Taxi Travel Time Prediction

Lan Liu, Wenlu Gou

March 16, 2018

Abstract

In this report, we focus on a fundamental supervised learning problem, which is the taxi travel time prediction between an origin and a destination. The available data consists of all taxi trips running in New York city within 2015 with limited features. We presented the construction of new reasonable features, and applied machine learning models such as Linear Regression, Regression Tree and Support Vector Regression. Furthermore, the ensemble methods such as Random Forest and Gradient Tree Boosting are also applied.

The best model we found is the ensembling methods, both the Random Forest Model and Gradient Tree Boosting Model significantly outperforms the single model. The main reason is that the linear regression model fails to capture the nonlinearity, while the single tree model tends to overfit or underfit, however, the ensemble method combines the weak models and works in a robust way. The best model achieves RMSE to be 5.54 minutes, with 25.7% mean relative error rate.

Keywords: travel time prediction, Linear Regression, Regression Tree, Support Vector Regression, Random Forest, Gradient Tree Boosting.

1 Introduction

The GPS devices track the operation of vehicles and provide navigation service, meanwhile a significant amount of trajectory data are collected. Nowadays, one can find the available GPS based cab services such as Uber, Lyft, etc. For example, the Lyft has released all the taxi trips in New York City within 2015, and make a Taxi Trip Time Prediction Challenge run by Kaggle.com, which aims to build a predictive framework that is able to predict the travel time of a taxi. The dataset contain more than 12 million taxi trips within the year 2015.

The project is a supervised data mining task involves predicting the travel time for cab ride. The given data contains the start and end location in terms of longitude and latitude, the start time in the form of epoch timestamp. The output feature is the duration of the travel time.

Multiple work in literature has investigated the travel time prediction of a taxi services. For example, Christ et al.[1] considered linear regression and random forest model to forecast the travel time, Thomas [2] investigated the ensemble approach, Leone et al.[5] took the movements of the vehicles into account and treated it as semantic variables, which requires more feature and achieved better performance. The work of Arnab et al.[6] specifically resolved the case with the stream of data. Hong et al.[3] has developed a neighbor based method which outperforms more sophisticated models.

In this report, we covered 5 regression model and compare their performance. The first section covers the literature review and introduction of the project, the second section mainly aims to clean the data and construct new features such as the distance and the starting hour of the journey, which is correlate closely with the duration. The third section investigated several regression models such as Linear Regression, Regression Tree and Support Vector Regression, and further developed ensembling models which outperform the single models. The forth section presents the main result and conclusion, and the last section involves the discussion such as the future work that we can do.

The main conclusion of this report is that ensembling techniques are really well and tend to outperform a single learner which is prone to either overfitting or underfitting. After generating thousands or hundreds of them, we can combine them to produce a better and stronger model.

2 Data Preprocessing

We used the data provided by Lyft to build our model. Each row in the dataset corresponds to a single journey. There are approximately 12,900,000 observations in New York City from 2015.1.1 to 2015.12.31. Because of the data set is humongous and the computational limitations, subsets of the data were used. We randomly sampled 50,000 observations, of which 40,000 were used for training and 10,000 for validation.

The features include start longitude, start latitude, end longitude, end latitude, start timestamps which indicates the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970. Below is a screenshot of the data.

```
> head(data)
  row_id start_lng start_lat   end_lng  end_lat start_timestamp duration
1      0 -74.00909  40.71382 -74.00433 40.71999      1420950819      112
2      1 -73.97118  40.76243 -74.00418 40.74265      1420950819     1159
3      2 -73.99496  40.74508 -73.99994 40.73465      1421377541      281
4      3 -73.99113  40.75008 -73.98861 40.73489      1421377542      636
5      4 -73.94551  40.77372 -73.98743 40.75571      1422173586      705
6      5 -73.99283  40.70148 -73.99290 40.71998      1422173589      509
```

Figure 1: head of the original data

2.1 Feature Construction

From Figure 1, there are only 5 features (the row id is excluded) that can be used to predict the duration. In addition to the features given, we can generate several important features.

