Technical Exercise

Overview

How much should a competitor to Uber and Lyft recommend a passenger tips their driver?

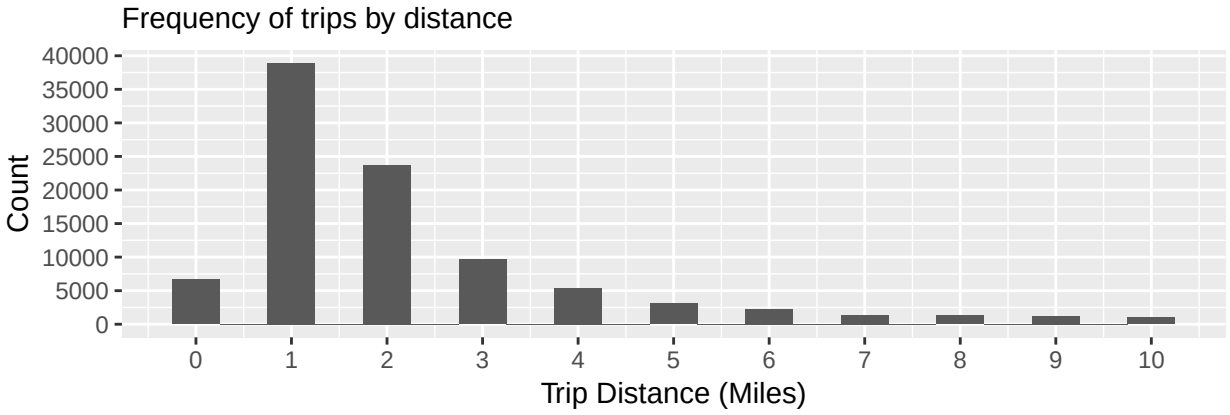
In order to provide an answer to this question, I took a deep dive into historical (March, June, and November 2017) yellow taxi data supplied by the city of New York. The purpose of this is to get insights on the transportation landscape of New York and to start building models to predict tips with the insights gained.

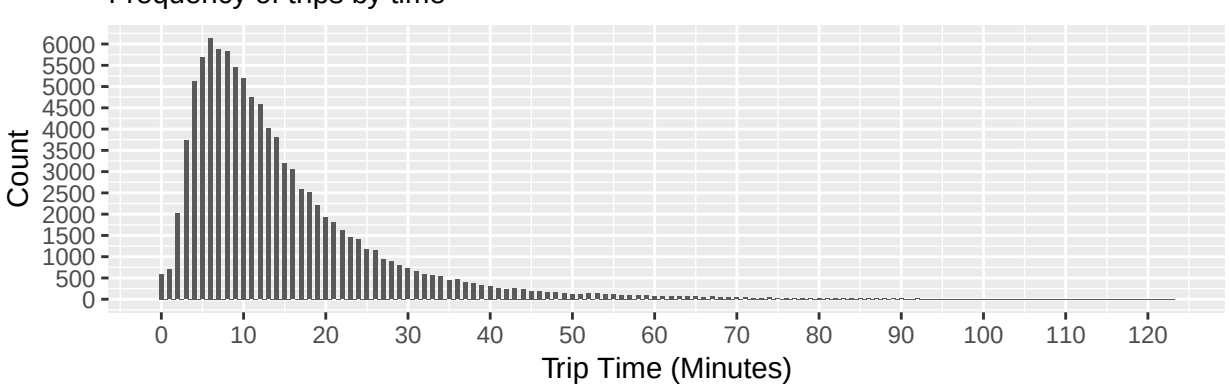
Data

The data had just under 30 million observations of 18 variables, both categorical and numerical. Credit cards (67.9%) and cash (31.4%) accounted for most of the transactions, the remaining 0.7% are either no charge, dispute, or unknown. Basic statistics for the numerical data can be seen below. Trip time was calculated by subtracting the pick-up time from the drop-off time, all other variables were already in the dataset.

```
##        passenger_count trip_distance trip_time fare_amount tolls_amount
## Mean          1.618000      2.919000   16.9200     13.1090     0.329000
## Median        1.000000      1.600000   11.3000      9.5000     0.000000
## SD            1.260992      4.476535  151.2975    147.2817     1.968881
##           mta_tax improvement_surcharge tip_amount total_amount
## Mean   0.49700000            0.30000000    1.87400      16.4470
## Median 0.50000000            0.30000000    1.36000      11.8000
## SD     0.07081708            0.01408904    2.64557     147.5248
```

The plots below show a random subset of 100,000 observations in order to visualize the breakdowns of trips by distance and time. For both plots, outliers (2+ standard deviations away from the mean) have been removed in order to make everything easier to visualize.



