMSLE), because we want our prediction to be as close as the actual duration as possible, so for the final result, the smaller the score is, the better its performance.

Benchmarking of Other Solutions

Kernel Name	Feature Approach	Model Approach	Score
①EDaongam	1.Delete outliers and duplicated values 2.Drop the feature indicates whether the trip was held in vehicle memory 3.Calculate distances from pickup to dropoff 4.Convert string to datetime for pickup and dropoff date time	Random Forest	0.40302
② NYTaxis_ComeMS: Machine Learning	1.Calculate distances between pickup and dropoff 2.Create a new feature called ‘zone’ indicates where the pickups are 3.Filter data, remove small distances and fast trips 4.Convert datetime	XGBoost	0.39939
③ML Workflow	1.Remove outliers and duplicated values 2.Make a log-transformation of trip duration’s data 3.One-hot encoding binary categorical features 4.Create date features and direction 5.Calculate distances between pickup and dropoff, remove distance outliers 6.Create speed features, remove speed outliers 7.Correlations and dimensionality reductions	LightGBM	0.37697

Initial Processing

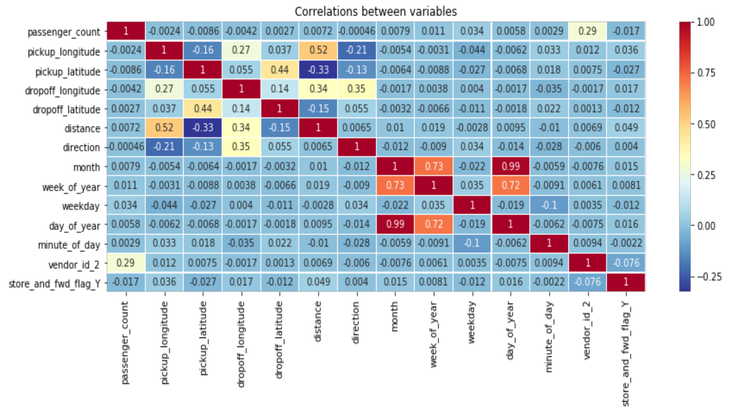
Included Features:

- → vendor id
- → store and fwd flag
- → passenger count
- → pickup longitude
- → pickup latitude
- → dropoff longitude
- → dropoff latitude

Derived Features:

- → month
- → week of day
- → weekday
- → day of year
- → minute of day
- → direction
- → distance

Analysis of Relevance of Independent Variables



Analysis of Performance of Different Model Types

Model		Feature Set 1	Feature Set 2	Feature Set 3
Random Forest	Before Tuning	0.3472	0.3475	0.3469
	After Tuning	0.3447	0.3452	0.3446
LightGBM	Before Tuning	0.3432	0.3432	0.3423
	After Tuning	0.3101	0.3098	0.3092
Gradient Boosting	Before Tuning	0.4050	0.4050	0.4050
	After Tuning	0.3880	0.3880	0.3880

Conclusions

- ➔ Overall, there are not very big difference of the performance of the three feature sets, removing features “store_and_fwd_flag”, “vendor_id” and “passenger_count” can slightly improves models.
- ➔ LightGBM and random forest get almost the same results before hyperparameter tuning, but after hyperparameter tuning, the performance of LightGBM is greatly improved, and LightGBM gets the best score of three models.