NEW YORK CITY TAXI TRIP DURATION

TIANJIAO WANG

Contents

Date: 2023-12-02.
2020 *Mathematics Subject Classification.* Artificial Intelligence.

1. Introduction

The project will 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. Accordingly, this project problem is taxi trips duration, which is a outlier detection. There are 6 steps including data loading and overview, data cleaning, features engineering, model selection, hyperparameters tuning, and training and predictions.

- Data loading and overview
- Data cleaning
- Features engineering
- Model selection
- Hyperparameters tuning
- Training and predictions

2. Data loading and overview

At first, I quickly look at the first 5 lines of a dataset to understand the structure, format, and content of the data . Then I take a overview of the type and amount and other information of df and test data.

Colonne	Description
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

Figure 1. overview of the data

3. Data cleaning

I do the data learning to check the duplicated and missing values and deal with outliers.

```
id                  0
vendor_id           0
pickup_datetime     0
dropoff_datetime    0
passenger_count     0
pickup_longitude    0
pickup_latitude     0
dropoff_longitude   0
dropoff_latitude    0
store_and_fwd_flag  0
trip_duration       0
dtype: int64
```

Figure 2. No duplicate and missing data

- Visualize outliers

There are outliers. I can't find a proper interpretation and it will probably damage our model, so I choose to get rid of them.

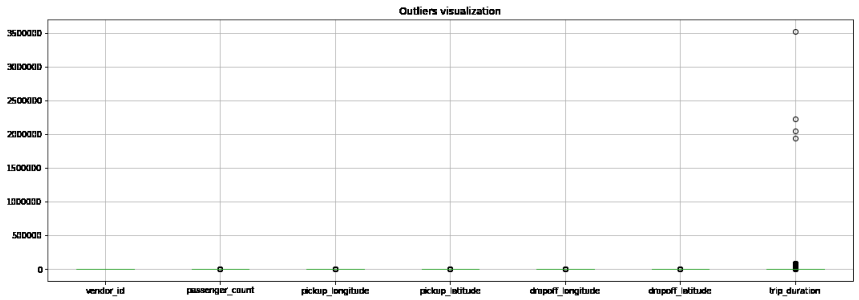

Figure 3. outliers for trip-duration

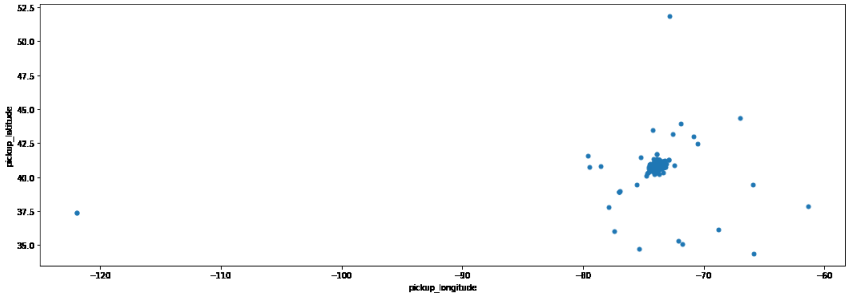

Figure 4. outliers for pickup positions

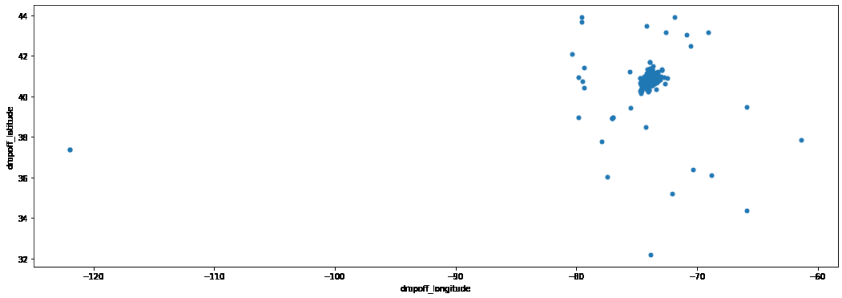

FIGURE 5. outliers for dropoff positions

In this step, I only keep trips that lasted less than 5900 seconds, and only keep trips with passengers.

4. FEATURES ENGINEERING

At first, I visualize the distribution of trip-duration values.

