ST4240 DATA MINING - PREDICT THE PRICE OF TAXI JOURNEYS IN CLASS CASE COMPETITION ON KAGGLE

BY GROUP 5

HU, XI	**A0133899L**
LI, RUIHAN	**A0130060E**
MA, XUTONG	**A0133912M**
MENG, XIANG	**A0133998L**
WANG JIXI	**A0133966W**

INTRODUCTION

In this data analysis case, we focus on some key issues currently in the transportation industry, i.e., to predict the duration as well as the distance of the trip. With the emerging phenomenon of new sharing economy, represented by Uber and Grab in Southeast Asia, our case could be scalable high enough to draw managerial group's attention. In addition, our case could also become important to both drivers and customers. Better understanding of the fundamental mechanisms could contribute to drivers' decision making and individual's ride strategy.

This case is brought up as a in-class data analytics hackathon in *National University of Singapore*, under a hardcore module *ST4240 Data Mining*, under the instruction of *Professor Alexandre Thiéry*. In our analysis, we firstly attempt to extract and understand the provided dataset, as well as all features included, to create a prediction model for this problem. In addition, we continue to conduct some exploratory analysis and extract more possible features (Feature Engineering) to better fit the model. Then we fit the prepared dataset into various models and evaluate their performance, discuss about the effectiveness, accuracy and shortcomings.

In this report, the following models are considered: Linear Regression Model, LASSO Regression, XGBoost Regression, Random Forest Regression. According to the instruction, we evaluate our model accuracies based on the Root Mean Square Percentage Error (RMSPE). Significance of different features are also considered and discussed in according prediction algorithms.

DATA

The dataset is provided by *Professor Alexandre Thiéry* and downloaded from *Kaggle*. Total dataset, including training data and testing data, provides nearly 1 million trips (930,344 trips, to be exact) from a disguised city. There are in total 800 drivers, indexed by a Taxi ID. Each trip records: Taxi driver's ID, pick-up coordinates and drop-off coordinates (coordinates are in the form of integers, not latitude and longitude), a timestamp when a trip starts, duration of trip (training data only, in integer), trajectory length of trip (training data only, in integer), and a list of coordinates recorded every 6 seconds to represent the exact trajectory of a trip (training data only, in integer). In this case, we use PRICE = DURATION + TRAJECTORY LENGTH for simplicity.

From a preliminary analysis on the dataset, we found that the dataset is clean and well structured. No missing data included and it is pretty evident that no discrepancy between training data and test. Some details from the preliminary analysis:

- Only 800 drivers included in this whole dataset, indexed from 1 to 800
- Pick-up and drop-off coordinates are sensible, ranging from -400 to 400
- No outliers found
- Available features do not show discrepancies between training data and testing data

EXPLORATORY ANALYSIS

To better understand the dataset and the problem statement at hand, to add on the preliminary analysis, we conduct an exploratory analysis. The purpose of this analysis is to obtain insights on complicated relationships among all features, and to exploit all possible new features which might contribute to the prediction model.

DURATION and TRAJECTORY LENGTH. We firstly explore the dependent variables that we are interested in. Figure 1.1a and Figure 1.2a plot the histogram on duration and trajectory length accordingly, and we could observe the right skewness. To make them more normal, we applied logarithm transformation, shown by Figure 1.1b and Figure 1.2b. We could see that the logarithm of trajectory distance appears normal, but that of duration still appear right skewed.

