NYC Taxi Trips Analysis and Prediction

Team Member: Jiahui Luo, Qin Yin, Zhe Feng

Introduction

In NYC, there are approximately 200 million taxi rides each year. In order to increase the efficiency of the city's taxi system, we can deeply understand the taxi supply and demand. Especially in NYC, people have more chances to take a taxi than any other cities. Instead of booking a taxi by phone by Uber or Lyft, NYC taxi drivers pick up passengers on street.

The ability to predict taxi ridership can present valuable insights to city planners and taxi dispatchers in answering questions such as how to position cabs where they are most needed, how many taxis to dispatch, and how ridership varies over time.

Our goal is, given the pickup and dropoff locations and the time window, predicting the ride duration of taxi trips in NYC.

Dataset

We use two datasets (training set and testing set) from Kaggle detailing all taxi trips in NYC from January to July in 2016, as provided by the NYC Taxi and Limousine Commission (TLC). Fig. 1 demonstrates that the data associates each taxi ride with information including vendor id (1 and 2), datetime, and longitude and latitude of pickup and dropoff locations, passenger count and trip duration. The training set contains 1458644 trip records and the testing set contains 625134 trip records.

or_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration
2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.982155	40.767937	-73.964630	40.765602	N	455
1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.980415	40.738564	-73.999481	40.731152	N	663
2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.979027	40.763939	-74.005333	40.710087	N	2124
2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.010040	40.719971	-74.012268	40.706718	N	429
2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.973053	40.793209	-73.972923	40.782520	N	435

Fig. 1: Data fields of training set

Data Preprocessing

To better understand the problem at hand and the features, we try to understand the issues with our data and perform data preprocessing on the data.

1) Issues:

- There are some trips that have no passengers.

- The range of trip duration is between 1s to 41 days.
- A number of taxi pickups and dropoffs in the dataset locate well outside the NYC area.

Fig. 2: A number of taxi pickups location

2) Missing Data:
   No missing data is found.
3) Passenger Count Clean-up:
   We eliminate all the trips with no passengers.
4) Trip Duration Clean-up:

To deal with the outliers associated with the trip duration, specifically the 1s and 41 days trip duration, we exclude data that lies outside 2 standard deviations from the mean.

5) Latitude and Longitude Clean-up:

