Travel Duration Prediction of Taxi and Bike Trips in New York City

Weiqiang Li, Liheng Gong

***Abstract*—**

Is it possible to predict the trip duration of taxies and bikes based on weather and trip parameters in New York City? To study this problem, we collected the weather data, green and yellow taxi trip data, and CitiBike trip data in New York City, and applied machine learning algorithms to predict the trip duration. In this paper, three machine learning algorithms in Spark MLlib - Linear Regression, Decision Tree and Random Forest was used and analyzed; the performance of these three algorithms are also compared. The features used for our predictions are carefully selected and constructed, and k-means clustering was used to improve our prediction precision by clustering the starting coordinates and the ending coordinates to an optimized series of centers. Trip distances were also estimated based on starting and ending coordinates for our prediction model, to further improve the accuracy. The mean square error (MSE) was used to evaluate our algorithms, and the errors were dropped by over 50% compared to the prediction using profiled data directly. Our algorithm and model gave a very good estimation of trip duration. Finally, an application written in Scala was built with a unified interface for users to easily use our model for their own predictions.

Keywords—big data analytics, linear regression, decision tree, random forest, trip duration prediction, k-means clustering, machine learning, feature extraction, feature construction, Scala, Spark, MLlib

I. INTRODUCTION

Can we predict the trip duration of taxi and bike trips in New York City? Which features have the greatest impact upon the trip duration? To study this problem, we collected the weather data, taxi trip data, and CitiBike trip data in New York City and utilized machine learning algorithms with several optimizations to estimate the trip duration in New York City. All the data was put in HDFS for our analyzation and application programming, using Spark with Scala. Weather data was joined to the trip data in order to find out how much effect weather conditions will cause to the trip durations. Mainly three machine learning regression algorithms in Spark MLlib were used - Linear Regression, Decision Tree and Random Forest, with several features carefully selected and constructed based on the profiled data. Trip distances were calculated based upon the coordinates provided by the datasets, by converting the coordinates to cartesian coordinates and calculating the Euclidean distances, to be provided in our prediction model for better accuracy; k-means clustering was also used to improve the prediction precision by clustering the starting point and ending point coordinates to a reasonable number of cluster centers. The performance of these three machine learning algorithms are evaluated and compared, and our final model was proved to reduce the mean square error (MSE) by over 50%. For the convenience of users to use our model, an application with unified interface was built, so users can use our optimized algorithm for training and testing, based upon their own selection of data and machine learning algorithm.

II. MOTIVATION

For travel methods like taxis and bikes, we study whether we can use features like location coordinates, time and weather conditions to predict trip time. We also analyze which features have the most impact on the prediction of trip durations. If the predictions are reasonable, then we can use historical trip and weather data to make reliable predictions of trip duration if location, time, weather and other required features are provided. We can also predict future trips based upon the starting and ending points together with weather forecast and other important features. Commuters using taxis and bikes can utilize the predictions to better schedule commute plans and taxi/bike service providers can also use the prediction to optimize taxi/bike dispatching and improve service quality.

III. RELATED WORK

There are many previous studies and works related to trip duration predictions and trip pattern recognitions. Summaries of four related studies are presented below.

Study [2] and [3] focus more on finding the pattern through deep analyzation of trip data, and try to find interesting and useful conclusions. In study [2], the paper uses the GPS trajectory data of two subjects over 8 months to analyze the travel behavior on different seasons and weather conditions at Hakodate city, Hokkaido, Japan. Study [2] also uses random forest method to predict the transportation mode based upon several attributes.

Hakodate city is a port city located in south-west Hokkaido. Unlike most part of Hokkaido, Hakodate city is warmer in the winter with lesser snowfall. This city, with four distinct seasons, is suitable for analyzing the travel behavior of people under different seasons and weather conditions. To do this, a Probe Person Travel Survey (PP survey) was conducted for four months in both summer and winter 2013. A smartphone application was designed to ask for the trip purpose, destination and transportation mode from subjects when they travel, and began to collect GPS coordinate data every 30 seconds during the trips. The two subjects are a 60-year-old male (subject A) and a 40-year-old female (subject B).

After all the collections, the data were cleaned, filtered and manually labeled, to make the “ground truth” for training data.

The destinations and travel pattens of two subjects were studied first. Subject A has 4 workplaces with different visit frequencies, and he usually visits fitness club. Subject B has only one workspace, and she usually goes to supermarkets. Then the impacts of different seasons were analyzed. After calculating the average number of trips of different kinds during summer and winter, it can be derived that the travel behavior of both subjects changed in accordance with the change of season. Subject A did not change the frequency to go to fitness club during the winter, but did decrease the frequency for commute greatly and increase the frequency for shopping, meal and recreation. Subject B has a more steady life, with no significant change in frequencies during different seasons. Both subjects travel a little less during winter than during summer.

As for the transportation mode in summer and winter, 43.5% of the trips were by bicycle in summer for subject A, but this percentage dropped to 0% during winter. On the contrary, the percentage of walking increased significantly during winter. This is because of the road and weather condition in Hakodate city, which makes it impossible to ride during winter. The trips via bus and trams did not change greatly, and the delay of buses during winter did not change the behavior of subject A. As for subject B, since she almost always used her car for trips, the season did not change her transportation mode. The impacts from weather on travel behavior were also studied. Precipitation does influence the travel behavior, in that the number of average trips decreases for subjects A during rainy days or snowy days. However the exercise activities were not influenced.