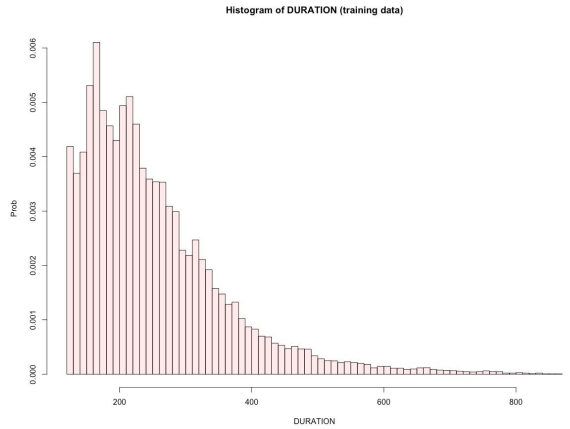

Figure 1.1a

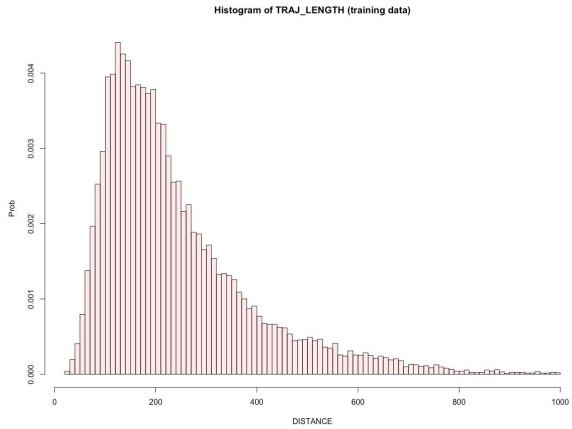

Figure 1.2a

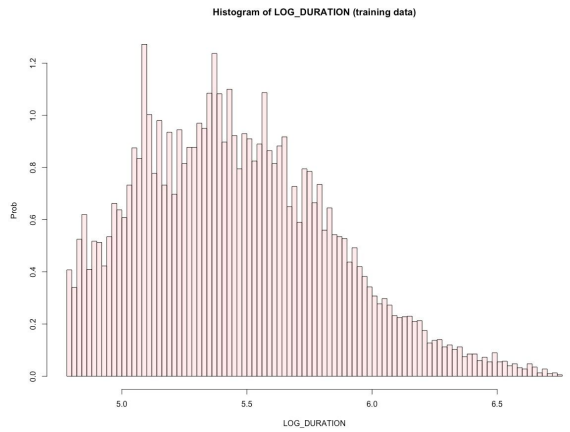

Figure 1.1b

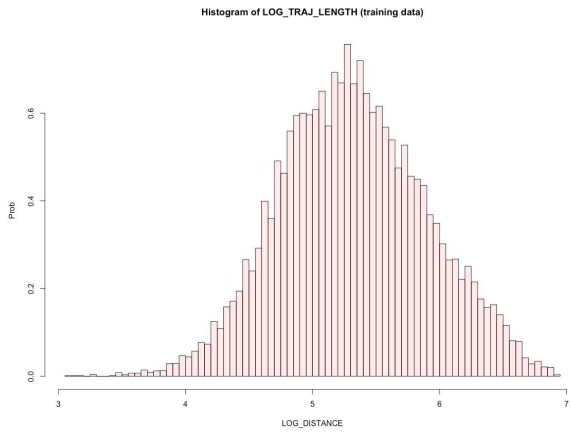

Figure 1.2b

DURATION ~ TRAJECTORY LENGTH. We observe a positive correlation (ρ = 0.53) between duration and trajectory length, demonstrated by Figure 2.a. Also, by further observing the plot, we could find that the variance of duration increases as the trajectory length increases, because the scatter plot displays as a fan-shaped area. After we apply the logarithm transformation, the linear relationship between duration and trajectory length becomes clearer, represented by Figure 2.b, with a positive correlation coefficient of 0.58.

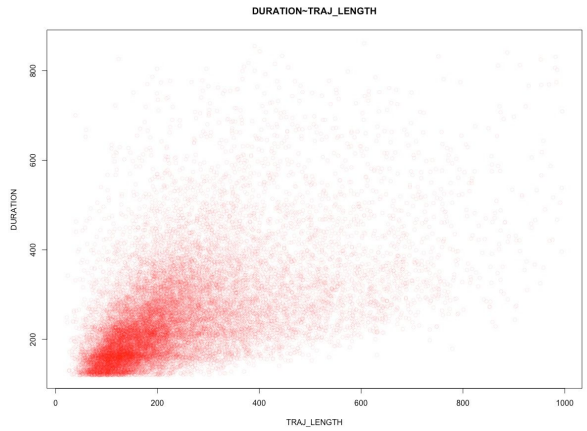

Figure 2.a: positive correlation between DURATION and TRAJECTORY LENGTH (ρ = 0.53)

