New York City Taxi Trip Duration Prediction

Yuanxu Wu
Courant Institute, NYU
NY, USA
yw2983@nyu.edu

Shuyi Yu
Courant Institute, NYU
NY, USA
sy1144@nyu.edu

Biqi Lin
Courant Institute, NYU
NY, USA
bl2379@nyu.edu

***Abstract*—Predicting the duration of a taxi trip is helpful for drivers and customers to plan their trips. With the advent of technology and data analytics tools, the task is feasible. This paper uses two models to predict the duration of a taxi trip in New York City. By calculating the root mean square log error, we find that the gradient boosted regression model performs better than the linear regression model.**

Keywords—Taxi Trip Duration, NYC, Spark, Gradient Boosted Regression

I. Introduction

Taxicabs is an important part of daily life in New York City. Both residents and visitors would call a taxi to send them to the right place at any time. There are millions of rides taken each month, which shows the vibrance of the market.

We expect an application with the functionality to predict the duration of taxi trips to be popular in the market. This application will help passengers. When they plan a trip, it could be helpful to know the duration in advance. Based on the results, they can make an informed decision as whether to take a taxi or not, and/or when is the optimal time to start their commute. Imagine a scenario that a passenger wants to go to JFK to catch a plane. The taxi trip may take more or less time based on how busy the traffic is. If it takes long, the passenger had better switch to subway. If not, then he/she needs to decide when to finish packing and call the taxi. This application also helps taxi providers like Uber, as they can identify cars within the minimum driving time to the customer.

This paper implements this idea as to predict taxi trip duration in New York City. For prediction, only data that will be available at the beginning of a ride is used, including start time, pick-up and drop-off coordinates, road traffic and weather condition. We apply different models (linear regression, gradient boosted trees, etc.) and compare the results.

II. Motivation

The NYC taxi trip data provides lots of valuable information including travel patterns and city structure which makes the prediction of trip time possible. Successfully predicting the duration of the trip can not only help passengers manage their travel time and avoid traffic but also help drivers and companies find the better trip routes and predict fare. For example, an important functionality in the UBER app is to estimate the time and fare when we enter the pick-up and drop-off locations. And the customer will evaluate the result to decide whether to take.

Some studies found the travel time and distance have the linear relationship in a short trip, but they failed to predict in a long trip. We are also trying to do the similar prediction, but we will introduce real-time traffic, weather condition data into a more complexed model which is expected to give a more accurate result in both short and long trip.

III. Related Work

The first paper is about trip duration prediction based on up to 21 months of historical data consisting of approximately 250 million paid taxi trips in Singapore [1]. They deployed a real-time trip information system for a large GPS-enabled Singaporean taxi company. The project we are going to employ is also based on real-time information and also about trip duration prediction. This type of information system is potentially useful to a wide range of public transportation operators (including buses, trains, and taxis). They found out five challenges when building their system. First, the mount of the data collected by the taxi company was huge. Also, our taxi trip data is about 10GB. Second, the system needs to answer the question in real-time. For instance, for any query, the system needs to give a relatively accurate prediction in a short time. Third, this application needs to account for various time-related factors. For instance, you couldn't predict the trip duration with the data of a future time stamp. Fourth, we need to determine how much historical information is necessary to provide accurate answers to trip-related queries. For example, is one month of historical data sufficient for accurate predictions or six months? Fifth, the raw data is quite noisy and contains both outright errors and non-standard behavior patterns. To solve the time-related factors issue, they used time windows, which is split the start time dimension inf the search space into non-overlapping time windows so that trips belonging to the same window can be treated as having similar start time and queries for trips with similar start time can be significantly sped up. For example, they used hourly windows. In our project, we are going to use the same method, but in a much shorter time window. To solve the real-time too-partial location issue, they used a dynamic location zones method, which is using K-Nearest Neighbor to achieve dynamic zoning. They find they proper K and treat the start (end) location in the same cluster as sharing similar start feature. I think this is a nice method to employ in our project too. About the noisy raw data and big data problem, we are going to use Spark to solve it, because spark can store the large-scale data into a distributed file system automatically, and process the data really fast. In addition, Spark provides many functions to eliminate the noise in the raw data, and mine the data efficiently.