Next, random forest method was used to predict the transportation mode of subjects based upon the training data and test data. 17 different attributes were selected, with 12 numeric attributes and 5 descriptive attributes. Random forest method not only has the satisfactory accuracy, but also reports the importance of every attribute, so we can understand which attribute contributes most to the identification of transportation mode. Several training set and the corresponding test set were generated and studied. The accuracies are not satisfactory when only using data from one season to predict the other season. When using the combined data for both seasons as training set, the accuracies on test set are stable at around 86% - 89%, and it behaves similarly on both test data from single season and test data from mixed seasons. As for the importance of the attributes, random forest method suggests the average speed during the trip as the most important attribute when identifying the transportation mode. Other important attributes include distance from trip end to work place, trip distance, time spent during the trip and distance from trip end to home. Weather and season do not influence significantly on the algorithm here.

In summary, study [2] concludes that there are changes in travel behavior when season changes or when weather changes. In winter, the average trip counts drop for both subjects, and the travel pattern (including transportation mode) is very different for subject A. When facing rainy or snowy days, the average trip counts also drop for subject A. Random forest model shows satisfactory accuracies when predicting transportation mode using combined training data set for both seasons, and the average speed during the trip is the most important attribute during the prediction when using this model.

Study [3] explores a large-scale dataset containing information of bike sharing customers and trips in Nanjing, China. The data was collected via the smart card system in which the bike sharing customers are required to use smart cards for the bikes. The datasets were provided by Transportation Authority of Jiangning District, containing bike sharing customers information, bike sharing managers information, and bike sharing trips information. They are collected from Sept 1 to Oct 31 in 2012, including 101,709 trip records and 152 docking station records. The trip records data were preprocessed to filter out those trips with significantly short and long travel durations, and only the trips longer than 2 min and shorter than 120 min were considered.

Bike sharing trip chains (BTC) were defined in order to distinguish between different patterns of bike sharing trips. The following symbols are used to describe trip BTCs: “O” stands for the starting station, “D” stands for a different ending position, in which “sD” stands for single destination, and “mD” stands for multiple destinations. In this way, some typical BTCs can be easily described: “O-O” means a trip starts from and ends at the same station, “O-D” means a trip starts from one station and ends at a different station, and it can be further divided into two groups: “O-sD” and “O-mD”; “O-sD-O” means a trip starts from one station, passes through a single destination and ends with the station same as the origin; “O-mD-O” means a trip starts from one station, passes through multiple destinations and ends with the station same with origin. Given this, it is found that among all the 53,537 valid trip records, 59.87% of them are O-D trips, indicating that O-D is the most common trip pattern. 7.21% of them are O-O trips, indicating that some customers may use bike sharing system as a method for exercise. 28.38% of them are O-sD-O trips, and only 4.54% of them are O-mD-O trips.

Among all those BTCs, z-score technique is used to evaluate the differences of bike sharing trip time quantitatively. There are two different topics explored: z-score regarding gender and z-score regarding weekdays and weekends. To avoid any ambiguity in statistics, only BTCs containing more than 30 trips are considered, and the threshold for significant difference between two groups is defined as 1.96. The difference will be considered big if the z-score is larger than this number. Also, the cross-tabulations using the chi-statistics are applied to find the differences of trips between gender and day of the week, classified by the BTCs. Finally, visual analytics technique is used to offer a intuitive feeling of the BTCs.

The overall datasets are analyzed first. The men and women shows almost identical cumulative possibilities regarding bike share travel time. As for the frequencies of using bike sharing services regarding the time of the day, men and women do not show obvious difference neither. However, 45% of the trips happens between 6-9 am and 4-7 pm on weekends, and the corresponding percentage on weekdays is 65%. The percentage on weekends is way more than that in London according to a previous investigation. This is because there are more people working on weekends in Nanjing than in London.

The travel time of different BTCs are evaluated next. O-O trips and O-sD trips are selected in this analysis. Men and women typically behave the same in the O-O trip chain, as the z-score is only 0.11; but they show significant difference in O-

sD trip chain, with z-score of 3.29, where women travel much longer time. This is because women cycle slower than men, and this is consistent with previous studies. As for the day of the week, the z-score is both big for O-O (4.09) and O-sD (6.71), indicating people cycle slower in the weekends than in weekdays.

Next, the different BTCs are fully evaluated and analyzed regarding gender and day of the week. Men tend to have more O-O trips but less O-sD-O trips. This probably means men are more likely to perform exercise on bike sharing systems, and women tend to use the same travel mode. Women also tend to have more O-mD-O trips, indicating that women is more likely to finish multiple tasks within a single ride. Women are also more likely to perform multiple circle BTCs. This is consistent with the studies that women usually play a more important role in family responsibilities. As for the day of the week, O-O trips tend to happen more on weekends because people have more time for exercises and leisures. O-sD-O trips are less but the O-mD-O trips are more, indicating that on weekends, bike sharing customers somewhat have more needs to finish multiple tasks within a single ride.

Finally, the trips related to subway stations are studied in detail, in order to offer better suggestions for the dataset supplier to provide better service. About 40% of the O-sD trips on weekdays are starting from residential areas and ending at subway stations, which is much more than the trips from the opposite direction. This unbalanced behavior of BTCs is because it is very hard to find a bike near the subway stations, especially during peek hours. Weekday trips are more likely to be related to subway stations than weekend trips. As the BTCs related to subway stations are the most important trips of all, the bike sharing storage problem is a major problem the service provider need to solve. Using the manager's dataset, it can be seen that the service provider has made some effort to balance the storage of the bike sharing stations. 33% of the balancing are moving bikes from other stations to the stations near the subway stations, and 62% of them happen between the peak hours, from 4 pm to 7 pm. Although these efforts were made, the supply and the demand are still seriously unbalanced. Further improvements should be made based on more detailed analysis, to precisely know the bike sharing demand and arrange the balance behaviors accordingly. Also, providing more stations with more bikes around subway stations might be able to relieve this problem.