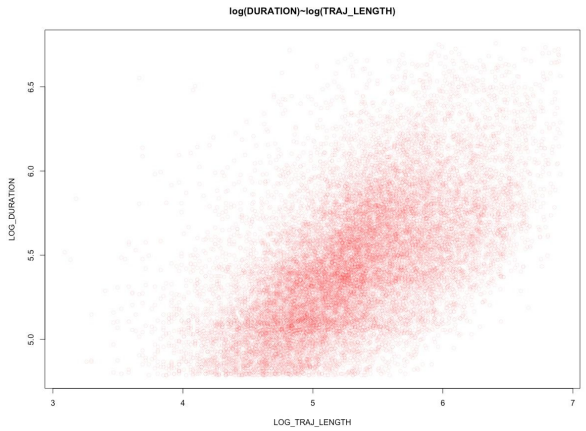

Figure 2.b. Linear relationship between log(duration) and log(trajectory length), (ρ = 0.58).

PICK UP & DROP OFF POINTS. 10,000 pairs of x and y coordinates are randomly sampled without replacements and plotted. They are separately plotted in Figure 3.a and Figure 3.b, showing the pick up points and drop off points of these trips. We notice that there is a geographical center in the southwest corner[1] in Figure 3.a, and there is a even more dominant geographical center in Figure 3.b. A sensible guess would be that the southwest area would be the CBD area or office area, where people would have to travel to work. This could be proved in a way that density of points in the east part is much more sparse in Figure 3.b than in 3.a. Further, by extracting available trajectory records in training data, we plot the city's map in Figure 3c. High density of traffic roads in the southwest area in a way proves our reasoning.

[1] Assuming top is north, and this applies to the entire case when referring to direction, unless otherwise stated.

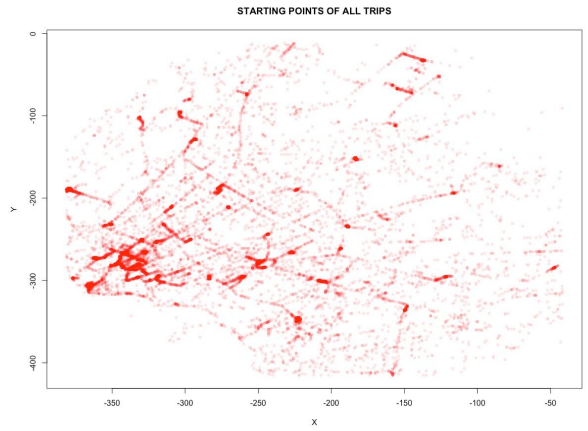

Figure 3.a. Starting points in the city

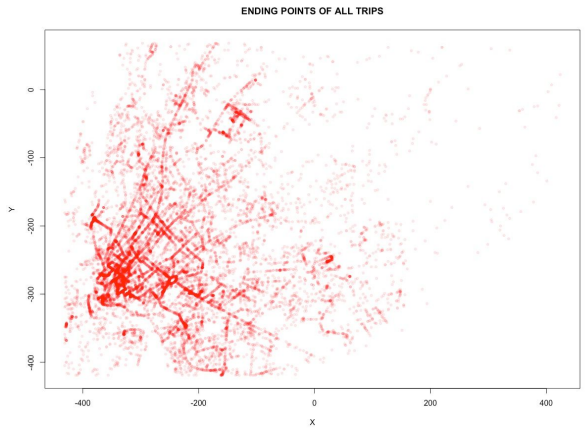

Figure 3.b. Ending points in the city

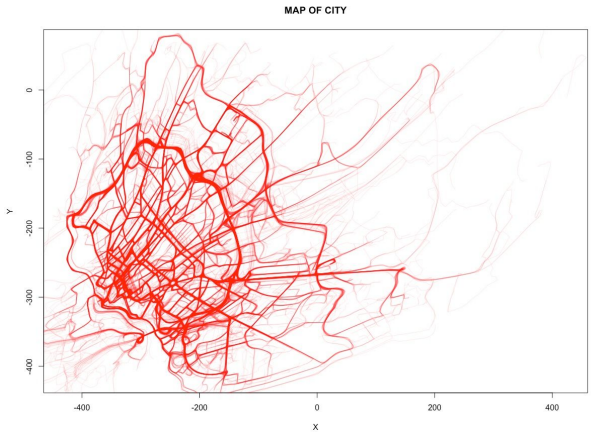

Figure 3.c. City map

FEATURE ENGINEERING

Given the condition that some features are only included in training dataset, for example, trajectory records. Thus, only features which are available in testing data before a trip starts, including those features engineered, are used to predict the trajectory length. Later on, predicted trajectory length is used to predict the duration.