The second paper tries to predict both duration and fare amount of a taxi trip in New York City [2], which is close to our paper. The paper offers many insights as how to implement the idea. The paper studies the impact of various features on the prediction and also attempts to find the best model to leverage those features. The data contains yellow taxi rides in New York City from Jan 2016 to June 2016 from NYC Taxi and Limousine Commission (about 10GB). The paper first does exploratory analysis. The paper finds that features like distance, time of day, day of week and location (pick-up and drop-off) have an impact on pace (the ratio of duration and distance). This ratio is later multiplied with distance to predict the duration. The paper also finds that features like duration and distance have an impact on fare. The paper adopts the following practices to represent features: 1. distance is inverted; 2. categorical features like month, weekday and hour of day are represented in the form of one hot encoded vectors; 3. pick-up and drop-off coordinates are clustered and the paper tries both using the clusters directly or as one hot encoded vector. This paper considers four models: Ridge Regression, Random Forest, Gradient Boosting and an ensemble of Random Forest & Gradient Boosting. The paper splits the data into training set (60%), validation set (20%) and test set (20%) and uses Root Mean Square Error (RMSE) to measure the accuracy of models. In the analysis, the paper finds that taking the combination of pickup and drop locations as a feature instead of taking them separately resulted in significant improvement in performance. And the paper gets lowest RMSE of 4.87 with Gradient Boosting Regressor for duration prediction and lowest RMSE of 2.49 with Ensemble of Random Forest and Gradient Boosting Regressor for fare prediction. This paper sheds light on what features and models to choose to conduct our prediction on taxi duration. The paper also suggests how to process taxi data to better represent these features in the model. But our paper considers extra datasets, i.e. traffic and weather data, so we need to think about how to fit these data into the prediction.

In 2016, Sun et al. proposed a mobile application, which produces services such as traffic conditions, frequent pick-up/ drop-off locations, and regulate rates at the different time of day in NYC[3]. The application is beneficial for both transport related organizations and individuals like taxi drivers who are looking for places with higher pickup chances and passengers who require fast pick-up and fare estimate. In this paper, they treated taxi as mobile sensor and then analyzed historical taxi trip dataset in NYC between 2009 and 2015 with big data tools to discover the traffic and travel pattern on time periods and the relationship between features including trip time, distance, and fare in the dataset. Based on the analytic result of the traffic patterns and features relationship, they are able to build a mobility-enabled application that produces useful trip data for its users. Our project is to predict the New York City Taxi Trip duration. The NYC Taxi and Limousines Commission including pick-up/drop-off time, geo-coordinates, number of passengers, is also one of our datasets. Sun et.al has explored the relationship between different features, which help us to select more relevant features to analyze. Also, in their result, they failed to discover the relationship between times and distance in the large trip which inspires us to introduce the weather, traffic data source as environmental factors to predict more accurate trip duration.

IV. Application Design

Our application first store the NYC yellow Taxi Data, NYC weather Data, NYC Traffic Data in HDFS. Then start to profile the attributes, remove the outline data and replace the missing value. Next we choose the attribute that have the predictive potential to trip duration. And we compute the trip duration from pick up time and drop off time attributes and treat it as label in Spark. To join these three dataset together we need to set the date time in the same format. After processing the data and get the final training data, we visualize the relationship between each features and the label. At last we train our data on linear regression algorithm and gradient boosted regression algorithm with Spark MLib to test which algorithm is better and find out the final features that have predictive potential to the trip duration. The design diagram is shown as Fig. 1. below.

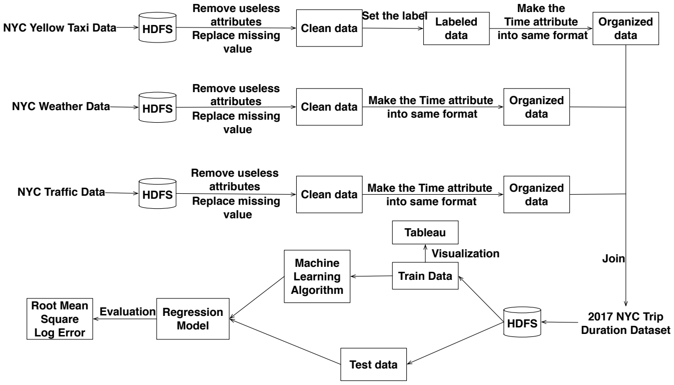


Fig. 1. Application Design Diagram

V. Datasets

This paper uses three datasets.