In summary, study [3] analyzed the different bike sharing trip chains, using z-score technique and cross-tabulation with chi-statistic technique, to show the difference in travel behaviors regarding gender and day of the week. Visual analytics is used as an aid for intuitive feelings of different BTCs and the corresponding changes. Men and women are similar in some trip patterns but very different in others. The day in the week also influence significantly in the trip patterns. The analysis on trips related to subway stations are also analyzed in detail, to further study the unbalanced supply and demand of bikes around these subway stations, and to help the service provider to offer better planning on bike storage regarding this issue.

Study [4] and [5] focus more on the analyzation and exploration of NYC trip data. In study [4], the authors analyzed the relationship between New York City Taxi Data (hourly number of taxi drop-offs) and four feature datasets: NYC Point-of-Interest (POI) dataset which is retrieved from FourSquare API; NYC Geo-Tagged Tweets for Local Events; NYC daily Weather dataset from National Center for Environmental Information; NYC Vehicle collision dataset.

While modeling the correlation between traffic and the four feature data sets, the authors used L2-regularization to avoid overfitting of linear regression. The authors also adopted kernel ridge regression model to address the problem that two feature data sets might be correlated with each other (e.g. weather feature might be correlated to the POI feature).

The authors used both quantitative and qualitative analysis in the paper. In their quantitative analysis, they found that among all the 4 feature datasets, POI feature correlates best with the taxi traffic data. Adding other feature datasets does not lead to significant improvement. The authors speculated that the reason is that traffic follows routine behavior, which can be captured by the POI feature dataset. In their qualitative analysis, they found that some local events, the information of which is extracted from Geo-tagged tweets, can help explain some traffic patterns. They also found that extreme weather conditions do impact traffic. The conclusion of [4] is that multiple feature datasets can help explain the traffic data pattern in NYC.

Study [5] also investigates the importance of data cleaning during data exploration process. Traditionally, data cleaning is performed before analysis process. The authors in this paper argue that data cleaning process should be an integral part of urban data exploration and analysis.

Using NYC taxi data as the study case, the authors identify multiple problems that contribute to the anomalies and outliers in the taxi data set and discuss various methods to address these problems.

The first problem is that some seemingly erroneous data points may embed uncovered features that can explain important phenomena. For example, for daily taxi trips in NYC, large drops are observed in August 2011 and October 2012. Those drops might be treated as corrupt data and would be cleaned, but if the daily taxi trips data is combined with weather data, those large drops in taxi trips can be explained by extreme weather conditions.

The authors proposed three approaches to address this problem. The first is to go beyond a single dataset and combining multiple datasets to seek explanations for anomalies in a specific dataset. The second approach is to aggregate data into different spatial and temporal resolutions and granularities to identify and pinpoint dirty data. One benefit of examining different data slices at different resolutions is that it helps determine the quality of the data. The authors also point out that finding the minimum and maximum in different spatio-temporal slices might help detect events and dirty data in urban datasets.

The second problem is that data providers might utilize inconsistent data cleaning methodologies between different datasets. The authors discovered in NYC taxi trip data in years around 2010, there are negative values in trip distance, which clearly indicates illegal data. Some values such as -21,474,834.00 are apparently caused by data overflow. In contrast, in years around 2012, there are no negative values for trip distance. The authors speculate that the data provider might

adopt some improvement methods to clean their data. To address such kind of inconsistency problems, the authors argue that including provenance information for data and detailing the cleaning operations applied to datasets are essential to ensure that analyses across different datasets are consistent. Another benefit of accessing to the provenance of the cleaning process is that the same improvement techniques for data cleaning can be re-used to older datasets.

The third problem is that some data acquisition methods are not reliable. For example, for NYC taxi trip data, GPS readings are not always accurate because GPS data is affected by multiple factors: tall buildings might block satellite signal; the quality of GPS receiver algorithm for processing the satellite signals might also affect the accuracy of the data. To address such kind of problems, the authors suggest that clustering methods can be used. For GPS data, if the geographical boundaries are known in advance, data can be clustered in accordance with these boundaries.

Conclusion of [5]: Data cleaning should not be treated as a pre-processing task and should be applied on the fly during data exploration process, and through data exploration, users can attain a better understanding of the data, which can lead to the discovery of better cleaning constrains and strategies, which in turn enables users to discern better between errors and features.

These studies not only show the importance of data preprocessing and analyzation, but also indicates that the right choice of features and analyzation methods are very important. Also, useful conclusions also derive from the exploration of different aspects of the same dataset or through different datasets. They all provide us with generalized methods and approaches for handling trip data, and also illustrate the importance in choosing the correct path in analyzation of big data.

IV. DESIGN

The weather data, taxi trips data and CitiBike trips data were downloaded and collected from Internet and uploaded to HDFS. The New York City weather data was downloaded from the NOAA Online Weather Data website, containing the temperature and climate information collected from NYC Central Park Station. The New York City Green Taxi and Yellow Taxi trip data were collected from NYC Taxi and Limousine Commission website, containing detailed records of taxi trips in New York City, including trip pick-up and drop-off dates, times and coordinates, number of passengers, trip distance and taxi fees, etc. The New York City CitiBike trip data was downloaded from CitiBike System Data website, containing detailed records of CitiBike trip histories, including start/end dates and times, station names, station coordinates, trip durations, user birth year and gender, etc.

