Fare and Duration Prediction: A Study of New York City Taxi Rides

Christophoros Antoniades, Delara Fadavi, Antoine Foba Amon Jr.

December 16, 2016

1 Introduction

New York City taxi rides paint a vibrant picture of life in the city. The millions of rides taken each month can provide insight into traffic patterns, road blockage, or large-scale events that attract many New Yorkers. With ridesharing apps gaining popularity, it is increasingly important for taxi companies to provide visibility to their estimated fare and ride duration, since the competing apps provide these metrics upfront. Predicting fare and duration of a ride can help passengers decide when is the optimal time to start their commute, or help drivers decide which of two potential rides will be more profitable, for example. Furthermore, this visibility into fare will attract customers during times when ridesharing services are implementing surge pricing.

In order to predict duration and fare, only data which would be available at the beginning of a ride was used. This includes pickup and dropoff coordinates, trip distance, start time, number of passengers, and a rate code detailing whether the standard rate or the airport rate was applied. Linear regression with model selection, lasso, and random forest models were used to predict duration and fare amount.

2 Related Work

The fare of a taxi ride is function of the mileage and the duration of the ride (sum of drop charge, distance charge and time charge). The drop charge is constant and the distance can easily be estimated but evaluating the duration is not a trivial task. It is the result of complex traffic processes that are nonlinear.

One way to predict duration is by doing short term prediction with the help of real time data collection. In [1] the authors tackle the problem by using data from from buses (GPS) and an algorithm based on Kalman filters. Using a similar approach, [2] uses real time data from smartphone placed inside vehicles.
Estimating travel time for highways yields better results than in the cities. This allows for more accurate predictions. In [3] the authors use a combination of traffic modelling, real time data analysis and traffic history to predict travel time in congested freeways. They try to overcome the assumption that real time analysis communication is instantaneous. A lot of other papers also work on freeways. In [4] the prediction is done using Support Vector Regression (SVR) while in [5] Neural Networks (SSNN) are used.

Predictive estimates of future transit times is a feature that was released in 2015 in the Google Maps API [6]. This shows the importance of being able to predict time travel without having real time data of traffic.

We are trying to solve a similar problem: estimating ride duration without real time data, by analysing data collected from taxis. Being able to do such estimation would help making better future predictions.

3 Data

The data used in this study are all subsets of New York City Taxi and Limousine Commission's trip data, which contains observations on around 1 billion taxi rides in New York City between 2009 and 2016. The total data is split between yellow taxis, which operate mostly in Manhattan, and green taxis, which operate mostly in the outer areas of the city. For the main analyses of this study, the data for yellow taxi rides during the month of May 2016 were used, although the models were validated on additional data. Since each month consists of about 12 million observations, and there were computational limitations, subsets of the monthly data were used for model building, and other subsets were used for validation. To build the models, a random subset of 10,000 observations from May 2016 were used, of which 8,000 were used for training and 2,000 for validation.

The original dataset contains features as pickup and dropoff locations, as longitude and latitude coordinates, time and date of pickup and dropoff, ride fare, tip amount, payment type, trip distance and passenger count (as well as other, for this study, less relevant variables). The data was processed to extract separate features for year, month, day, weekday, hour and minute from the date and time of each ride, as well as trip duration as the difference between dropoff and pickup time. Furthermore, with the objective to model and account for traffic in the predictions, two more features were calculated from the data; rides in an hour and average speed during the hour. Rides in hour represents the number of started rides within the hour of each observation, and the average speed represents the average speed of all those rides.

Figure 1 show the distributions of ride duration and fare amount, which are clearly similar (the spike at $52 represents rides to JFK International airport). The objective of this study has been to predict both, although as the results and models chosen are very similar, the illustrations and results have been focused on the prediction of trip duration.