The first dataset is the NYC Yellow Cab trip record data, Jan-June 2016. The dataset is downloaded at http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml. It's about 20GB total. After profiling the data, we chose eight columns from the original dataset. They are Pickup date time as String, Passenger count as Int, Trip distance as Double, Pickup longitude as Double, Pickup latitude as Double, Pickup LinkID as Int, Dropoff LinkID as Int, Trip Duration as Int. We will talk about how we compute the pickup link id and drop off link id in the experiment section.

The second dataset is the NYC Real Time Traffic Speed Data Feed Archived, Jan-Jun 2016 (gathered from http://data.beta.nyc/dataset/nyc-real-time-traffic-speed-data-feed-archived), including the monthly traffic speed data and a LinkID to coordinates lookup table. This dataset is collected in batch (Five Minutes Intervals). Our extraction is about 400 MB in total. After preprocessing, we keep four columns including LinkId as String, Speed as Double, DateAsOf as String, and Borough as String.

The third dataset is the Local Climatological Data of NYC, central park, Jan-June 2016 (gathered from https://www.ncdc.noaa.gov/data-access/quick-links#loc-clim). It's about 3MB. After preprocessing the data, we chose four columns, Date time as Sting, Hourly visibility as Double, Hourly precipitation as Double, Hourly wind speed as Double.

After join these three datasets together (the specific join detail is shown at experiment section), the final dataset schema is shown as Table I below:

TABLE I
FINAL DATASET SCHEMA

Feature	Type	Description
Trip Duration (Label)	Int	Time duration o the trip (in seconds)
Passenger Count	Int	The number of passengers in the vehicle
Trip Distance	Double	The elapsed trip distance in miles reported by the taximeter
Pick up Loc Traffic Speed	Double	Traffic speed at pickup location at pickup time
Drop off Loc Traffic Speed	Double	Traffic speed at drop off location at pickup time
Visibility	Double	Visibility at pickup
Precipitation	Double	Hourly precipitation at pickup
Wind Speed	Double	Hourly wind Speed at pickup

VI. Remediation

Given the data on total trip distance, traffic speed in drop-off location and weather condition, our model can predict the taxi trip duration with a reasonable error. Therefore, our model can be used to build an application for taxi companies to provide trip duration estimate for their customers. This application has the potential to be automated if real time data collection technology is incorporated.

In the future, we need to train the model every year, because the pattern of weather information and traffic data various every year, so that in this way we can keep the miss prediction in a reasonable range and maintain the high practicability of our application.

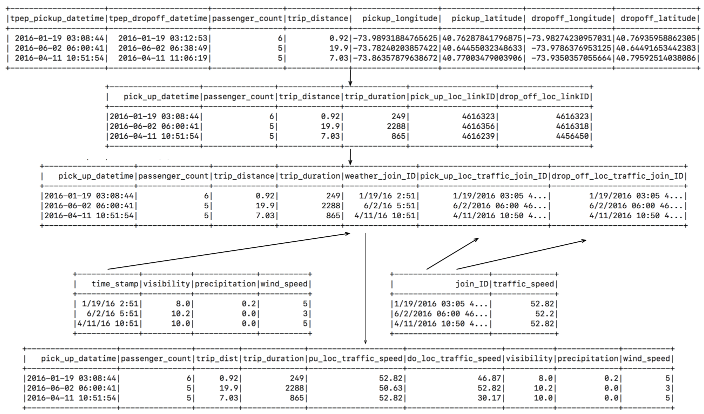
VII. Experiment

A. Data Source

The NYC taxi trip data are all made public on the NYC Taxi & Limousine Commission (TLC) website. The dataset includes trip records from all trips completed in yellow taxies in NYC of 2016. The dataset contains many useful information such as coordinates and time of pick-up and drop-off, trip duration, and so on. However, to predict the trip duration, these information is not enough, so we decide to add another two datasets. The first one is U.S. Local Climatological Data, from National Centers for Environmental Information, including the weather information of NYC whose attributes contain temperature, wind speed, precipitation and so on. The second one is NYC Real Time Traffic Speed Data in a five minutes interval, from data.beta.nyc., including the traffic information of about 100 sensors among NYC, whose attributes contain traffic speed, sensor id, and so on.