Spark and Spark SQL were used to load, clean and profile the raw data. Only a selected portion of columns of the raw data were extracted, in which those features were considered useful in the upcoming predictions. The profiled weather data and trip data were joined based on the date, to provide temperature and climate information in each prediction. Then feature selection and construction were conducted, in order to transform the value in the columns to decimal numbers for regression algorithms, and for better performance of our prediction precision. Trip distance calculation and K-means clustering are the most important part in this process. To calculate the trip distance, the coordinates of starting and ending points were first converted to cartesian coordinates, and the Euclidean distance was calculated based upon the cartesian coordinates. As for k-means clustering, it was used to cluster the location coordinates into a series of cluster centers, and assign the starting coordinates and ending coordinates to those centers for each trip record. Future experiments prove that this process will greatly improve our accuracy for predictions. After feature selection and construction, machine learning regression algorithms in Spark MLlib were used to analyze the data set. We mainly trained Linear Regression, Decision Tree and Random Forest regression models to do the analytics, and mean square error (MSE) was used to evaluate every model for each kind of trip. These processes are shown below in Figure 1 and Figure 2.

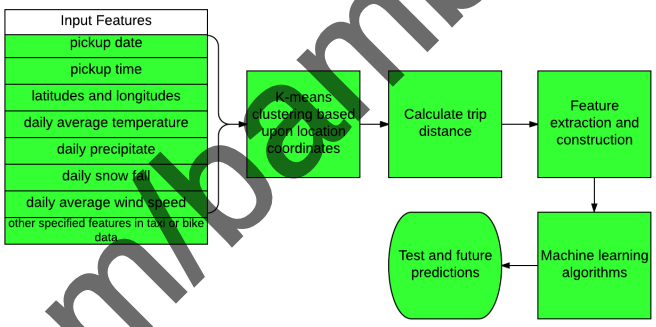

Figure 1 | Feature Selection and Construction with K-means Clustering in the Model Training Process

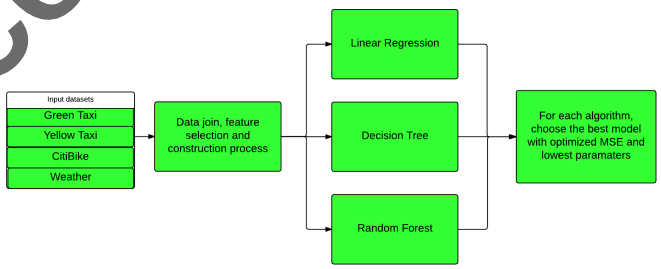

Figure 2 | Regression Machine Learning Algorithms Used and the Performance Evaluation

The following features were used in our weather data: date, temperature, daily precipitate, daily snow fall, and daily average wind speed. The trip duration feature was used as ground truth for our training for both taxi and bike trip data. The features used in our profiled taxi trip data were date, time, pick-up coordinates, drop-off coordinates, and passenger number. The features used in our profiled bike trip data were date, time, starting station ID and coordinates, ending station ID and coordinates, birth year of user, and gender of user.

There are two modes in our application. For each data set, the training mode will train a series of regressors with different parameters and choose the model with best precision (lowest MSE) and save it on HDFS. It will also save the k-means clustering model using the best number of clusters to HDFS. All those processes are automatically finished and the user only needs to specify the input data directories, output model directories, and the algorithm that the user wishes to use. In the training mode, after feature selection and construction, the joined data set will be split into two parts: the training set

(70%) and the test set (30%). The user-defined machine learning algorithm will be used to train the model based on the training set and use the test set to evaluate the model to get the best one with lowest MSE. Then the best model will be saved. This process guarantees that the user will get the best model with highest accuracy, and the only thing the user needs to do is to choose the algorithm (between Linear Regression, Decision Tree and Random Forest). The best models of these three algorithms were also compared, but we gave the maximum freedom to the users to choose the preferred algorithm.

As for the test mode, the profiled and joined data with ground truth was applied as inputs. The user needs to specify the input data directories and the saved model directories. In this mode, no model was trained, and the program only uses the model previously saved on HDFS to output the MSE. Using these two modes, users can easily train the best model with a small portion of the whole data and test the model upon the whole data, to save the computing time without losing accuracy. Moreover, the test mode can be easily modified to make a prediction mode, to output predictions based upon the weather and trip parameters. The diagram for these two modes are shown below in Figure 3.

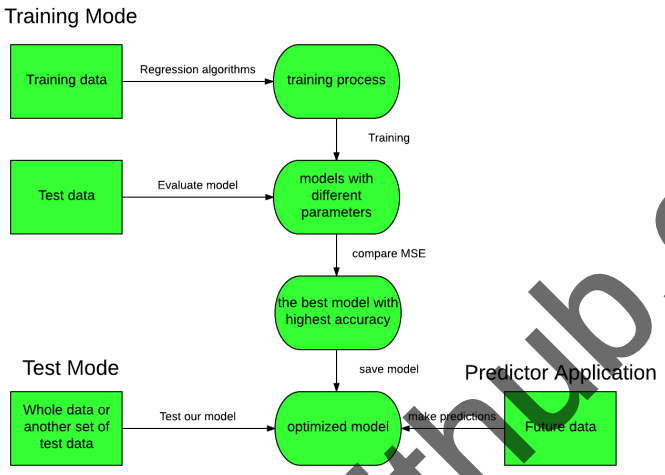

Figure 3 | Training Mode and Test Mode of Our Application

To conclude, our whole design diagram is shown below in Figure 4.