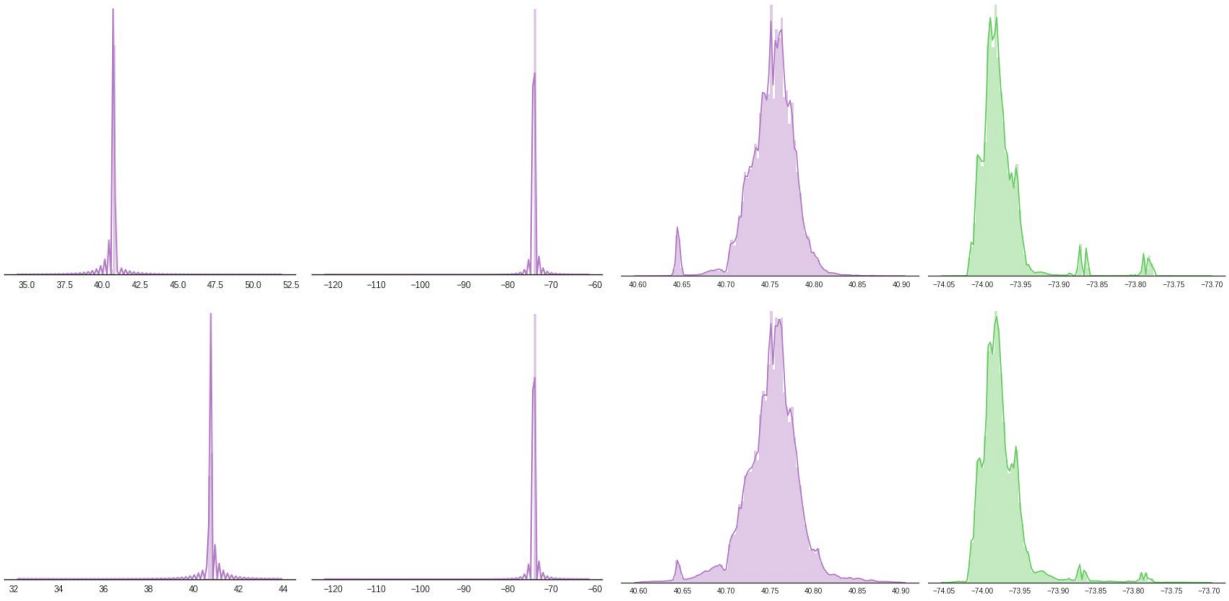

Fig. 3: Latitude and longitude before Clean-up

Fig. 4: Latitude and longitude after Clean-up

Fig. 3 shows that pickup and dropoff latitude are centered around 40 to 41, and longitude are situated around -74 to -73. Trips which are very far from each other like latitude 32 to latitude 44,

are taking very long time, and have affected this plot. As a result, we remove those large duration trips by using a cap on lat-long and visualize the distributions of latitude and longitude on Fig. 4.

Then we put the following caps on lat-long:

- latitude should be between 40.6 to 40.9
- Longitude should be between -74.10 to -73.75

We can see that most of the trips are getting concentrated around longitude -73.98 and latitude 40.76. At the meanwhile, we get one additional distribution spike in latitude around 40.65 and two additional spikes in longitude around -73.87 and -73.79. Where are they?

After searching google map, these two airports give us the answer:

- LaGuardia Airport (40.77, -73.87)
- John F. Kennedy International Airport (40.65 , -73.79)

Now, let's go further to perform an exploratory analysis on the data.

Data Exploration

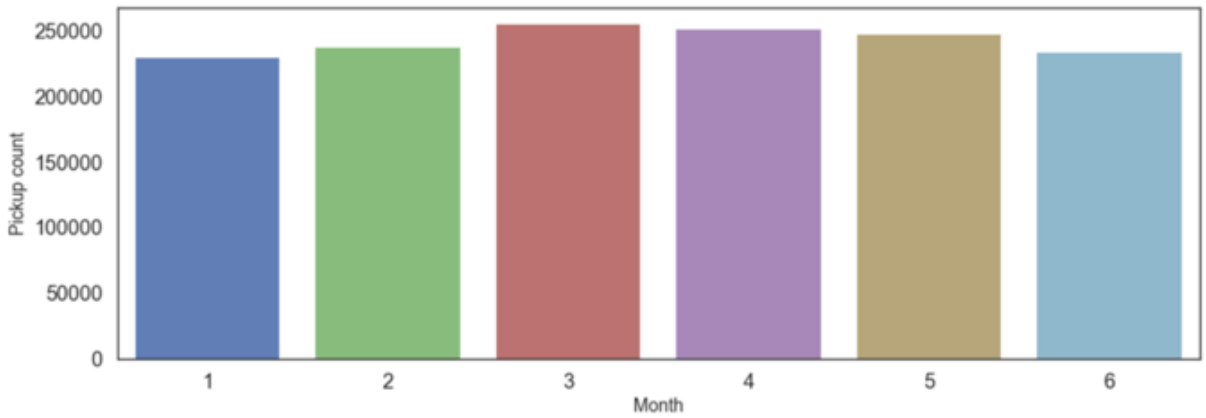

Fig. 5: Pickup count vs month of year

Fig. 5 shows the distribution of taxi request numbers over the year. Since the dataset only contains taxi data between January 2016 and June 2016, only half of the year's data is shown here. We can tell that the busiest month for taxi request is on March while the least busiest month is on January. Though the difference is not that big by the visualization, March has about 10% more request than that in January, which is about 20,000 request differences.