B. Data Profile and Filtering

The dataset contains many anomalies and potentially erroneous entries. This is common with such a large dataset. The data anomalies includes null value, outline value, wrong type value and so on. We encountered and solved many inconsistencies and anomalies during the analysis. Often, entries in the dataset must be filtered out in order to obtain a feasible and concrete analysis and relationship among the data. Some inconsistencies were found while running analytics jobs, causing another iteration of data cleaning and filtering to be performed.



```
+--------------------+---------------------+---------------+-------------+------------------+-----------------+------------------+-----------------+
|tpep_pickup_datetime|tpep_dropoff_datetime|passenger_count|trip_distance|  pickup_longitude|  pickup_latitude| dropoff_longitude| dropoff_latitude|
+--------------------+---------------------+---------------+-------------+------------------+-----------------+------------------+-----------------+
| 2016-01-19 03:08:44|  2016-01-19 03:12:53|              6|         0.92|-73.98931884765625|40.76287841796875|-73.98274230957031|40.76935958862305|
| 2016-06-02 06:00:41|  2016-06-02 06:38:49|              5|         19.9|-73.78240203857422|40.64455032348633| -73.9786376953125|40.64491653442383|
| 2016-04-11 10:51:54|  2016-04-11 11:06:19|              5|         7.03|-73.86357879638672|40.77003479003906| -73.9350357055664|40.79592514038086|
+--------------------+---------------------+---------------+-------------+------------------+-----------------+------------------+-----------------+
```

```
+-------------------+---------------+-------------+-------------+------------------+-------------------+
|   pick_up_datetime|passenger_count|trip_distance|trip_duration|pick_up_loc_linkID|drop_off_loc_linkID|
+-------------------+---------------+-------------+-------------+------------------+-------------------+
|2016-01-19 03:08:44|              6|         0.92|          249|           4616323|            4616323|
|2016-06-02 06:00:41|              5|         19.9|         2288|           4616356|            4616318|
|2016-04-11 10:51:54|              5|         7.03|          865|           4616239|            4456450|
+-------------------+---------------+-------------+-------------+------------------+-------------------+
```

```
+-------------------+---------------+-------------+-------------+---------------+---------------------------+----------------------------+
|   pick_up_datetime|passenger_count|trip_distance|trip_duration|weather_join_ID|pick_up_loc_traffic_join_ID|drop_off_loc_traffic_join_ID|
+-------------------+---------------+-------------+-------------+---------------+---------------------------+----------------------------+
|2016-01-19 03:08:44|              6|         0.92|          249|   1/19/16 2:51|       1/19/2016 03:05 4...|        1/19/2016 03:05 4...|
|2016-06-02 06:00:41|              5|         19.9|         2288|    6/2/16 5:51|       6/2/2016 06:00 46...|        6/2/2016 06:00 46...|
|2016-04-11 10:51:54|              5|         7.03|          865|  4/11/16 10:51|       4/11/2016 10:50 4...|        4/11/2016 10:50 4...|
+-------------------+---------------+-------------+-------------+---------------+---------------------------+----------------------------+
```

```
+-------------+----------+-------------+----------+
|   time_stamp|visibility|precipitation|wind_speed|
+-------------+----------+-------------+----------+
| 1/19/16 2:51|       8.0|          0.2|         5|
|  6/2/16 5:51|      10.2|          0.0|         3|
|4/11/16 10:51|      10.0|          0.0|         5|
+-------------+----------+-------------+----------+
```

```
+--------------------+-------------+
|             join_ID|traffic_speed|
+--------------------+-------------+
|1/19/2016 03:05 4...|        52.82|
|6/2/2016 06:00 46...|         52.2|
|4/11/2016 10:50 4...|        52.82|
+--------------------+-------------+
```

```
+-------------------+---------------+---------+-------------+--------------------+--------------------+----------+-------------+----------+
|   pick_up_datatime|passenger_count|trip_dist|trip_duration|pu_loc_traffic_speed|do_loc_traffic_speed|visibility|precipitation|wind_speed|
+-------------------+---------------+---------+-------------+--------------------+--------------------+----------+-------------+----------+
|2016-01-19 03:08:44|              6|     0.92|          249|               52.82|               46.87|       8.0|          0.2|         5|
|2016-06-02 06:00:41|              5|     19.9|         2288|               50.63|               52.82|      10.2|          0.0|         3|
|2016-04-11 10:51:54|              5|     7.03|          865|               52.82|               30.17|      10.0|          0.0|         5|
+-------------------+---------------+---------+-------------+--------------------+--------------------+----------+-------------+----------+
```