PREDICTIVE TASKS. Initially we incorporate the the logarithm of trajectory length, day, date, hour, pick up point, and drop off point, also some more driver's qualities engineered. Since predicting the duration directly is more complicated because of its huge variance (demonstrated in Figure 2.b). Therefore, we choose to predict the ratio of trajectory length and duration, i.e., mean velocity of the trip (mean velocity = trajectory distance/duration). Figure 4.a shows that the variance of velocity increases as the trajectory length increases. This could suggest that as the trajectory length increases, the more likely that the trip would go to some less crowded area so

that the driver normally could driver faster. To fix this, we continue to apply the logarithm transformation to the velocity. Figure 4.b shows that we achieved a more linear relationship.

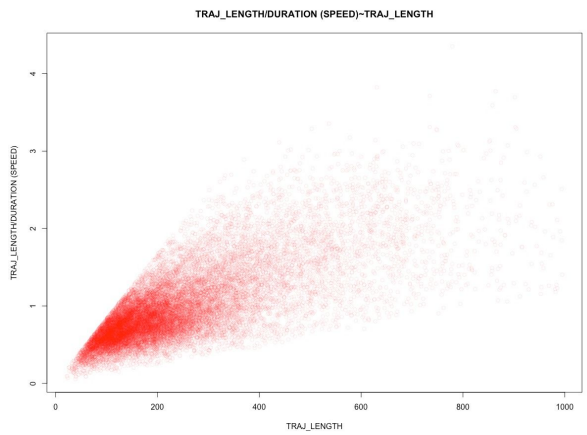

Figure 4.a. Relationship between mean velocity and trajectory length

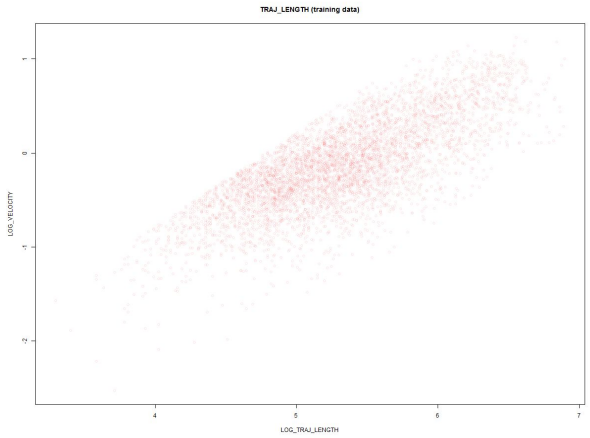

Figure 4.b. Relationship between logarithm of mean velocity and logarithm of trajectory length

STRAIGHT DISTANCE BETWEEN PICK UP AND DROP OFF POINTS. Based on given coordinates, we calculate the straight distance (logarithmically transformed), as well as the second order. Figure 5.1.a to Figure 5.2.b demonstrated the linear relationships to dependent variables accordingly.