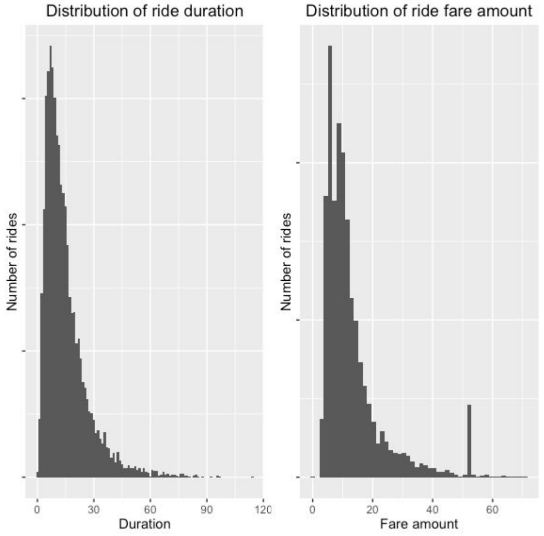

Figure 1: Duration and Fare Amount Distribution

4 Models and Methodology

4.1 Linear Regression

As a baseline prediction, the mean duration and fare from the training set were used to predict a constant value for the validation set.

Model Selection

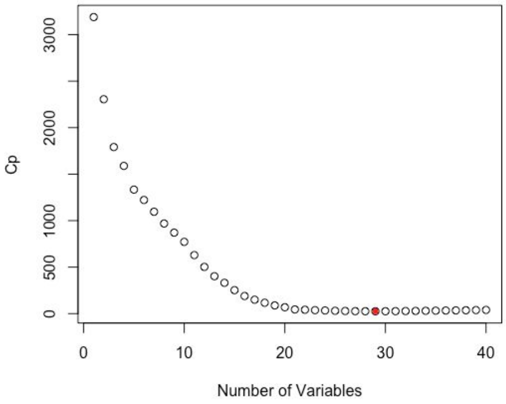

Figure 2: Forward Selection for Linear Regression

To avoid selecting a sub-optimal model by selecting covariates by hand, forward selection was used to identify which subset of covariates would be best to use. When iteratively adding the variables that minimizes the RSS one at a time it is evident that selecting a model with all covariates does not improve the C_p score (proxy for test error) over a 20-covariate model (figure 2). Therefore, for simplicity reasons, the smallest model was selected for linear regression.

Lasso

To further confirm the best set of covariates to use, the lasso method was used to shrink coefficients. Lasso results in a sparser model, which makes interpreting the model easier.

$$\sum_{i=1}^{m}(y^{(i)}-\theta_0-\sum_{j=1}^{n}\theta_j x_j^{(i)})^2+\lambda\sum_{j=1}^{n}\left|\theta_j\right| = RSS+\lambda\sum_{j=1}^{n}\left|\theta_j\right| \quad (1)$$

For both the fare and duration prediction models, lasso was run using a range of values for the penalizing parameter, λ. Cross validation was used to find the lasso model with the lowest error and select the value of λ to use. In both cases, the λ values that gave the lowest cross validation error were close to zero. For example, the optimal λ parameter for the fare regression was $1.19 \cdot 10^{-5}$. Because of these results, variables were not penalized using lasso, and the linear regression model selected with forward selection was used.

Linear Regression Model

From feature selection, the linear regression with all covariates available at the pickup time was predicted to be the best for both duration and fare prediction. The linear regression model finds the set of θ coefficients that minimize the sum of squared errors

$$y^{(i)} = \theta_0 + \sum_j \theta_j x_j^{(i)} \tag{2}$$

Interaction and Higher Order Terms

Because the available covariates alone cannot model the nonlinear effects of traffic, interaction and higher order terms are considered to allow the model to fit these effects more closely. Plotting the covariates used in the linear models against each other shows that there are no strong correlations between them, which suggests that interaction terms should not be included in the model.

However, second order terms do make logical sense to be used in the model. Since the nonlinearities arise out of traffic patterns, it can be assumed that longer distance trips experience more instances of traffic. Therefore, adding squared trip distance to the model would increase the trip distance's importance in the duration and price prediction.