- Distance(km)

  The distance between the start and the end point is important, since the travel time and the distance have strong correlation with each other. We have computed the Mean Haversine distance between any two points given the longitude and latitude, which is defined as follows:

$$a = \sin^2(\frac{\phi_2 - \phi_1}{2}) + \cos(\phi_1)\cos(\phi_2)\sin^2(\frac{\lambda_2 - \lambda_1}{2})$$

$$d = 2r\arctan(\sqrt{\frac{a}{1-a}})$$

  where $r = 6378.145km$ is the radius of earth, ϕ_1, ϕ_2 are the latitude of two points in radians, and λ_1, λ_2 are the longitude of two points in radians.

- Day and Hour

  The epoch timestamp itself does not have a clear relationship with the duration. However, the day within a week and the hour within a day can be easily created from the timestamp. The fact is that the speed of the traffic flow varies significantly from day to day and hour to hour, which demonstrate that we should include these two features in the dataset. We have created the Day feature with 7 levels from Monday through Friday, and the Hour feature with 24 levels from 0 to 24.

- Weekend and Busy Hour(categorical)

  In addition to Day and Hour, the categorical feature Weekend/Weekday, Busy/Free Hour have also been constructed. The reason is that we can expect more traffic on Weekday than the Weekend, and the traffic flow is more smooth on free time compared with busy hour, where the traffic get dense. We have defined 7AM-10AM, 4PM-8PM as busy traffic hour and the rest as smooth traffic hour.

These new defined features combined with the original features constitute the feature space, as illustrated in Figure 2.

```
> head(Train%>%select(-row_id))
  start_lng start_lat   end_lng  end_lat start_timestamp duration  distance  weekday     day hour busy
1 -73.99709  40.73742 -73.99406 40.72504      1433663802      487  1.401514   Sunday weekend    0 free
2 -73.96141  40.77434 -73.98355 40.78424      1434244379      655  2.167330 Saturday weekend   18 busy
3 -73.99810  40.74100 -74.03619 40.63644      1429949464     2475 12.075242 Saturday weekend    1 free
4 -74.01166  40.71343 -73.86561 40.77085      1436479989     2419 13.878685 Thursday weekday   15 free
5 -73.99168  40.74897 -73.99168 40.74897      1446873133        9  0.000000   Friday weekday   21 free
6 -73.97123  40.79803 -73.95965 40.78113      1426179129      695  2.119374 Thursday weekday    9 busy
```

Figure 2: head of the data after feature construction

2.2 Data Cleaning

In the data set, we observe many anomalous trips, which will cause error in the estimation. For example, there are data with travel time which is less than 15 seconds, or with distance less than 150m, these data are not likely to be generated in a normal taxi ride, and they might be created accidentally by the wrong read of GPS.

Figure 3 presents the distribution of distance and duration time, we can see that the distribution is left skewed, i.e., most journey of taxi ride has relative low duration and distance. At the same time, we can also observed a cluster of data at the left boundary which break the smoothness of the plot.

In the dataset, we eliminated all observations with duration less than 15s or distance less than 150m.

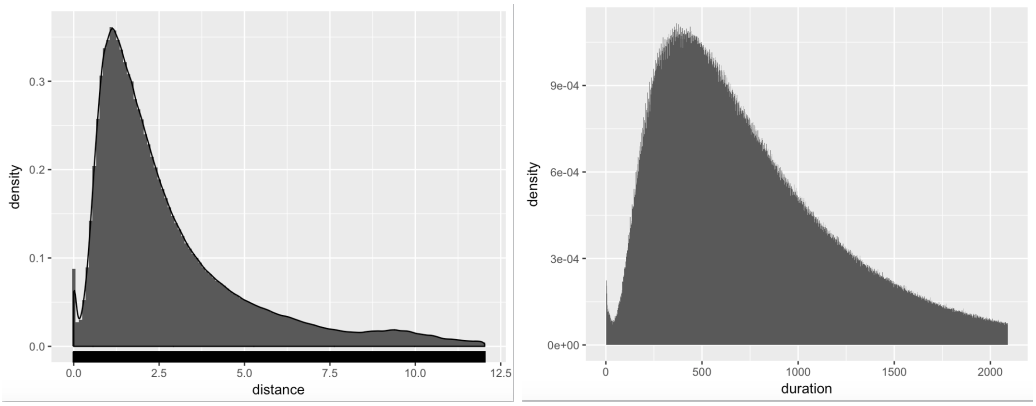


Figure 3: Left: the distribution of duration time. Right: the distribution of distance

2.3 Data Visualization

By visualizing the data, the possible relationship between features can be observed.