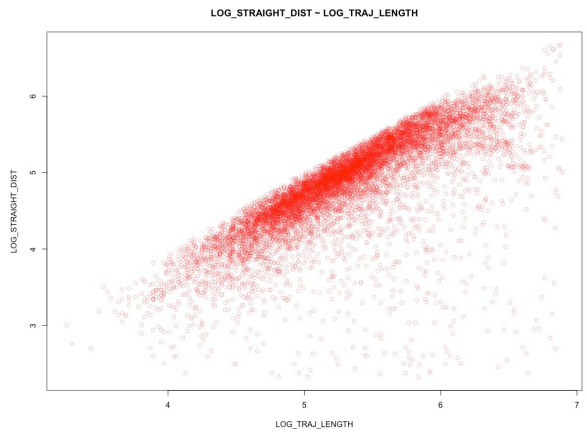

Figure 5.1.a logarithm of straight distance to logarithm of trajectory length

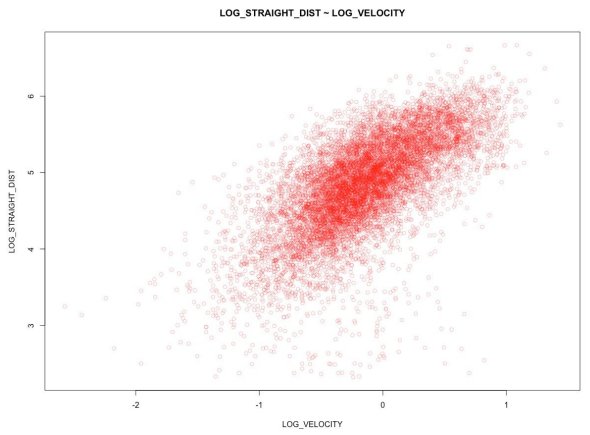

Figure 5.1.b logarithm of straight distance to logarithm of velocity

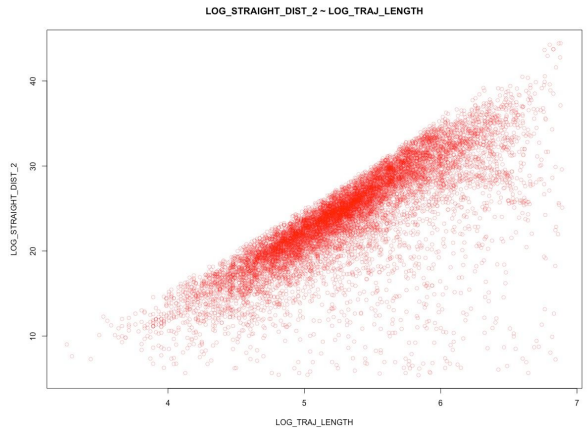

Figure 5.2.a second order of logarithm of straight distance to logarithm of trajectory length

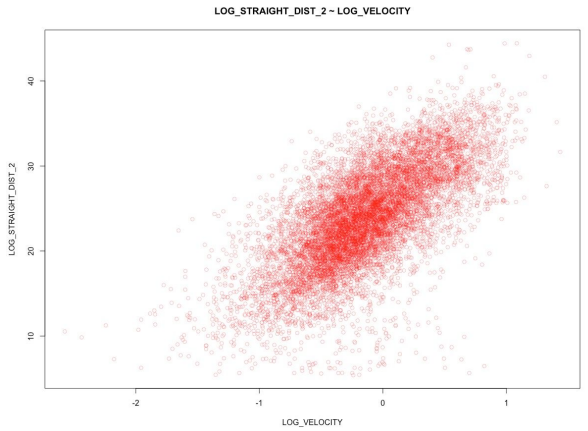

Figure 5.2.b second order of logarithm of straight distance to logarithm of velocity

MONTH, WEEKDAY, HOUR. With the purpose to examine the possible effects of timing, we extract these information out of the timestamp when trip starts. Interestingly, we notice that drivers tend to drive slower in summer and early autumn (July, August, September and October). On weekends, average speed is significantly faster than that on weekdays. Further, average velocity is about 1.5 times faster in midnight than in afternoon times, with the fastest of over 1.3 at 6 AM and the slowest of about 0.8 at 6 PM.