4.2 Random Forest

As traffic is clustered and aggregated more densely to different locations at different times, the location of the ride will clearly have an affect on the trip duration. Although there is no straightforward way of considering all locations between the start and end points of a ride, the pickup and dropoff locations are available in the dataset and can be used to model some of the effect of traffic and conjunctions. In the linear regressions, the locations' effect on trip duration is modeled simply by the magnitude of the longitude and latitude coordinates. As traffic is clearly not varying solely based on the magnitude of the coordinates, the linear models fail to account for the nonlinear effect the locations have on traffic and hence trip duration (and also fare amount). An algorithm that can better account for these nonlinearities is the random forest.

The random forest algorithm 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 [7][8]. Each of these decision trees aims to divide the predictor space, i.e. the set of all possible values for the features $x_1, x_2, ..., x_n$, in J distinct and non-overlapping regions $R_1, R_2, ..., R_J$. The predictor space is divided into high-dimensional rectangles, with the goal to find rectangles $R_1, R_2, ..., R_J$ that minimize the RSS,

$$\sum_{j=1}^{J} \sum_{i \in R_j} \left(y^{(i)} - \hat{y}_{R_j}\right)^2 \tag{3}$$

where $\hat{y}_{R_j}$ is the mean response for the training observations within the *j*th rectangle. When building each tree, a top-down approach is taken. Beginning with all points in the same region, the algorithm successively splits the predictor space into two halves, stopping when there are no more than five points in a region. At each split, a prediction x_j and cutpoint s are chosen such that splitting the predictor space into the regions $\{x \mid x_j < s\}$ and $\{x \mid x_j \geq s\}$ leads to the biggest reduction in RSS. Defining the pair of halves as $R_1(j, s)$ and $R_2(j, s)$, at each split we seek to find j and s that minimize the equation

$$\sum_{i:\, x^{(i)} \in R_1(j,\, s)} \left(y^{(i)} - \hat{y}_{R_j}\right)^2 + \sum_{i:\, x^{(i)} \in R_2(j,\, s)} \left(y^{(i)} - \hat{y}_{R_j}\right)^2 \tag{4}$$

Once the regions are defined, a prediction by a single tree is made by averaging the responses of the training observations in the region to which the test observation belongs. In the random forest, a large number of trees are fit, each using a bootstrap sample from the training data, and a prediction of a new observation is made using the mean of the predictions by all the trees. At each split, only m of the total n predictors are randomly chosen to be considered. This approach is taken to decorrelate the trees, as considering all predictors might yield very similar trees when one or a few predictors are particularly strong. As averaging many uncorrelated trees leads to a larger reduction in variance, this approach often yields better prediction results. As can be seen in figure 3, the model performs better for a smaller choice of m. Also, averaging over a larger number of trees yields a better results, although the effect is flattening out after a few hundred trees. To optimize prediction accuracy, $m = \sqrt{n}$ and 500 trees were used.

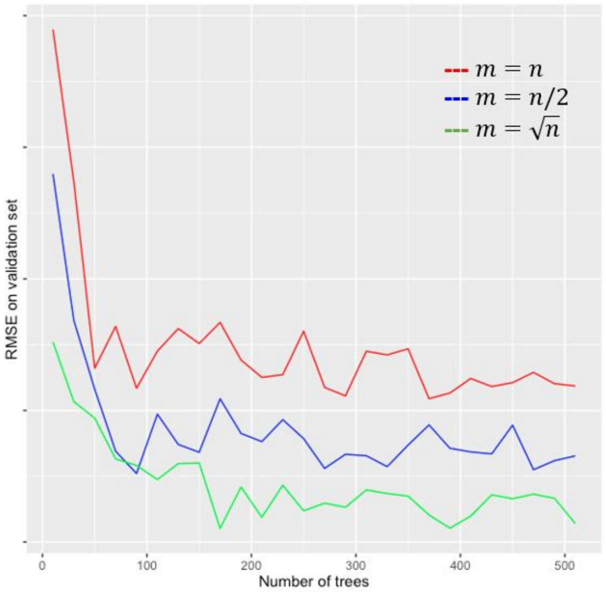

Figure 3: Effect of m on Random Forest Results

As discussed above, a training sample of 8,000 observations were used to train the models. Although the random forest performs better when trained on a larger sample size, the incremental improvement is decreasing with the number of observations. As seen in figure 4 the effect of using a larger training set is very small for a training set with more than 4,000 observations.