Fig. 2. Join Dataset Diagram

C. Join Datasets together

A big challenge in our project is to join our three datasets together. The diagram of joining the datasets together is shown in Fig. 2. To join the taxi data and weather data, we need to set the date time into same format and use it as key to join them together. Because the date time in weather data is in 10 minutes interval and all at 51 minutes of each hour, we set the date time in taxi data to the nearest 51 minutes before the pick-up time. To join the taxi data and traffic data together, we need to compute the nearest traffic sensor of each pick-up and drop-off location in the taxi dataset and assign the sensor id (named linkID in original dataset) to the each data. Because the traffic data is in a five minutes interval, we need to round up the date time in taxi dataset also to a five minutes interval, which means we need to transfer the date time to the nearest minutes ends with 5 or 0 before the pick up date time. After that we concatenate the rounded up pick-up date time of taxi data and nearest sensor id together and treat this as the key to join the traffic dataset. We both compute the traffic dataset join id of pick-up location and drop-off location, to get the pick-up location traffic speed and drop-off location traffic speed.

After join the datasets together, we visualize some of the attributes (as in Fig. 3, Fig. 4 and Fig. 5) to find the potential relationship between features and label (trip duration) as in Fig. 6.

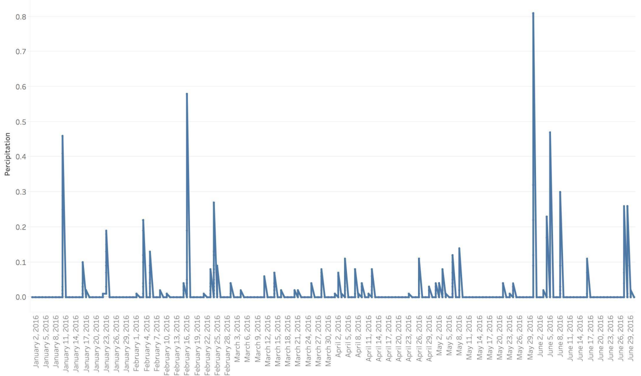


Fig. 3. Precipitation Distribution

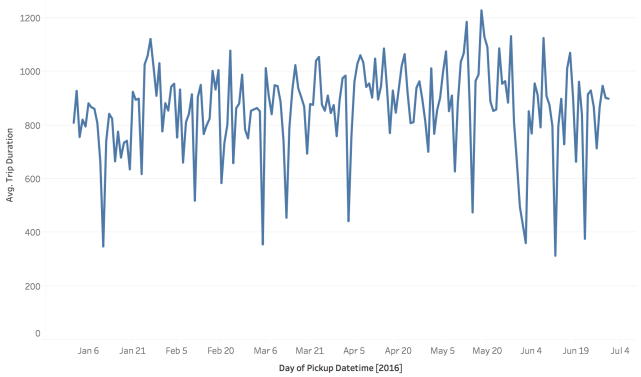


Fig. 4. Trip Duration Distribution

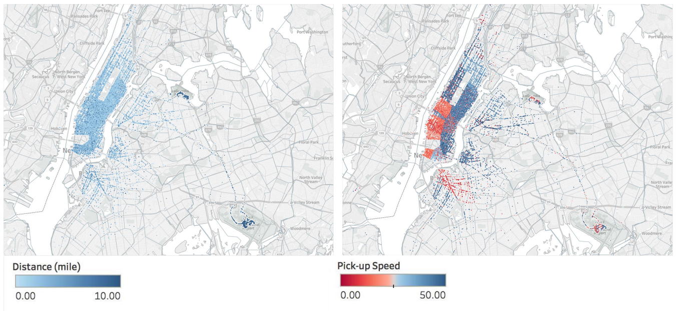


Fig. 5. Trip Distance & Trafffuc Speed Distribution

D. Training Model

We deployed two machine learning algorithm. The first one is linear regression. After visualize the relationship between each attributes and label (trip duration), which is shown in Fig. 6, we find out there is a linear relation between trip distance and trip duration, so we try the linear regression first. It gives us a relatively good result with root mean square log error 0.3445197221401201.

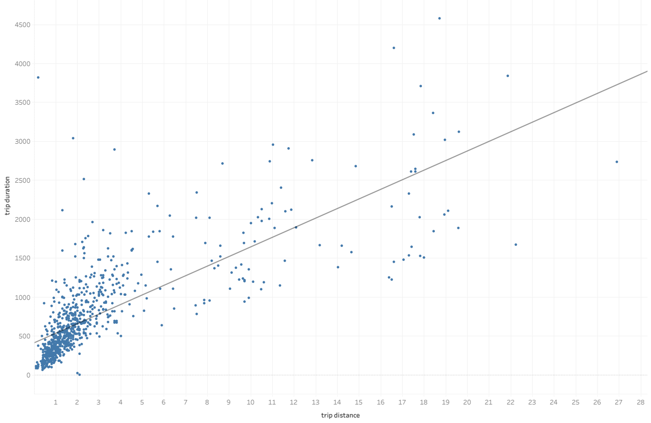


Fig. 6. Relation between Trip Distance & Duration