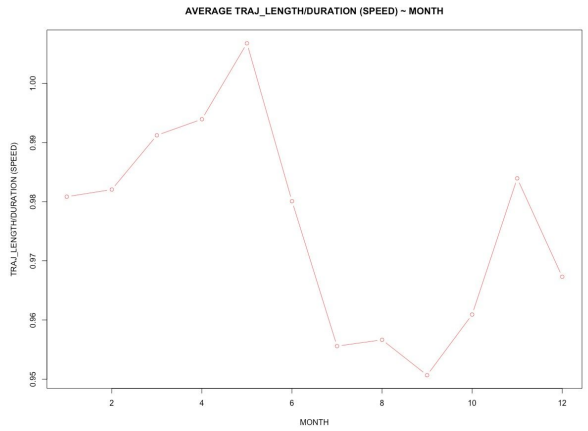

Figure 6.1 velocity in different month

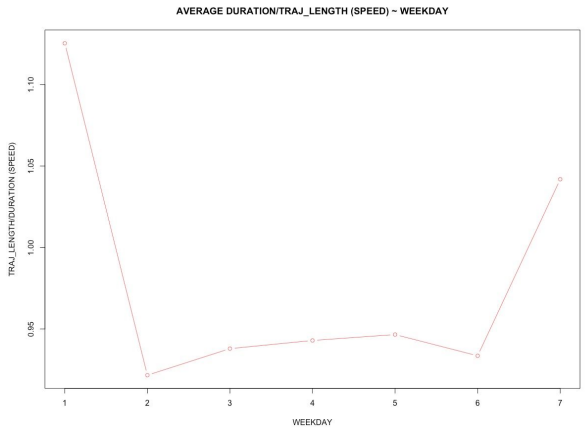

Figure 6.2 velocity in different weekday

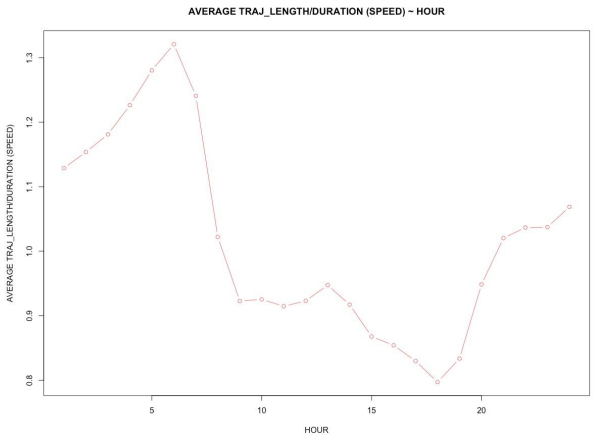

Figure 6.3 velocity in different hour

DRIVER'S IDENTITY. With only 800 drivers included in the dataset, we hypothesize that idiosyncrasy might differentiate them. For example, some drivers are smarter with excellent tactics, so that they are more likely to drive further for each ride. Some drivers are more experienced so that they could normally driver faster than the others[2]. We filter by each driver and calculate their average speed, Figure 7.1 demonstrates that their speed roughly follows a normal distribution with a few hardworking drivers and a few lazy drivers. By considering k-means algorithm and cross-validated using *Within Sum of Squares,* we decide a clustering of 4 is the best (Figure 7.2). Figure 7.3 demonstrates that total 800 drivers are classified into 4 clusterings according to their average speed.

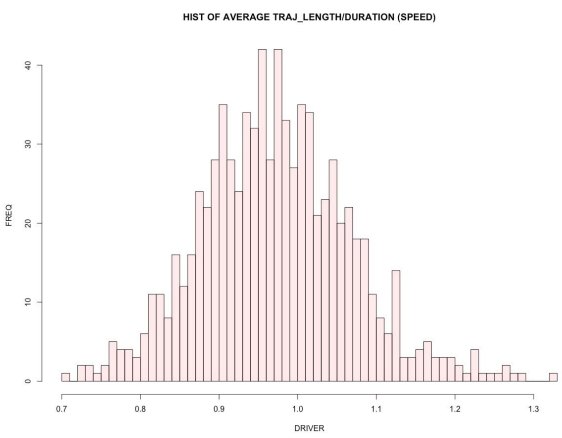

Figure 7.1 Histogram of driver's average velocity

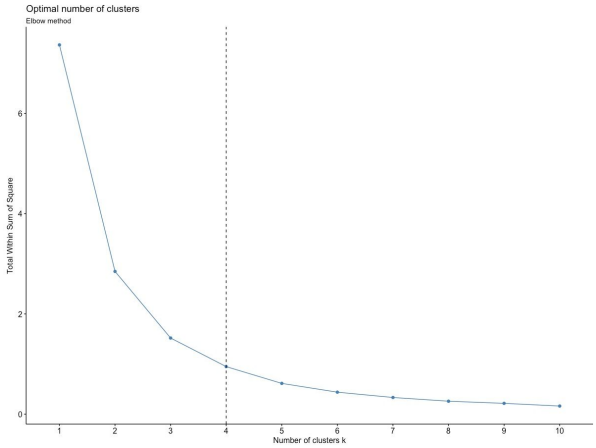

Figure 7.2 Optimal cluster of (no. = 4) driver's diligence rank

[2] Defined as Driver Trajectory Rank and Driver Diligence accordingly.