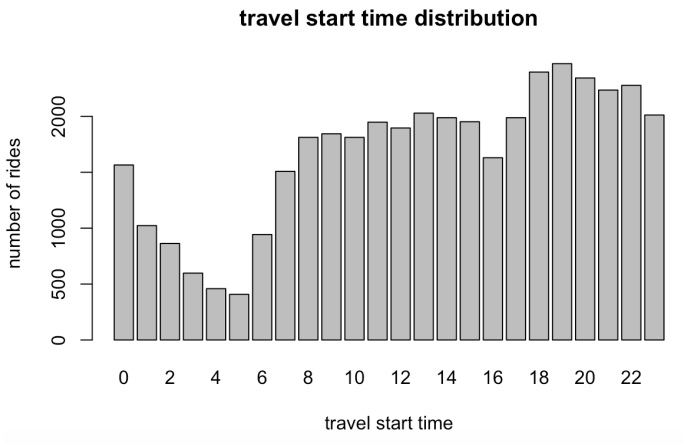


Figure 4: Travel Start Time Distribution within a Day.

Figure 4 displays the frequency of the taxi rides in the 24 hours of a day. It can be observed that the taxi ride is quite intense in New York City from 8am to 23pm, and gets smoother from 1am to 7am, which is within expectation. This also indicates that the new feature Hour that we have constructed is informative and may affect the duration of the travel ride.

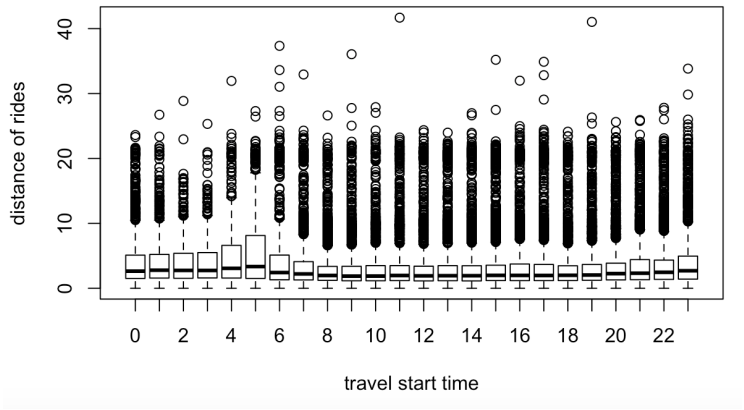


Figure 5: Trip Distance versus Travel Start Time.

Figure 5, instead, shows the relationship between the distance and the start time of a ride. The graph indicates that most journey is less than 20km. Meanwhile, the journey in the midnight tends to be long, while the distance of the journey in the day time varies a lot.

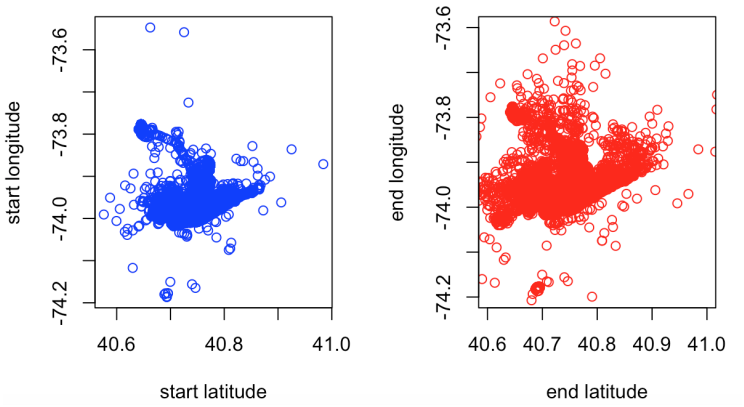


Figure 6: Coordinates of the pick up and drop off points.

Figure 6 shows the coordinates of the start point and end point of the journey. It can be seen that there is a densed region which contains most trip, which is the busy City region, and there are also remote data points spread outside the central part. From this distribution, it is reasonable to conjecture that the pick up and drop off location also affect the duration of the taxi ride in a nonlinear way. The more dense the traffic is, the lower speed that a vehicle can move, which then impact the travel time.

In the next section, we will build the machine learning models to predict the travel time based on the feature space.