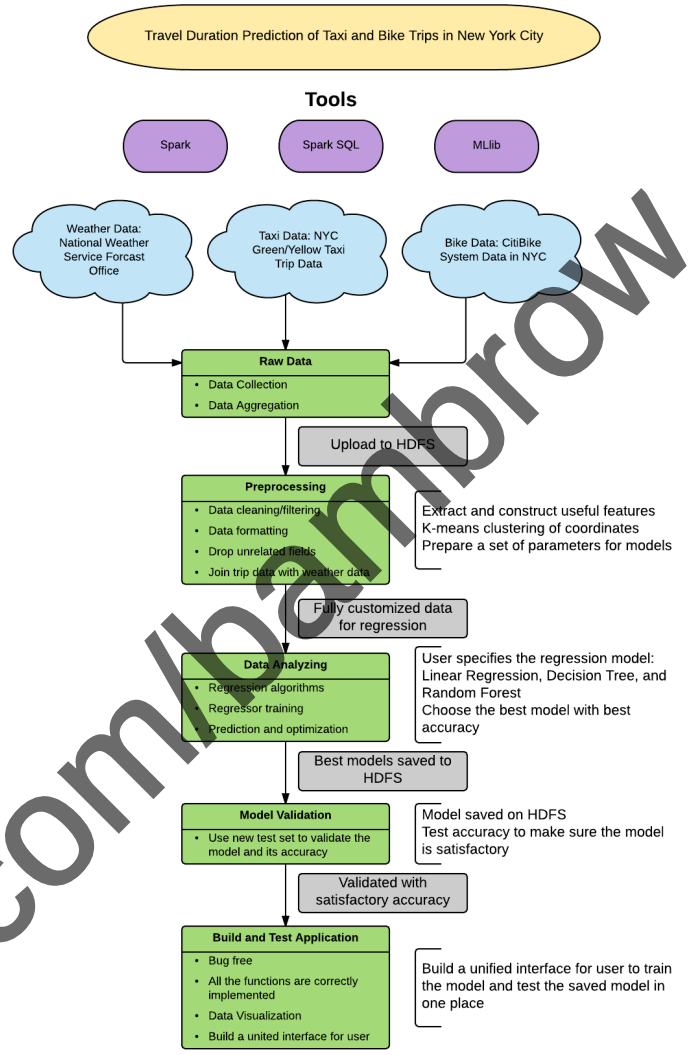

Figure 4 | Our Whole Design Diagram

V. EXPERIMENTS

There are two taxi providers in New York City. Green Taxi mainly operates outside Manhattan and Yellow Taxi mainly operates in Manhattan. The plots of sample Green Taxi pick up locations and sample Yellow Taxi pick up locations are shown below in Figure 5. In these plots, it is very clear that the Green Taxi only operate in northern Manhattan, from north of West 110th street and East 96th street.

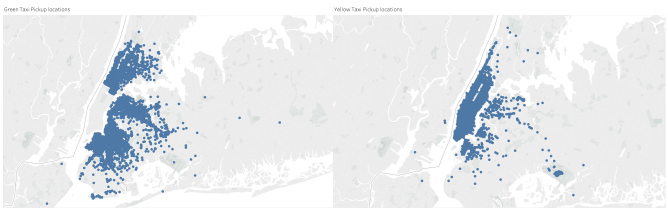

Figure 5 | The Service Area of Green Taxi (left) and Yellow Taxi (right)

Different from the taxi datas, the CitiBike provides services only at their specified bike locations. The plot of sample CitiBike locations are shown below in Figure 6.

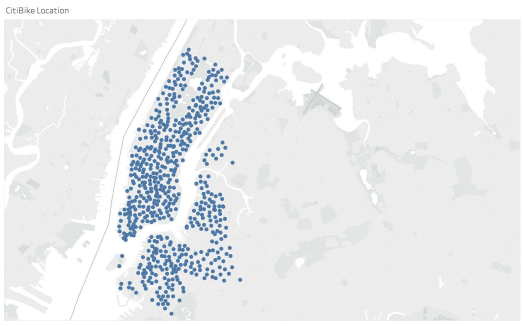

Figure 6 | The Bike Stations of CitiBike

In the experiment, the data from Green Taxi and Yellow Taxi were joined for our prediction algorithms. Since the service areas of green taxis and yellow taxis basically do not overlap, meaning that the data set of Green Taxi and Yellow Taxi are independent, which makes the joining of these two data sets reasonable. The joining of these data was performed in advance in the data profiling process.

The profiled trip data will firstly be used to join the weather data using the dates as keys. After joining, each data record has four more features: daily average temperature, daily precipitate, daily snow fall and daily average wind speed. Then the new features will be constructed.

First and most importantly, all the coordinates of the dataset will be applied to k-means clustering. Our k-means clustering will try to find the best number of cluster centers in the following ways: first it will try every possible cluster number from 5 to 12, and use kmeans++ strategy to cluster all the coordinates; after calculation and optimization of each number of clusters, it will calculate the Within Set Sum of Squared Errors (WSSSE); finally, it will choose the cluster number with the elbow point of WSSSE. The elbow point is estimated based on the calculation of absolute second derivative of each point: for every point, the absolute second derivative is (#former + #next - 2 * #current), where #former is the WSSSE of the last clustering, #next is the WSSSE of the next clustering, and #current is the WSSSE of the current clustering. The algorithm will find the highest absolute second derivative, and the number of clusters corresponding to this highest absolute second derivative is used as the best k. Then the corresponding model will be saved on HDFS for future test and predictions, and the starting coordinates and ending coordinates were both assigned to a cluster center as two features adding to our data records.