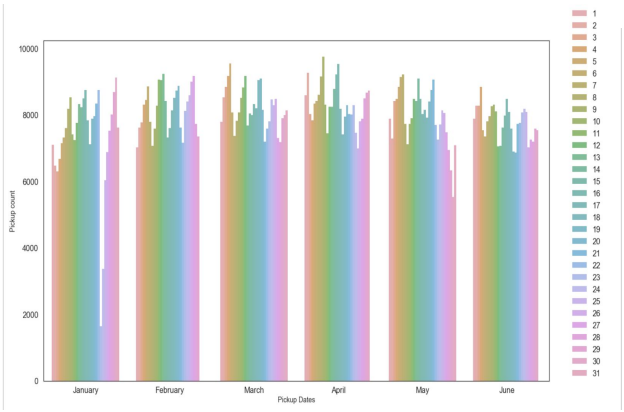

Fig. 6 : Pickup count vs month of year

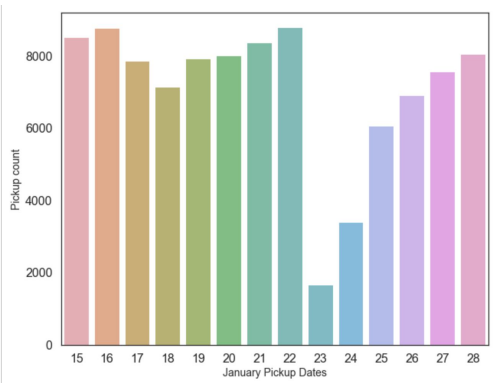

Fig. 7 : Pickup count vs day of January

Fig. 6 also shows the distribution of taxi request numbers over the year. The difference between Fig. 5 and Fig. 6 is Fig. 6 also shows the taxi request numbers every day. The monthly distribution pattern looks very similar for each month, which means our data is pretty homogeneous. However, we notice that the count number decreases abnormally at the end of January. Fig. 7 shows the zoomed version of that “jump” and we can find that the change happened on January 23rd. We initially thought it was because of the data missing on that particular week; however, after searching for New York’s weather on that day, we found out that the huge decrease in picking up count was because of the heavy snow storm [1]. We can also infer that a lot of tourists had to cancel their trips because of the severe weather. Since one of the major taxi customers are tourists, those cancelled trips also influenced the taxi requests in the following days.

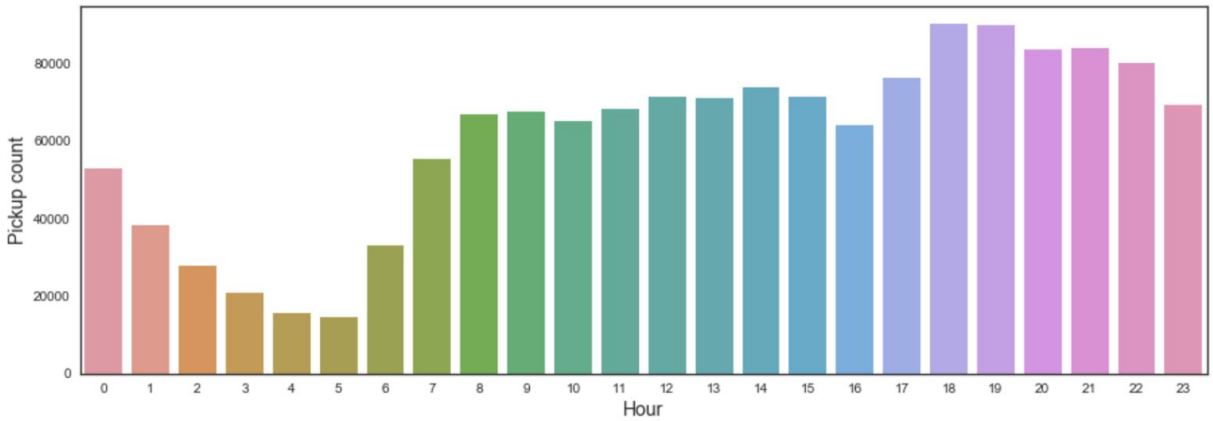

Fig 8: pickup count vs hour of the day

Fig 8 shows that the trip requests number is not flat all the time. The busiest hour happens between 5pm and 11pm while the least busiest hour happens between 1am and 6am. This inspires us that the hour itself might be useful features in data training model part.