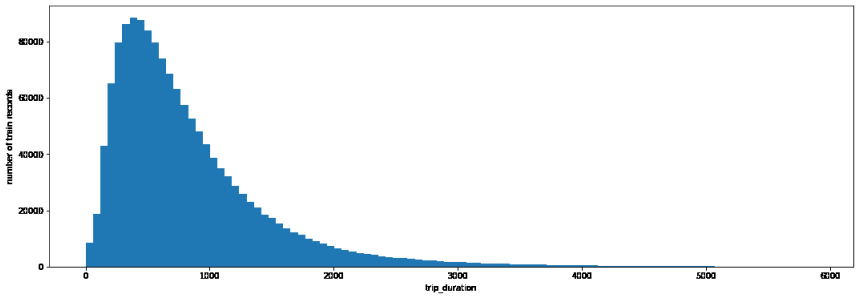

FIGURE 6. the distribution of trip-duration values

The distribution is right-skewed so we can consider a log-transformation of trip-duration column.

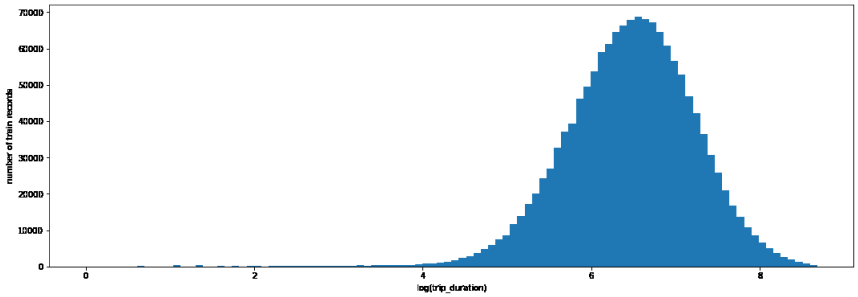

FIGURE 7. log of trip-duration

Then I deal with categorical features using One-hot encoding binary categorical features. One-hot encoding is a common technique used in data preprocessing,

particularly in machine learning, for handling categorical data. It's particularly useful when the categorical data is nominal (i.e., there is no intrinsic order to the categories).
Then I deal with dates, and ccreat distance and speed. Finally, I remove the outliers. Then we test the correlations between variables and made a figure.

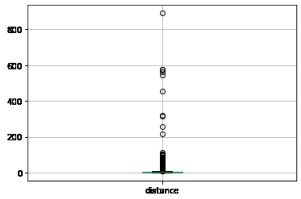

FIGURE 8. outliesrs of distance

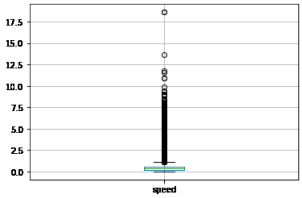

FIGURE 9. outliesrs of speed

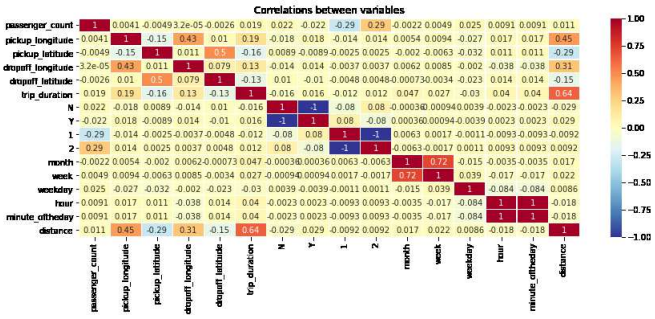

FIGURE 10. correlations between variables

5. MODEL SELECTION

In this part, we split the labeled data frame into two sets: features and target to train, and then test the models. Then, for this specific problematic, we'll measure the error using the RMSE(Root Mean Squared Error). We tried Gradient-Boosting, RandomForest, and LightGBM. LightGBM is blazingly fast compared to RandomForest and classic GradientBoosting, while fitting better.It is our clear winner. Then we did cross-validation. The result shows that our LightGBM model is stable.

6. Hyperparameters tuning

We did hyperparameters tunning using RandomizedSearchCV, and Test the following parameters:'learning-rate', 'max-depth', 'num-leaves','objective':'regression', 'feature-fraction', 'bagging-fraction', 'max-bin'.

7. Training and predictions

We trained on all labeled data using the best parameters in hyperparameters tuning, and trained on all labeled data using the best parameters (sklearn API version), and trained on all labeled data using the best parameters. Then we make predictions on test data frame, create a data frame designed a submission on Kaggle. Lastly, create a csv out of the submission data frame.

	id	trip_duration
0	id3004672	716.070826
1	id3505355	672.125770
2	id1217141	455.368356
3	id2150126	938.637832
4	id1598245	354.432595

Figure 11. predict-result

List of Todos

(A. 1) Business School,, Beijing Technology and Business University, Beijing 100048, China

Email address, A. 1: wangtianjiaoj@gmail.com