STAT 479

Prediction of taxi trip duration in New York City

Jason Su | Quenu Chen | Zheming Lian | Jiahui Jiang

Introduction

The goal of this project is to predict the duration of taxi rides in New York City based on features like trip coordinates and pickup date and time. The data comes from 1.5 million observations on the NYC Taxi and Limousine Commission (TLC). Data sets about NYC weather is also included for prediction accuracy. We will provide users with our result through shiny app.

Motivation

NYC, as a popular city, attracts more and more visitors nowadays. Visitors, however, are not familiar with native transportations in the city. Thus, we intend to do a prediction of visitor's travel time by taxi in order to help them properly manage time. By comparing the data of Uber and taxi in NYC, we decide to focus on taxi which is more popular in NYC.

Data cleaning

Extreme trip duration: we removed data that the duration is close to or longer than 10 hours since it is inconceivable that someone takes a taxi for a trip that lasts almost a day.

Shorter than a fewer minutes: we also removed observations that the distance is close to zero or the trip duration is less than 30 seconds.

Visualization

After cleaning the dataset, we needed to find some new features and their impact on the target trip duration values.

Based on trip coordinates and trip duration, we used the actual trip distance[1] and calculated the average speed (actual distance/trip duration). Then We plotted the median speed over six

[1] We accessed google api and used googleway package to get the actual distance based on the longitude and latitude of pickup point and drop off point.

months. It turns out that there exists similar patterns in every week. (speed oscillates almost regularly by date)

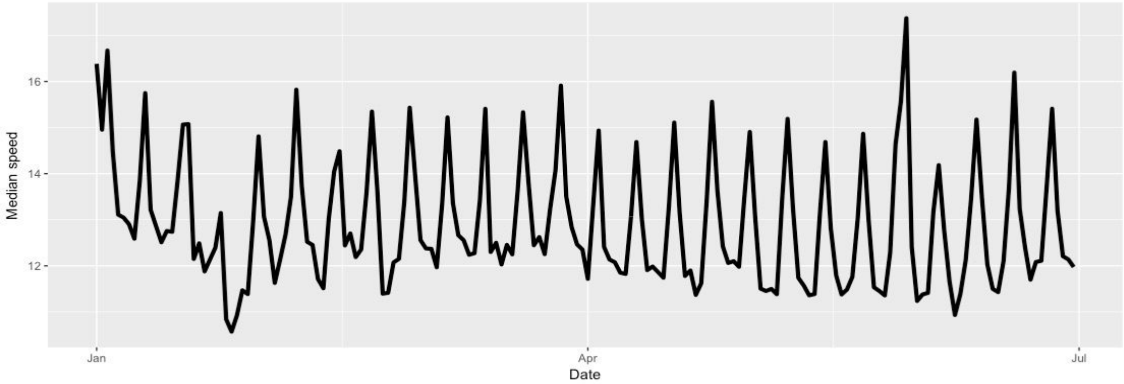


Figure 1

Based on the result we found, we added another variable— day of the week and made a plot to see their effects on median speed.