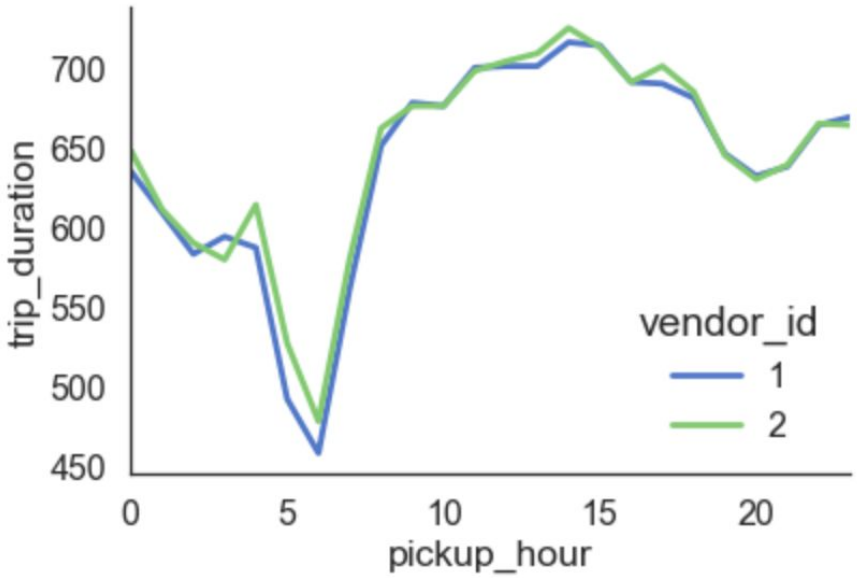

Fig 9. trip duration vs pickup hour

Since we have two vendors in the dataset, we are curious to find out whether they are different in terms of trip duration. We use median duration as our datapoint and draw the plot to show the trip duration vs time of the day (Fig. 9). As we can see, the trip duration pattern looks very similar among those two vendors, which indicates that vendor_id will contribute little as features in trip duration prediction.

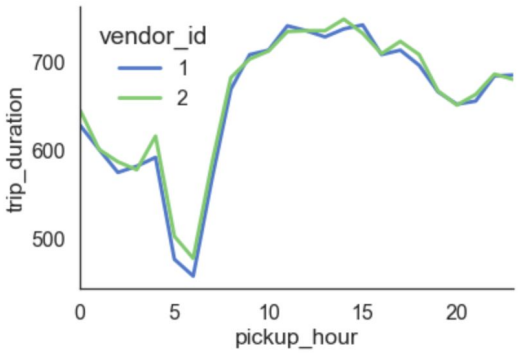

Fig 10. Trip duration vs day of weekday

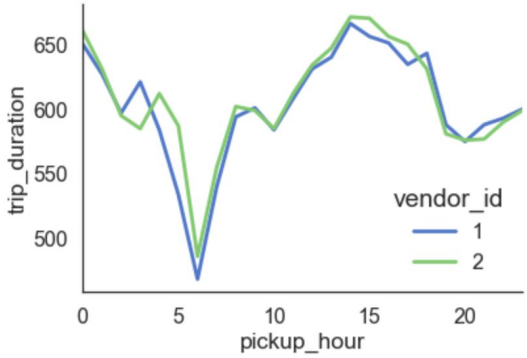

Fig 11. Trip duration vs day of weekend

Inspired by Fig. 8, we are also curious to find out whether weekdays and weekends have influence on trip durations. We plot weekdays (Fig. 10) and weekends (Fig. 11) separately, and found out that they look very similar, which indicated that weekday/weekend might not be important features in our model trainings.

Now let's see how we can apply those features in our data models.

Data Modeling

A. Ridge Regression
We started with using the most basic model which is linear regression, but it was overfitting on the training set leading to very high values of weights. To overcome this overfitting, we decided to use the Ridge Regressor. The Ridge Regressor solves a regression model where the loss function is defined as the linear least squares function

B. Random Forest
Random Forest is an ensemble learning method, which constructs a multitude of decision trees at training time and outputting the class that is the mode of mean prediction of the individual trees. It correct decision trees' habit of overfitting. It aggregates many decision trees built on bootstrapped samples of the training data in order to reduce the high variance of a single decision tree and improve prediction accuracy.

C. XGBoost
XGBoost, another ensemble method, is short for "Extreme Gradient Boosting", which is an implementation of gradient boosted decision trees designed for speed and performance. It has advantage of fast execution speed and excellent model performance.

Evaluation

A. Evaluation Metrics
In regression problems, the output is always continuous in nature and requires no further treatment.