The second one is gradient boosted trees. Compared to linear regression, gradient boosted regression is faster, robust, and have a readable output (,which is several weighted decision trees) but sensitive to noise and extreme value and more memory consuming. Gradient-Boosted Trees (GBTs) are ensembles of decision trees. GBTs iteratively train decision trees in order to minimize a loss function. GBTs handle categorical features, do not require feature scaling, and are able to capture non-linearities and feature interactions. Gradient boosting iteratively trains a sequence of decision trees. On each iteration, the algorithm uses the current ensemble to predict the label of each training instance and then compares the prediction with the true label. The dataset is re-labeled to put more emphasis on training instances with poor predictions. Thus, in the next iteration, the decision tree will help correct for previous mistakes. With these advantages, gradient boosted regression gives us a really good result with with root mean square log error 0.15093662004590197.

E. Evaluation method

we use root mean square log error (RMSLE) instead of root mean square error (RMSE) to evaluate our model.

$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2}$$

RMSLE measures the ratio between actual and predicted. The RMSLE was used because we don't want to penalize huge differences when both the values are huge numbers and we want to penalize under estimates more than over estimates. Lets have a look at the below example.

Case a) : Pi = 600, Ai = 1000; RMSE = 400, RMSLE = 0.5108

Case b) : Pi = 1400, Ai = 1000; RMSE = 400, RMSLE =

0.3365

As it is evident, the differences are same between actual and predicted in both the cases. RMSE treated them equally however RMSLE penalized the under estimate more than over estimate.

VIII. Conclusion

The prediction results of linear regression model and gradient boosted model are show below in Fig. 7. and Fig. 8. The RMSLE of linear regression is about 0.3445, and the RMSLE of gradient boosted regression is about 0.1509. In Fig. 7. the orange line represents real label, the red line represent linear regression predict label, the blue line represent gradient boosted regression predict label. We can clearly see that blue line is closer to the orange line than the red line, and most of the blue line is higher than orange line, but red line is lower than orange line, which means the miss-prediction of gradient boosted regression tends to be over estimation but the miss-prediction of linear regression tends to be under estimation, so gradient boosted regression model get the better performance.