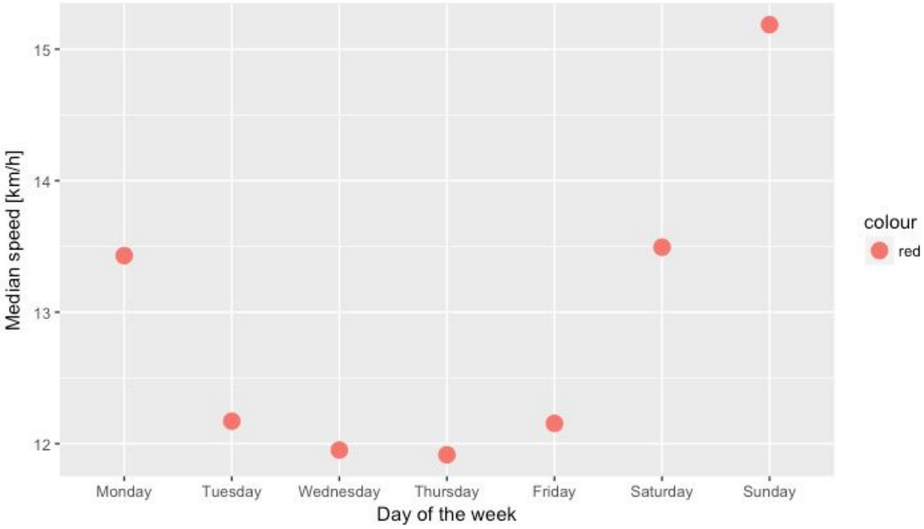


Figure 2

From figure 2, taxis appear to be travelling faster on the weekend than on weekdays. It makes sense that traffic is busier on work days than on weekends. It also accords with our inference that the day of the week will affect travel time.

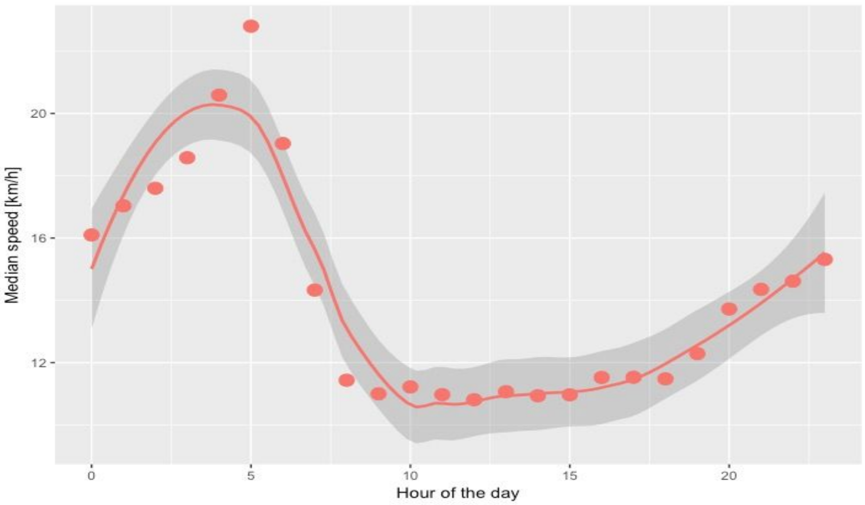


Figure 3

Moreover, rush hours in a day may also lower the travel speed. To prove our guess, we made a plot contains speed and hours of a day (Figure 3). It turns out that the early morning hours allow for a speedier trip, with everything from 8am to 6pm being similarly slow.