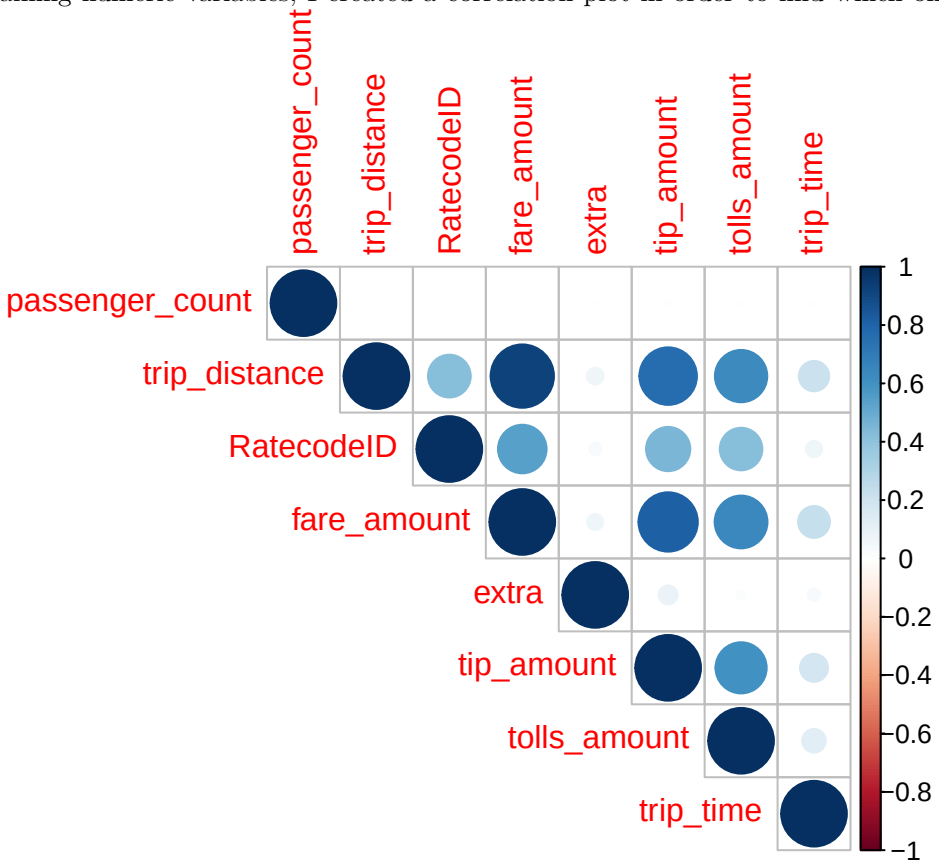


The number below shows the average tip percent of all those who tipped. Surprisingly, New Yorkers are better tippers than I expected.

```
## [1] 0.2001922
```

Model

The first step to deciding which model to fit to the data was to find out which explanatory variables properly explained some of the variance with my response, the tip amount. I immediately discarded VendorID, pickup and drop off time, store and fwd flag, and total amount. VendorID and store and fwd flag are both factors which have absolutely no impact on tip amount; pickup and drop off time are accounted for with the travel time variable; and total amount is all of the monetary variables, including tip amount, summed. With the remaining numeric variables, I created a correlation plot in order to find which ones mattered.



In order to help decide which of the categorical variables are necessary, and also reinforce which of the numeric variables need to be used, I used boruta variable selection. Below are the results from running the boruta variable selection.

Confirmed

```
## [1] "trip_distance" "RatecodeID"    "PULocationID"  "fare_amount"
## [5] "extra"         "tolls_amount"  "trip_time"     "pre_tip"
## [9] "tip_."
```

Rejected

```
## [1] "passenger_count" "DOLocationID"
```

I trained a bayesian GLM with 10-fold cross validation on a random sample of 6,000 observations using the above confirmed explanatory variables and then tested it on 2,000 observations. A bayesian GLM was the preferred model for this for two reasons: the computation time is extremely fast (even when using a laptop) and the accuracy was still pretty decent. Those two factors made it the optimal choice over other machine

learning techniques.

```
## Bayesian Generalized Linear Model
##
## 6402 samples
##    9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 5761, 5762, 5761, 5762, 5762, 5762, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1.222998  0.8379804  0.3587578
```

With an RMSE around ~1.30 and an R-Squared around ~0.75, the bayesian GLM accounts for the majority of the variance. While I would prefer to minimize the RMSE even further, this is a good baseline for a "first draft" model.

The below number is the average difference (in cents) the prediction is from the actual tip amount for all test data.

```
## [1] 0.3363204
```

A few issues that may come up with this model, is what happens during busy times? For instance, if the client implements something similar to surge pricing during rush hour, the algorithm would still recommend the tip amount of a regular price instead of a surge price. In this case, the best thing to do would be to add a similar multiplier to the tip amount during these specific timeframes.

Next Steps
The next steps in order to increase accuracy would be to turn this into a stacked regression. The possible models to be used for this would be the bayesian GLM, prinicpal component regression, a parallelized random forest, cubist regression, and a gradient boosting machine. The predictions from each of these models would be averaged out to come up with one number derived from all 5 models. While this would be significantly more time consuming, putting the algorithm on a server or spark instance would make this a feasible option.

API
The simplest way to share this in a functioning manner with the client would be to host the actual algorithm on a shiny server (either somewhere in AWS or Azure) to speed up the processing and build a shiny app on top of it; this would be a clean, simple, fast version which could be shared.

Another option would be to, once again host it on a server, but then pass it along to a software/front end engineer to build a more traditional GUI.