The coefficient of determination R-squared (1) of the prediction, where R-squared is a statistical measure of how close the data are to the fitted regression line.

$$R^2 = \text{Explained variation} \,/\, \text{Total variation} \qquad (1)$$

R-squared is always between 0 and 100%: 0% indicates that the model explains none of the variability of the response data around its mean; 100% indicates that the model explains all the variability of the response data around its mean. In general, the higher the R-squared, the better the model fits your data.

RMSE (2), a frequently used measure of the differences between values predicted by a model and the values actually observed, is the most popular evaluation metric used in regression problems. It follows an assumption that errors are unbiased and follow a normal distribution.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(predicted_i - Actual_i)^2}{N}} \qquad (2)$$

It indicates the absolute fit of the model to the data–how close the observed data points are to the model's predicted values. Lower RMSE indicate better fit.

B. Performance:
(a) Ridge Regression

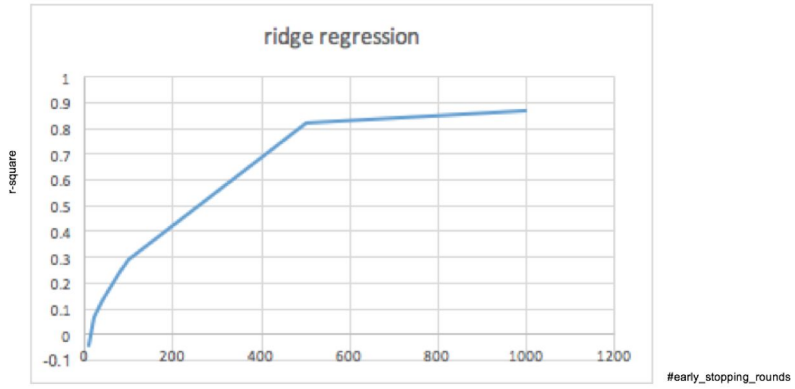

Fig. 12 Ridge Regression Performance

From Fig. 12, we can see that when we try to tune the early-stopping-rounds for ridge regressor, the r-squared increasing accordingly, which means that it performs better and better. We also noticed that when the early-stopping-round is up to 500, its performance become stable, the highest r-squared score near 0.88 is achieved when the early-stopping-round up is 1000.

(b) Random Forest

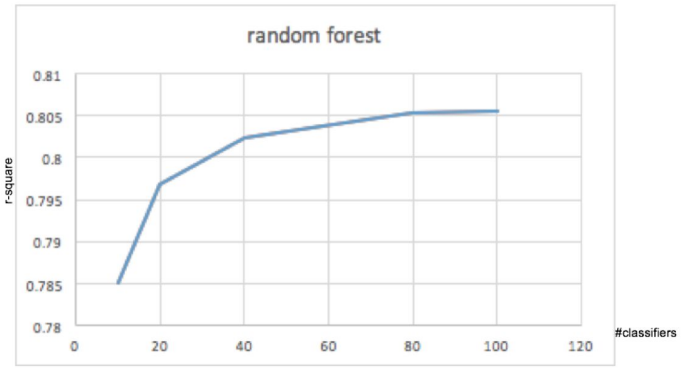

Fig. 13 Random Forest Performance

From Fig. 13, we can see that when we try to tune the number of classifiers for random forest, the r-squared increasing accordingly, which means that it performs better and better. We also noticed that when the number of classifiers is up to 80, its performance become stable, the highest r-squared score near 0.806 is achieved when the early-stopping-round up is 100.