3 Machine Learning Models

3.1 Linear Regression

First we want to start from the most basic model, the linear regression model; to find the relationship between travel time and other variables in the data set.

Since we define the variables Weekend and Busy Hour based on the variables Day and Hour, so Weekend and Day are highly correlated, the same for Busy Hour and Hour. We cannot put all those four variables in the same regression. In this way, we do two linear regressions separately. The fist model, we don't include the variables Day and Hour. For the second model, it doesn't contain the variables Weekend and Busy Hour.

$Model1$:

$$Durtation = \beta_0 + \beta_1 Start.longitude + \beta_2 Start.latitude + \beta_3 End.longitude + \beta_4 End.latitude + \beta_5 Start.Time.Stamps + \beta_6 Distance + \beta_7 Weendend + \beta_8 Busy.Hour$$

the inference group: Weekend = Weekday, Busy.Hour = Busy

$Model2$:

$$Durtation = \beta_0 + \beta_1 Start.longitude + \beta_2 Start.latitude + \beta_3 End.longitude + \beta_4 End.latitude + \beta_5 Start.Time.Stamps + \beta_6 Distance + \beta_{7i} Day + \beta_{8j} Hour$$

the inference group: Day = Tuesday, Hour = 0; $i \in [1, 6], j \in [1, 23]$

The adjusted R^2, the Mean Squared Error (MSE) from the train set and the test set for Model 1 and Model 2 are showed in the Table 1

Table 1: Model 1 vs Model 2

	Model 1	Model 2
Adjusted R^2	0.5833	0.6124
MSE Train	182827.9	160836.2
MSE Test	173464.4	169933.2

As we can see, the Model 2 has higher adjusted R^2 and lower MSE from both the train set and test set than the Model 1. Model 2 performs better than Model 1. This result doesn't surprise us. In Model 1, we only consider Weekend and Busy Hour, which are a little bit too general. We lose some information based on this model. However, in Model 2, we contain the variables Day and Hour, which covers more details about each taxi trip. Those detail information help we improve the model performance.

The next step we want to check the Model 2 is overfitting or not. We use the Ridge regression and the LASSO regression.

Ridge regression:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}\beta_j^2 \quad (1)$$

LASSO regression:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}|\beta_j| \quad (2)$$

The Ridge and LASSO regression add the shrinkage penalty; which is the second term in those two formulas above. The goal for us is to minimize both (1) and (2). As we are adding more and more parameters, we reduce the residual sum of squares, but we increase the shrinkage penalty. The advantage for Ridge and LASSO regression is that those two methods will shrink unnecessary close to zero or exactly zero. In this way, we can find the balance between bias-variance trade-off and the model complexity. For Model 2, the estimated coefficients from the linear regression, the Ridge and the LASSO are almost the same. The estimated coefficient for the variable Start Time Stamp is 2.11×10^{-6} from those three methods, which is very close to zero. No other estimated coefficients are shirked to close to zero or exactly zero.We update the Model 2 by removing the variable Start Time Stamp and name it Model 3.

Model 3

$$Durtation = \beta_0 + \beta_1 Start.longitude + \beta_2 Start.latitude + \beta_3 End.longitude + \beta_4 End.latitude + \beta_5 Distance + \beta_{6i} Day + \beta_{7j} Hour$$

the inference group: Day = Tuesday, Hour = 0; $i \in [1, 6], j \in [1, 23]$

The estimated coefficients and corresponding p-value for Model 3 are showed in the Table 2. Most of estimated coefficients are highly significant and reasonable. The coefficient for distance is positive. The long distance will take more time than the short distance trip. The coefficients for Saturday and Sunday are negative. Comparing with Tuesday, the weekend has less traffic, so travel time should be short. The coefficients for weekdays are positive, except Monday, are positive. During weekdays, the traffic is not good, it will take longer time to arrive at destination. For the hour part, the inference level is 12 am. AS we can see, from 1 am to 6 am, all coefficients are negative. From 7 am to 11 pm, all coefficients are positive. The traffic is better in midnight than day time and evening, so we expect shorter travel time during midnight and longer travel during day time and evening. The coefficient of start longitude, end longitude and end latitude are negative. The coefficient of start latitude is positive. It seems that for certain starting point and destination will take shorter time. For some other starting places, it will have longer travel. However, for those 4 coefficient, we don't have a clear explanation.