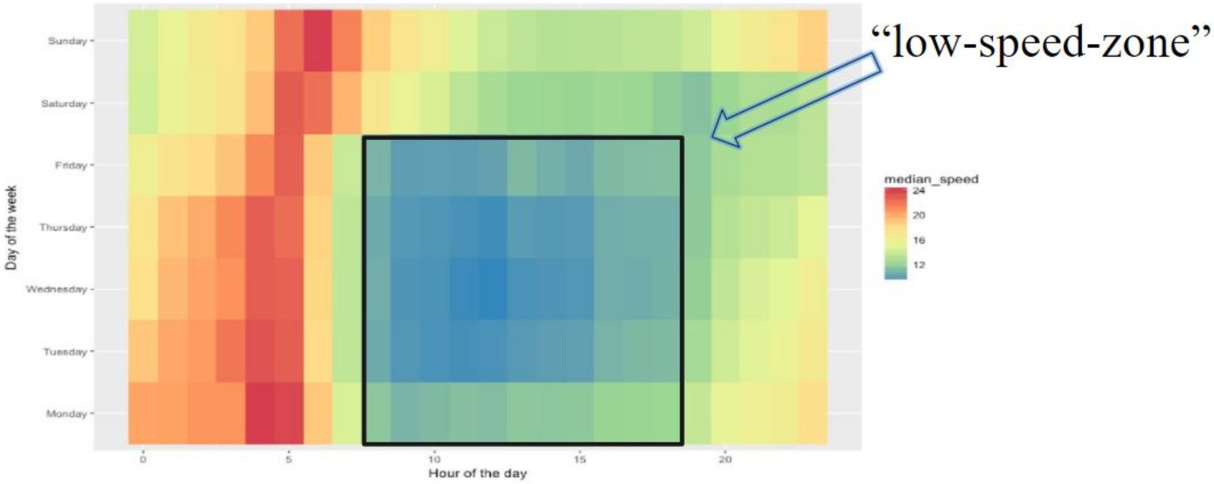


Figure 4

Next we wanted to figure out the overall pattern of speed in different days and hours. The heatmap (Figure 4) visualises how these trends combine to create a "low-speed-zone" (from 8am to 6pm through Monday to Friday).

Model

Model selection: We chose KNN regression model and we used the "FNN" package to build the model. KNN is a simple machine learning algorithm for classification problem. But it can also be applied to regression problem. Our taxi trip data set contains 1.5 million data, which is large enough to have strong predictive power. Furthermore, the short running time is very important for our final shiny app since users will lose patience during long waiting time.

Ideal training feature: In the visualization part we just made several assumptions on possible features, and here we used cross validation to determine the training features. More specifically, we used 10-folds cross validation, and the results are as follows:

cv10(pickup_latitude, pickup_longitude,dropoff_latitude,dropoff_longitude) = 408.7
cv10(pickup and dropoff coordinates, pickup_hour, weekday) = 369.62
cv10(pickup and dropoff coordinates, pickup_hour, weekday, rained, snowed,) = 391

408.71, 369.62, and 391 represent the RMSE (in seconds) of cross validation on only geographic coordinates, geographic coordinates plus hour and weekday, geographic coordinates with hour and weekdays plus weather conditions, respectively.

$$\text{CV}(10) = \frac{1}{10}\sum_{i=1}^{10} RMSE\,i$$

The result shows that RMSE minimizes with geographic information and time information. Therefore, we chose pickup latitude, pickup longitude, drop off latitude, drop off longitude, pickup hour and weekday to be our training features.

Baseline comparison: To get a sense about the power of model, we made a comparison between KNN and a plain model. Such model generated a prediction by randomly sampling from the distribution of trip duration in the training set. [2]To measure the error rate of this plain model, we made a 10-folds cross validation and the average error rate is 953.544. Therefore, Compared with this plain model, KNN reduces 61.2% of error.

ShinyApp

In order to apply our research to solve practical problem, we built a Shiny App with interactive map which can let users estimate their trip duration conveniently and arrange their travelling plan reasonably.

Users can choose their pickup location and drop off location freely through clicking on the map. In addition, users are able to choose their estimated pickup time and date to obtain more accurate estimation of duration. After setting up, users can click the button on the right panel to confirm their trips to get accurate trip duration.

[2] In R, although it is possible visualize the distribution by plot(density()), how to directly sample data from the density() remains unclear. Hence, we use approxfun() function to simulate the distribution, which can be imported in sample() as "prob" parameter. A plot is included in the Appendix to show the difference between distribution generated by density() and by approx().