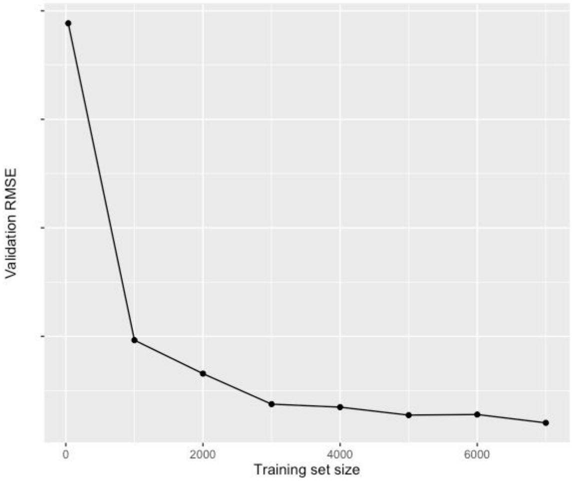

Figure 4: Random forest performance with larger training size

Figure 5 below shows the a variable importance plot for the random forest model on trip duration. The variables for rides in hour and average speed in hour explain the most variance, supporting them as ways of modeling at least parts of the effect of traffic.

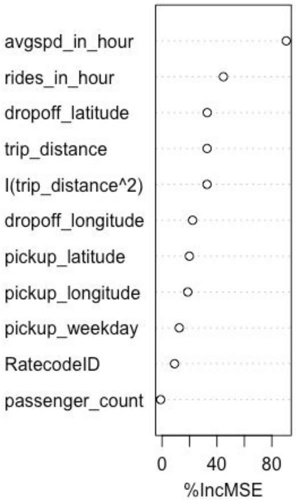

Figure 5: Random forest variable importance

Coordinate Transformation

One additional approach taken to further model the effect of the pickup and dropoff locations was to transform the coordinates [9]. Most of the streets and avenues in Manhattan are aligned in a grid structure. With the hypothesis that the avenue or street could explain some of the effect of the location, transforming the coordinates so that the splits in the random forest algorithm will be made aligned and perpendicularly to the avenues and streets, could potentially yield better predictions.

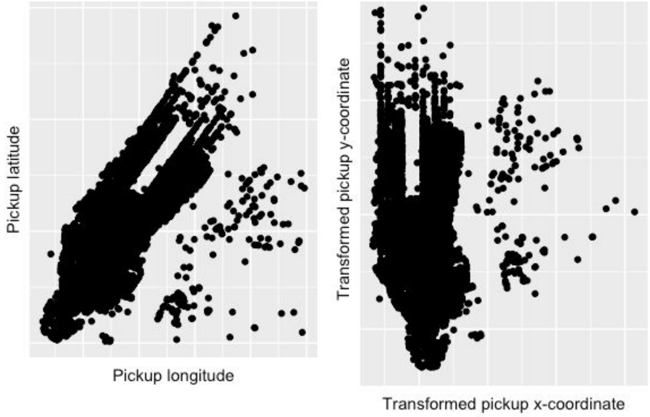

Figure 6: Transformation of location coordinates

Finding two points on the same avenue on opposite sides of Manhattan, the angle ϕ to rotate the coordinates was calculated to 36.1° using the differences in longitudes and latitudes. The rotated coordinates were then calculated using equation (5) and (6),

$$x_{rot} = x\,cos(\phi) - y\,sin(\phi) \tag{5}$$
$$y_{rot} = x\,sin(\phi) + y\,cos(\phi) \tag{6}$$

where x is the location longitude and y the latitude.