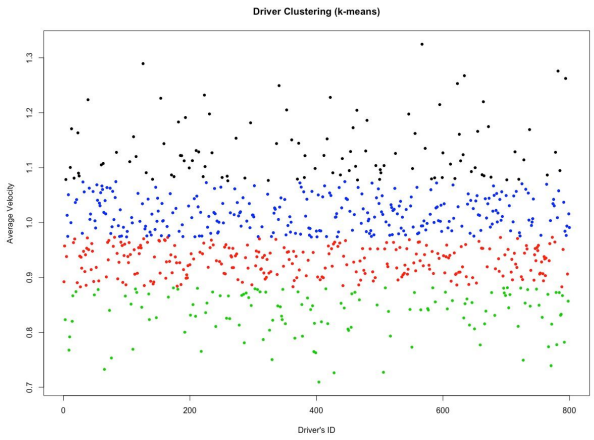

Figure 7.3 Drivers are clustered into 4 groups based on the average speed, cross-validated using 'within sum of squares', k-means clustering

PICK UP & DROP OFF POINTS.

It's important to represent pick up and drop off points appropriately. We hypothesize that coordinate data is an rather important aspect of features, because given the road path is fixed in the city, representing pick up and drop off points appropriately would possibly increase the prediction accuracy. However, we fail to cross validate the optimal number of clustering given the computation resource available to us. In addition, we would assume the number of clustering would exceed 100 if we apply the same cross validation approach described above. This would generate even more dummy variables when we construct the prediction model. Our computation capability does not allow us to do so, so that we take a step back and choose the number of 40. Figure 8.1 and Figure 8.2 represent the predicted clustering results using k-means algorithm.

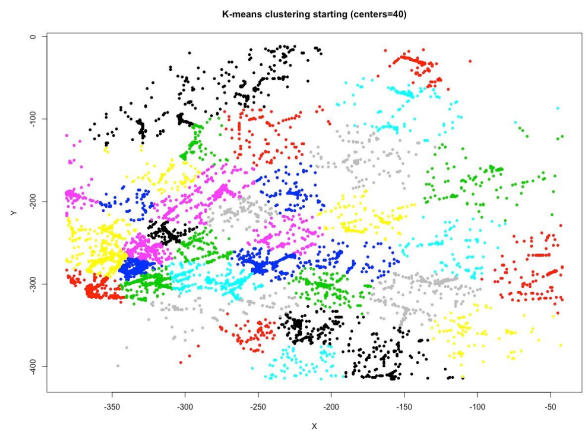

Figure 8.1 clustering of starting points

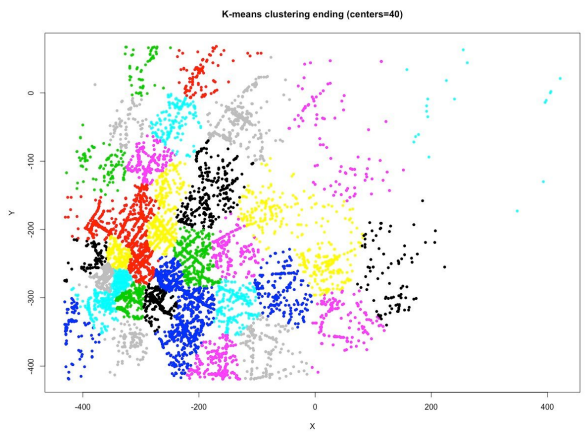

Figure 8.2 clustering of drop off points

MODELS SELECTION

With all features generated and integrated, we fit them into different algorithms. Predictive models we considered are: linear regression, lasso regression, XGBoost regression and random forest regression. We use RMSPE (Root Mean Square Percentage Error) as the measurement of accuracy.