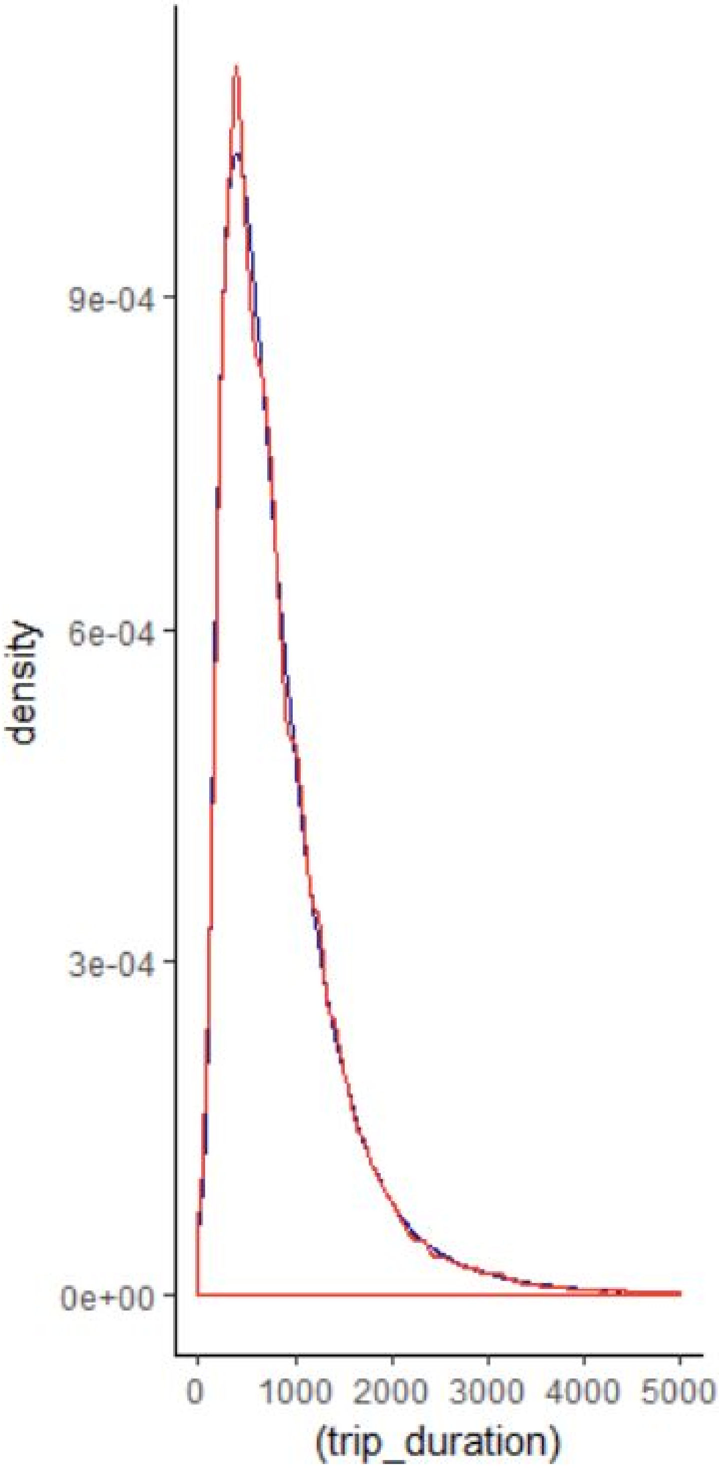
Except for the estimated duration, we also provided roughly estimated durations of this trip for different pickup time. Users can get general idea about the peak-hours of this trip. These estimated durations are rough because it was not obtained by strict KNN method. To reduce running time, we picked similar trips of this trip from dataset and get the mean of these picked trips for different pickup time as roughly estimated durations. The definition of being similar is that the distance of estimated trip's pickup location and picked trip's pickup location should not exceed one kilometer. The same rule applies to the drop off location. If they decide to change their pickup time, they could reset the pickup time to get accurate estimated duration for reset pickup time. Due to the lack of professional knowledge of website design and CSS system, the Shiny App might exceed the display field for some relative small screens. To solve this issue, we designed the operation panel to be draggable. When the bar graph is blocked, users can drag the panel upwards to clearly see the graph.

Appendix

1. ShinyApp Links: https://479groupproject.shinyapps.io/taxiapp/

2. Comparison between distribution generated by different function:

Red curve stands for distribution generated by approxfun(), black curve stands for distribution generated by density().


density
9e-04
6e-04
3e-04
0e+00
(trip_duration)