Besides k-means clustering, other features were also constructed and transformed. The trip distance was the most important one. It was calculated via the coordinates of starting coordinates and ending coordinates. Firstly, these coordinates will be converted to cartesian coordinates representing by x, y and z. Then, the Euclidean distance will be calculated simply adding the square difference in the three dimensions. This trip distance is of course not the actual trip distance, it is only the straight line distance between those two points. However, it will be useful in our machine learning algorithm because it reflects the tendency of the actual trip distance. The higher the actual trip distance, the higher our estimated trip distance should be, because the two coordinates should be further. Also, the trip distance is corresponding to the actual trip duration. Therefore, this estimated calculation is necessary and will be extremely helpful. The distance feature will be added to the records in the unit of kilometers.

Other features that are constructed and converted are for better prediction and usage of the upcoming regressor training. The date will be converted to a number between 0 and 1 to reflect the day of the year on that day, and is calculated via (current day of the year / total days of this year). The time will be converted to a number between 0 and 1 to reflect the number of seconds on that day, similarly to the conversion of date, and is calculated via (number of seconds of the current time / 86400). One new feature constructed is the weekday feature, which reflects whether the current day is a weekday. We assume that during weekdays, the trip duration will be slightly higher for same starting location and destination, because the traffic condition will be worse in weekdays. The value of this feature is either 0 or 1, in which 1 represents the current day is weekday, and 0 indicates the current day is not a weekday. Similarly, another new feature that will be constructed is the peak hour, indicating whether the current time is within the peak hour period. The value is also either 0 or 1, and the peak hour is defined as 6 AM to 10 AM and 4 PM to 8 PM only on weekdays.

After all these joining and feature extraction/construction processes, our data will be ready for the regression algorithms.

For machine learning algorithms, Linear Regression is firstly used for our regression. Linear regression will take the features as variance vector and find a linear function with a list of parameters to fit the value to be predicted, and try to minimize the lost and/or the model complexity. The loss function used is the squared loss function. The main parameters that should be specified in this model are number of iterations (numInterations) and step size (stepSize). Different set of parameters are used in our study, and the optimized parameters for taxi trip data are: number of iterations = 25, step size = 1.0E-7, MSE = 166183, with a square root of MSE of 407s. The optimized parameters for CitiBike trip data are: number of iterations = 100, step size = 1.0E-6, MSE = 103504, with a square root of MSE of 321s. The error distribution, calculated as the difference of actual trip duration and the predicted trip duration, are plotted below in Figure 7 and Figure 8. As we can see, the errors for taxi data are mainly between -500s and 500s, which is around an 8-9 minute period. The errors for bike data are lower, mainly between -200s and 200s range, which is within 4 minutes. Linear regression gives a better estimation of bike data compared to taxi data.

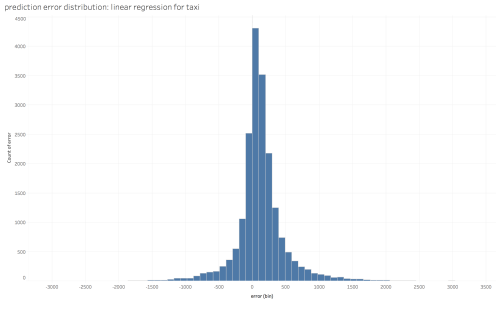

Figure 7 | Error Distribution of Taxi Data, Linear Regression

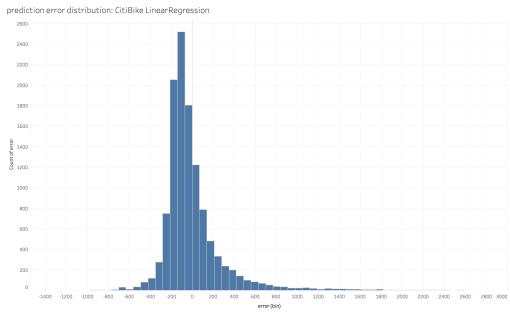

Figure 8 | Error Distribution of CitiBike Data, Linear Regression

Decision Tree is secondly used for our regression. Decision tree will split the training data into several subsets based on an indicator, and continue to do the same for the subsets recursively, and stops when the labels are the same in a subtree. The variance is used to evaluate the node impurity, which is the measure of the homogeneity of the labels in one node. The main parameters that should be specified in this model are max depth of the tree (maxDepth) and max bins that should be split into (maxBins). Different set of parameters are used in our study, and the optimized parameters for taxi trip data are max depth = 8, max bins = 48, MSE = 79335, with a square root of MSE of 281s. The optimized parameters for CitiBike trip data are max depth = 4, max bins 48, MSE = 95883, with a square root of MSE of 309s. The error distribution, calculated as the difference of actual trip duration and the predicted trip duration, are plotted below in Figure 9 and Figure 10. As we can see, the errors for taxi data are mainly between -400s and 400s, which is around a 6-7 minute period. The errors for bike data are basically the same, mainly between -400s and 400s range. Decision Tree gives lower MSE for both taxi and bike trips, which will give the better estimations.