5 Prediction Results and Discussion

Prediction	% Error Validation	RMSE Validation	RMSE Train
Fare amount Baseline mean	54.2%	$10.45	$10.36
Fare amount Linear regression	21.1%	$3.52	$3.04
Fare amount Random forest	14.0%	$2.28	$2.16
Duration Baseline mean	97.8%	12.05 min	11.43 min
Duration Linear regression	38.5%	6.51 min	6.17 min
Duration Random forest	24.3%	5.24 min	5.09 min

Table 1: Transformation of location coordinates

5.1 Linear Regression Prediction Results

The linear regression improves its predictions as covariates are added to the model. However, there is a limit to its performance. At a certain point, adding more higher order terms or training on more data does not improve predictions. As discussed previously, this is a result of the nonlinear patterns in traffic, which affects both duration and fare.

5.2 Random Forest Prediction Results

The random forest model outperform all other models used, as it manages to model the nonlinearities of traffic and location effect. Although the model accounts for the effect of pickup and dropoff locations, it has no way of modeling the effects of the locations along the route. A ride between two locations with high traffic can still be relatively fast if it goes through high-speed areas with little or no traffic. Considering what is accounted for in the models, they are believed to predict both duration and fare relatively precisely.

Although using rides in hour and the average speed in hour improves the models and hence works as proxies for traffic modeling, rotating the location coordinates does not yield any significant improvement in prediction accuracy.

5.3 Comparison to External Predictions

As a benchmark, duration predictions by Google Maps, and a fare prediction by a tool on the Taxi Fare Finder website [10] were explored. As an example, for a ride between the Empire State Building and Brooklyn Bridge the tool predicts a price in the range between $17 and $59 depending on traffic. For the same ride, on a Tuesday around noon, Google Maps predicts it to take between 12 and 22 minutes. Our model predicts the ride to take 18 minutes and cost around $25. The models clearly make similar predictions to the tool and Google Maps, that both make predictions in a very wide range. With an average error around $2 and 5 minutes the models seem to make precise predictions comparing to the tool and Google Maps.

6 Conclusions and Future Work

Considering what is and what is not accounted for in the models built in this study, their predicting results are fairly accurate. To further improve the prediction accuracy, more variabilities need to be considered and modeled. Although the rides in hour and average speed in hour work as proxies for traffic, more modeling on the effect of location is needed. These quantities could be calculated for different areas to further model local effects of traffic. Also, modeling traffic and the effect of location in between pickup and dropoff points should be considered as well as difference in drivers' speed.

These further steps could be taken both by analyzing larger sets of the data to infer relationships and effects of location and traffic at different times, as well as aggregation with other datasets, as data on traffic, speed limitations, etc.

References

[1] Vanajakshi, L., S. C. Subramanian, and R. Sivanandan. "Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses." *IET intelligent transport systems* 3.1 (2009): 1-9.

[2] Biagioni, James, et al. "Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones." *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2011.

[3] Yildirimoglu, Mehmet, and Nikolas Geroliminis. "Experienced travel time prediction for congested freeways." *Transportation Research Part B: Methodological* 53 (2013): 45-63.

[4] Wu, Chun-Hsin, Jan-Ming Ho, and Der-Tsai Lee. "Travel-time prediction with support vector regression." *IEEE transactions on intelligent transportation systems* 5.4 (2004): 276-281.

[5] Van Lint, J. W. C., S. P. Hoogendoorn, and Henk J. van Zuylen. "Accurate freeway travel time prediction with state-space neural networks under missing data." *Transportation Research Part C: Emerging Technologies* 13.5 (2005): 347-369.

[6] Kelareva, Elena. "Predicting the Future with Google Maps APIs." Web blog post. *Geo Developers Blog,* https://maps-apis.googleblog.com/2015/11/predicting-future-with-google-maps-apis.html Accessed 15 Dec. 2016.

[7] James, G, D Witten, T Hastie, and R Tibshirani. *An introduction to statistical learning*. Vol. 6. , New York, Springer., 2013.

[8] Friedman, J, T Hastie, and R Tibshirani. *The elements of statistical learning*. Vol. 1. , Berlin, Springer, 2001.

[9] Blaser, Rico, and Piotr Fryzlewicz. "Random rotation ensembles." *J Mach Learning Res* 2 (2015): 1-15.

[10] *Taxi Fare Finder*, https://www.taxifarefinder.com/main.php?city=ny&lang=en Accessed 16 Dec. 2016.