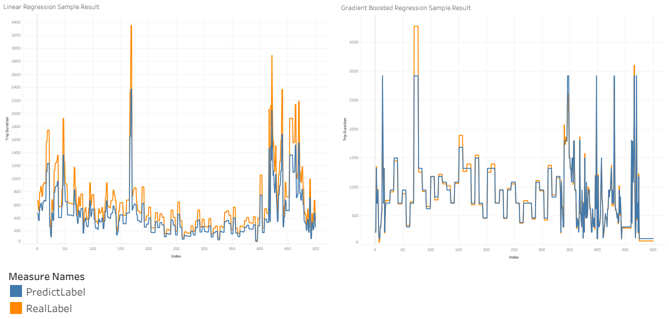


Fig. 7. Linear Regression & Gradient Boosted Regression Result

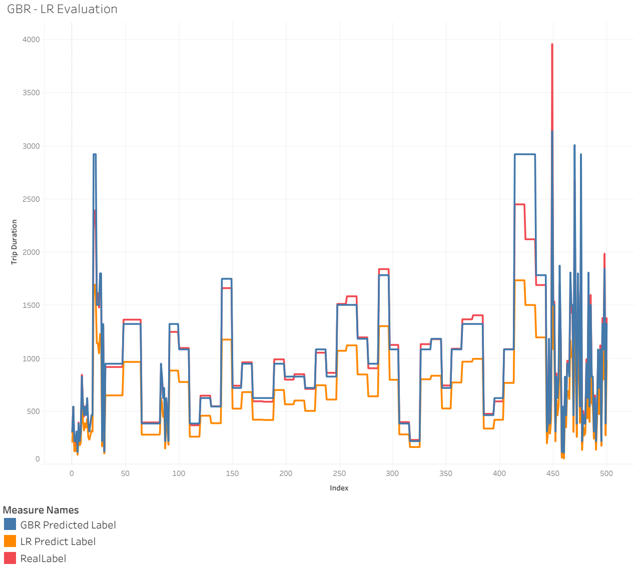


Fig. 8. Gradient Boosted Regression Predicted Trip Duration & Real Trip Duration

IX. Future work

Overall, we could predict the taxi trip duration in a decent error with the gradient boosted regression model after introducing two external datasets: weather and real-time traffic speed. In the future, we plan to improve our model by including more functionalities such as fare prediction and route recommendation. Moreover, we plan to build an application that can predict real-time taxi duration. Once the passenger enters the place to go, the application will give the prediction immediately based on real-time weather and traffic data. This would require applying new technologies such as Kafka, Spark Streaming, which can process real-time data.

X. Acknowledgment

We thank Professor McIntosh and TA Priyanka Vaidya for providing consistent help during the whole project. We are also grateful to NYU high performance computing group. In addition, we would like to express our gratitude to New York Taxi and Limousine Service , National Centers for Environmental Information and Beta.NYC for publishing the data.

References

[1] Balan, Rajesh Krishna, Khoa Xuan Nguyen, and Lingxiao Jiang. ”Real-time trip information service for a large taxi fleet.” Proceedings of the 9th international conference on Mobile systems, applications, and services. ACM, 2011.

[2] Himanshu Jaiwal, Tushar Bansal, Prateek Jakate, Tejas Saxena. ”NYC Taxi Rides: Fare and Duration Prediction.” UCSD CSE 258: Web Mining and Recommender Systems, Winter 2017.

[3] Sun, Huiyu, and Suzanne McIntosh. ”Big data mobile services for New York city taxi riders and drivers.” Mobile Services (MS), 2016 IEEE International Conference on. IEEE, 2016.

[4] NYC Yellow Cab Trip Record Data: http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

[5] NYC Real Time Traffic Speed Data: http://data.beta.nyc/dataset/nyc-real-time-traffic-speed-data-feed-archived

[6] the Local Climatological Data of NYC: https://www.ncdc.noaa.gov/data-access/quick-links#loc-clim