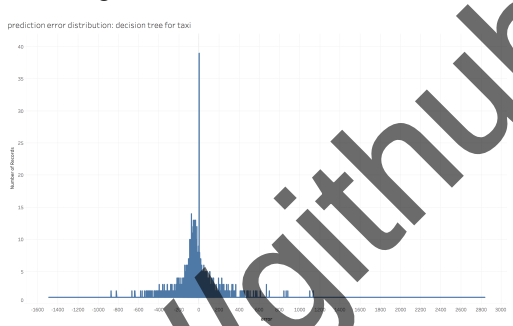

Figure 9 | Error Distribution of Taxi Data, Decision Tree

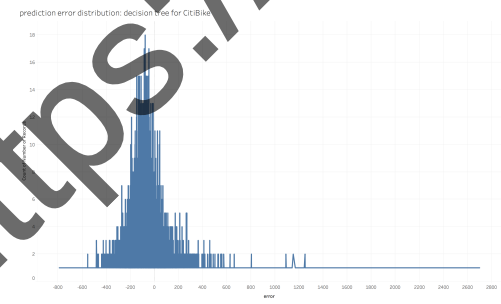

Figure 10 | Error Distribution of CitiBike Data, Decision Tree

Random Forest is the third algorithm used for our regression. It is an ensemble algorithm based on decision trees, trying to use multiple irrelevant decision trees for prediction, to minimize the overfitting problem. The predictions of every decision tree will be combined in a way that the variance of labelling can be reduced. The main parameters that should be specified in this model are max depth of the tree (maxDepth), max bins that should be split into (maxBins), and the number of trees (numTrees). Different set of parameters are used in our study, and the optimized parameters for taxi trip data are max depth = 16, max bins = 256, number of trees = 64, MSE = 69830, with a square root of MSE of 264s. The optimized parameters for CitiBike trip data are max depth = 12, max bins = 128, number of trees = 128, MSE = 84521, with a square root of MSE of 290s. The error distribution, calculated as the difference of actual trip duration and the predicted trip duration, are plotted below in Figure 11 and Figure 12. As we can see, the errors for taxi data are mainly between -300s and 300s, which is around a 5 minute period. The errors for bike data are basically the same, mainly between -300s and 300s range. Random Forest gives lower MSE for both taxi and bike trips for all three regression algorithms, which will give the best estimations and predictions.

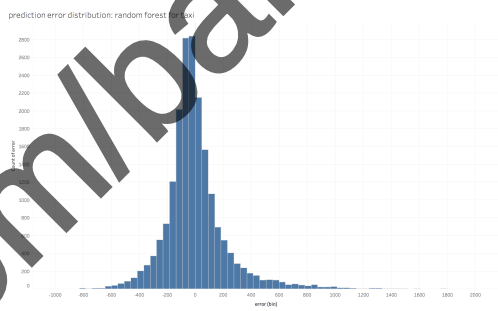

Figure 11 | Error Distribution of Taxi Data, Random Forest

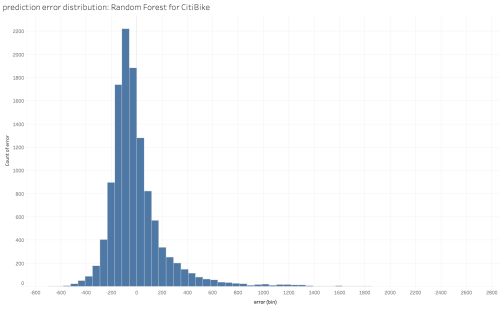

Figure 12 | Error Distribution of CitiBike Data, Random Forest

From the error distribution plots and the MSE for our models, it can be seen that our predictions are pretty precise with satisfactory accuracy. The model with the best accuracy, the best model of Random Forest, only gives ~5min error at most of the times. For the taxi and bike trips, ~5min error is a very satisfactory result, showing our best model can give the reasonable estimation of trip durations.

How did our k-means clustering and other feature construction help? We compared this using our Decision Tree model on taxi data. Using features without any new feature construction, including k-means, trip distance, weekday and peak hour, gave an MSE of 169373. If the trip distance was calculated and used as a feature in our prediction, the best model of Decision Tree gave an MSE of 113663. After k-means clustering was used, the best model of Decision Tree gave an MSE of 79335, as specified above. In this case, the

calculation of trip distance improve the performance by around 33%, and the use of k-means clustering improve the performance by around 30% compared to the best model with trip distance calculated, and by around 53% compared to the original model. This means that the feature extraction and construction are very important in our prediction. The implementation of calculating trip distance and k-means clustering require extra time when our algorithm was running, but the improvement of accuracy was truly impressive.

The performance improvement was also proved by using Linear Regression and Random Forest on our taxi data. Using features without any new feature construction gave an MSE of 300797 for Linear Regression and gave an MSE of 125332 for Random Forest. Using only trip distance calculated gave an MSE of 235783 for Linear Regression and gave an MSE of 99808 for Random Forest. The total improvements are 45% for Linear Regression and 44% for Random Forest, respectively. The improvements are truly remarkable and make our application more reliable.

In the regression model prediction, which features are the most important features? Unfortunately, the feature importance is not supported in Decision Tree and Random Forest regressors in Spark MLlib. However, we can use the weights of features in Linear Regression and the whole printed tree in Decision Tree to estimate the importance of features.