LINEAR REGRESSION MODEL

Many regression models with various regressors are attempted. From the very beginning, aspects that we have considered are:

- Predict trajectory length and duration independently, and sum them up as the predicted taxi charge
- Coordinates of pick up & drop off points are directly fitted into linear regression model as numeric values
- Taxi ID is also attempted to fit into a linear model as numeric values
- Second order of the logarithm of straight distance, along with the first order, is attempted. This is proved to be statistically significant and decreases the RMSPE greatly.
- Take all prepared features (i.e., date, hour, driver's diligence etc.), apply 1 hot encoded vector transformation to all categorical variables. Predict the trajectory length first, then fit the predicted trajectory length into a second linear model to predict the velocity.

We conclude that the model with all variables (with all categorical features 1 hot encoded) our optimal model. We obtain a training cross-validation RMSPE of 0.2578038 and a testing RMSPE of 0.25863.

LASSO REGRESSION MODEL

With all possible features obtained mentioned above, we consider take a step further to consider that with so many features in the linear model (generated by 1 hot encoded vector), it could lead to overfitting on the training set leading to very high values of weights.

Trying to overcome this problem, we attempted LASSO Regression. Applying the same method, we predict the logarithm of trajectory length firstly with all features available in testing set, then with predicted trajectory length as one of features, logarithm of velocity is predicted.

Worthy mention, when we cross validating the optimal regularization parameter lambda, we use Mean Square Error (MSE) as the performance measurement because RMSPE not found in their measurement type.

We obtain a training RMSPE of 0.2665651, testing RMSPE: of 0.272541. We did not go further beyond this since the performance hardly exceeds the basic linear regression model.

XGBOOST REGRESSION MODEL

When Generalised Boosting Method on Regression is introduced to us on the class, we proceed to using the boosting method on linear regression. Since the “gbm” method takes rather long time to run, we choose “xgboost” instead - Extreme Gradient Boosting Training. It is of the similar idea of GBM introduced on the class.

Similarly, with all data included in the model, we predict the trajectory length firstly. Then with the predicted trajectory length as one of the features, we proceed to predict the velocity. Lastly using predicted logarithm of trajectory length subtract by the logarithm of velocity, we obtain the logarithm of duration.

With 500 rounds of cross validation, during each round a random parameter list is generated. We determine the optimal learning rate to be 0.25. As a result, we obtain a training mean RMSPE of 0.2423479 and a testing RMSPE of 0.24176.

*RANDOM FOREST REGRESSION MODEL**

The last model we attempted is random forest. We still apply the same approach described above: predict the trajectory length first, then with the predicted value, predict the velocity.

Since we have many dummy variables (1 hot encoded vector), and random forest requires tremendous computation resource, we lack the capabilities to run many rounds of cross validations. Instead, we split the training dataset into two with 30% as validation set. Then we fit the remaining 70% of training data to the model. Unable to tune parameters using cross validation, we start by “ntree=200” and “nodesize=15”. Two random forest models (both trajectory and velocity) take in total around 6 hours to complete. We obtain a validation RMSPE of 0.1808429 and a testing RMSPE of 0.24774. We conclude that we have overfitted.

We then take a step back, and start with a basic random forest. Still, we follow the approach that predicting trajectory length first, with whose prediction values as one of features for velocity prediction. We avoid overfitting, we choose not to include all driver’s quality features, as well as pick up & drop off points clusters. Instead, we feed raw data points, i.e., coordinates, into the model, as well as driver’s ID. In addition, to facilitate the timing required, we decrease “ntree” to 50 and increase “nodeside” to 30. After many rounds of testing, we obtain a best training RMSPE of 0.2095589 and best testing RMSPE of 0.22751.

CONCLUSION & LIMITATION

With all of our analysis completed above, we conclude that the random forest model with Taxi ID and raw pick up and drop off coordinates in the random forest our best model. Throughout our analysis, we have observed the relationship and pattern of driver’s identity, analyzed the patterns behind the city’s map as well as the positive correlation behind the trajectory length and duration. However, due to the lack of features, even with all possible feature engineering and different algorithms, prediction accuracy may not display a reasonable good fit. Future studies

may focus on more detail-oriented data collection, and take a step further on driver's profiling. Also, if available future studies could tune parameters more systematically by introducing cross validations into the random forest.