(c) XGBoost

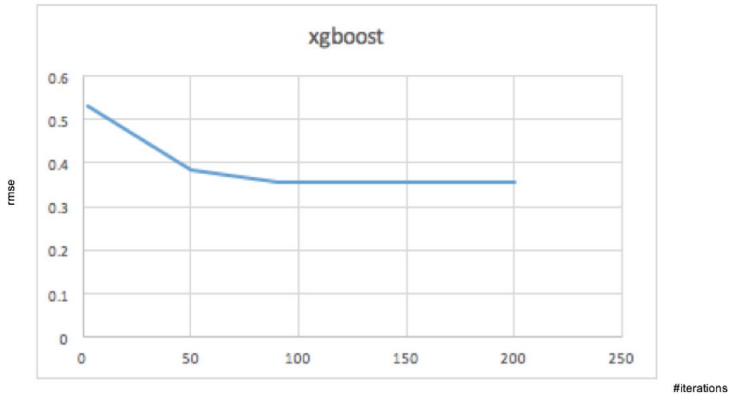

Fig. 14 XGBoost Performance

From Fig. 14, we can see that when we try to tune the number of iterations for XGBoost, the rmse decreasing accordingly, which means that it performs better and better. We also noticed that when the number of iterations is up to 100, its performance become stable, the highest r-squared score near 0.35 is achieved when the number of iterations is 100 and after.

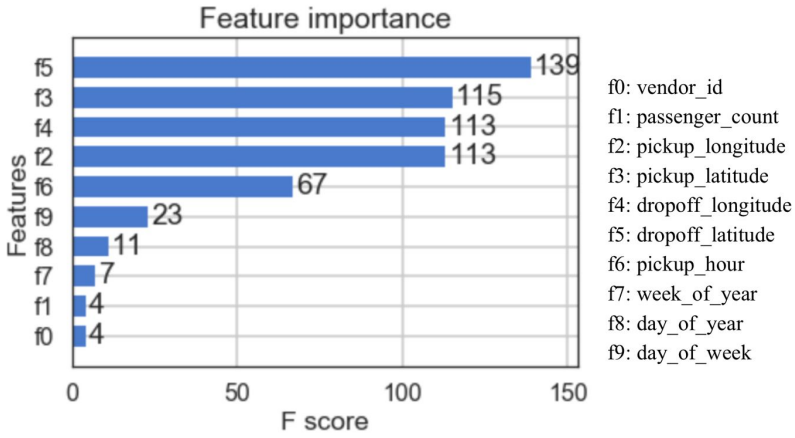

Fig. 15 Feature Importance

We also get the feature importance plot for XGBoost(Fig. 15) and find out that the most important feature for our prediction model is drop-off location, followed by pick-up location. At first, we thought the day of week will be the most important feature for prediction, it turned out to be less important. We guess that the reason of locations being important might be the traffic and parking issue, since there are a lot of one-ways and no-parking zones in Manhattan and other district of New York City.

(d) Discussion

Model	RMSE
Ridge Regression	274.97
Random Forest	107.12

XGBoost	0.35

Table 1 Performance of Three Models

Using the models with best trained performance of Ridge Regression, Random Forest and XGBoost for testing data, we get the RMSE in table 1. We find that Ridge Regression performs not very well compared with Random Forest and XGBoost, since Random Forest and XGBoost are both ensemble method, which generate several models that are combined to make a prediction and Ridge Regression is not. Besides, XGBoost performs dramatically well than Random Forest. We guess the reason might be in XGBoost we leveraged cross validation while in Random Forest we didn't.

(e) Feature work

For the future work, we could focus on following directions: training part and testing part optimization. Firstly, for the training part optimization, we could try train models with different training set and train weekday model and weekend models differently. Besides, tuning hyperparameter may also help us to achieve a better regression models. Secondly, for the testing part optimization, we could try some more evaluation methods.

Conclusion

Ridge Regression, Random Forest, XGBoost were trained to predict ride duration. The results show that Random Forest and XGBoost performed better than Ridge Regression, which was expected. An interesting insight gained was the most important feature in models. The day of week was not as important as what we expected. Drop-off locations turned out to be the most important feature in XGBoost model. We guessed the reason might be the special traffic situation in New York City. Besides, we need to use cross validation for Random Forest to improve its performance compared with XGBoost.

To further improve the prediction accuracy, more data need to be considered and modeled, eg. we could add the weather feature to improve the performance of our model, since it showed that snow storm could influence the trip duration. Further steps could also be taken by giving alternative choice for taking taxi, eg. given the current time and location of the customer, we could provide the customer several trip durations of three or five nearest location, so he/she can make choice by selecting the shortest one.

Reference:

1.https://www.nbcnewyork.com/news/local/NYC-New-York-City-Blizzard-Biggest-Ever-January-23-2016-377435221.html