For Linear Regression, the weights of features for taxi data model shows that the most important feature is distance (~0.2), and other important features are starting latitude and longitude, ending latitude and longitude, and temperature (~0.003 - ~0.005). K-means clusters for starting point and end point are also important features (~0.0002 - ~0.0004). For bike data, the most important feature is also distance (~0.3), and interestingly, the second important feature is birth year of the user (~0.08), and then starting station ID, end station ID and temperature (~0.003 - ~0.008). The clusters for k-means are not very important for bike data, because the bike trips can only start and end at bike stations, and station ID will clearly play more important role than the clusters, because the station ID is a sort of cluster of all the bike coordinates.

For Decision Tree, the printed structure of tree shows that for taxi data, the most important feature is distance, the second important feature is the clustering of starting location, and the other important features include time of the day, starting latitude, ending latitude and peak hour. This is different from the Linear Regression, but the most important feature is the same. For bike data, the printed structure of tree shows that the most important feature is distance, and the other important features include time of the day, ending longitude, and birth year of user. Same as Linear Regression, the most important feature is the same; also same as Linear Regression, k-means clustering does not help a lot in the bike prediction, as specified above; again, an interesting result is the birth year of user plays a rather important role in the trip duration.

The analyzation of feature importance not only prove that the calculation of trip distances is necessary for improvement of our model precision, but also prove that k-means clustering is very useful in taxi data but not very useful in bike data. This is consistent with our common sense.

There are also interesting discoveries. Firstly, as specified above, birth year of rider greatly influence the bike trip duration. We believe the younger riders will ride faster. Secondly, weather does influence the trip duration, but it is not very important. The temperature is the most important factor in influencing the results, and the precipitation and snow fall do not help much. Thirdly, the time of the day is another important factor, indicating that different hours in the day will affect the trip duration. This is also consistent with our common sense because there are bad traffics in the certain period of the day, but the traffic should be much better in the midnight.

Finally, an application with a unified interface was build for the benefit of users. With this interface, the user can finish the training and testing process without changing to another class. The command line arguments for users to input are: weather data directory, trip data directory, model directory, k-means directory, trip type, algorithm type, run mode. The directories should all be in HDFS. The trip type should be either taxi or bike. The algorithm type should either be LR (Linear Regression), DT (Decision Tree) and RF (Random Forest). The run mode should either be training (training mode) and test (test mode). In the training mode, the data will be read from weather data and trip data directory, and the models will be saved at model directory and k-means directory. In the test mode, the data will be read from weather data and trip data directory, and the models will be read from model directory and k-means directory. This unified interface is very convenient to use and users do not need to focus on the details of models and parameters, because everything is finished automatically and only the best model will be saved by our program.

However, this application can only work on profiled NYC Yellow/Green Taxi data and CitiBike data. It does not output the predictions into a file, and does not have the prediction mode. The prediction mode can be easily implemented but the data joining and feature extraction process need to be modified a bit. With this function, the user can give the features that to be predicted and get the prediction results with specified algorithm within one run. To expand this approach to data other than NYC taxis and bikes, the design pattern should be changed greatly and there should be more parameters for users to specify. These features can be expanded given adequate time.

Moreover, the application can be extended to a rather useful one: something like a Google Maps trip planning. The user only need to input the starting address and destination address, and the application will calculate the coordinates automatically (through Google Maps API), grab the weather (through Yahoo Weather API), use the current date and time to make the prediction. This trip planner will be rather useful and can be used in every day life, and more importantly, it can be extended to any city and any commute method, given the enough training data.

VI. CONCLUSION

In this study, three machine learning regression algorithms in Spark MLlib were used to predict the trip duration of NYC Taxi and CitiBike trips. Weather data was joined to analyze the importance of temperature and climate. K-means clustering, trip distance calculation and other feature construction approached were used. Among three regression algorithms, Linear Regression, Decision Tree and Random Forest, the Random Forest model gives the best prediction with lowest

MSE of 69830, with only ~5min error at most of the predictions. K-means clustering and trip distance calculation in feature construction play very important roles in improving the performance of our algorithms. Trip distance is the most important feature in our models, and the k-means clustering is important in taxi data but not in bike data. Time of the day, as well as the temperature, is also important in predicting trip durations. Our algorithm and model gave very good estimations of trip duration. A fully-functioned application was built with unified interface, for the convenience of the user, and this application can either train the user-specified model or test the saved model. This application can also be extended to a trip planner application for everyday use.

ACKNOWLEDGMENT

Special thanks to Professor McIntosh who is always behind us when we have difficulties in our study. Also thanks to HPC stuff who provided the computing platform and essential help for the coding and debugging environment.

REFERENCES

1. H. Karau, A. Konwinski, P. Wendell and M. Zaharia, Learning Spark. O'Reilly Media Inc., Sebastopol, CA, February 2015.
2. M. Araki, R. Kanamori, L. Gong and T. Morikawa, "Impacts of Seasonal Factors on Travel Behavior: Basic Analysis of GPS Trajectory Data for 8 Months", Serviceology for Smart Service System, pp. 377-384, 2016.
3. J. Zhao, J. Wang and W. Deng, "Exploring bikesharing travel time and trip chain by gender and day of the week", Transportation Research Part C: Emerging Technologies, vol. 58, pp. 251-264, 2015.
4. F. Wu, H. Wang and Z. Li, "Interpreting traffic dynamics using ubiquitous urban data", Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '16, 2016.
5. J. Freire, A. Bessa, F. Chirigati, H. Vo, T. H., and K. Zhao, "Exploring What not to Clean in Urban Data: A Study Using New York City Taxi Trips", IEEE Data Engineering Bulletin, vol. 39, no. 2, pp. 63-77, 2016