Table 2: Model 3

Coefficients	Estimated	p-value
Intercept	-8862	< 0.01
Start.longitude	-657.4	< 0.01
Start.latitude	425.4e	< 0.01
End.longitude	-861.4	< 0.01
End.latitude	-1001	< 0.01
Distance	133.8	< 0.01
Weekday.Mon	-68.49	< 0.01
Weekday.Wed	22.76	< 0.01
Weekday.Thu	42.52	< 0.01
Weekday.Fri	28.61	< 0.01
Weekday.Sat	-67.06	< 0.01
Weekday.Sun	-111.8	< 0.01
Hour1	-29.21	0.08
Hour2	-47.46	< 0.01
Hour3	-46.93	0.02
Hour4	-151.4	< 0.01
Hour5	-287.6	< 0.01
Hour6	-171.9	< 0.01
Hour7	25.30	0.09
Hour8	143.2	< 0.01
Hour9	164.3	< 0.01
Hour10	185.9	< 0.01
Hour11	181.1	< 0.01
Hour12	200.8	< 0.01
Hour13	209.9	< 0.01
Hour14	227.9	< 0.01
Hour15	264.8	< 0.01
Hour16	267.3	< 0.01
Hour17	244.9	< 0.01
Hour18	196.8	< 0.01
Hour19	123.9	< 0.01
Hour20	66.55	< 0.01
Hour21	31.07	0.02
Hour22	24.71	0.07
Hour23	21.95	0.12

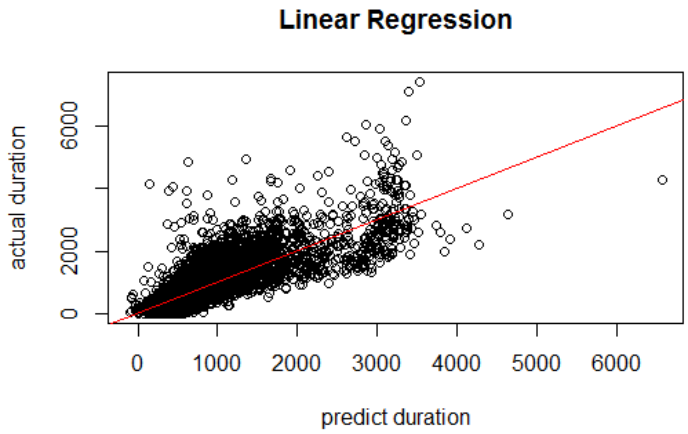


Figure 7: Actual duration versus predicted duration based on the Linear Regression Model 3

For Model 3 the MSE for the train set is 170299.4 seconds and the test set is 161146.7 seconds. The average time difference between the fitted value by using the train set and actual duration from the train set is 274.775 seconds about 5 minutes. Figure 7 shows the predict duration versus the actual duration from the test set. The red line in the graph represents that the predicted duration equals the actual duration. As we can see, a lot of points are far away from that line. At the average level, the time difference between the predicted duration and actual duration in the test set is 271.59 seconds, which is close to 5 minutes. On average, the Model 3 does a good job, but at general level we don't think a simple linear model is a suitable choice.

3.2 Regression Tree

As traffic is aggregated more densely at some locations than the others, the location of the ride will clearly affect the trip duration. This is also shown in Figure6. Although we can not consider all locations of the journey because of the lack of information, we can apply the start and end point of a ride into the model. In the linear regression model, the effect of the location is simply modeled using a linear relationship in terms of the longitude and latitude, however, the traffic is clearly not varying solely based on the magnitude of the coordinates, rather, we want to introduce some nonlinearity effect of the location. An algorithm that can better capture the nonlinearity is the Regression Tree.

Compared with linear regression model, which is a global model that has a single linear predictive formula holding over the entire data, the regression tree works better when the data has lots of features which interact at a nonlinear way, and in which linear model hardly predict well. In the regression tree model, it divides the data space into small subregions, where the interactions are more manageable. We sub divide the region recursively until we get to chunks of spaces that we can fit simple models with. The goal then contains two parts, first is the recursive partition, second is a simple model for each leaf of the tree.

In our tree model, the partition rule is by minimizing the RSS,

$$\sum_{c \in leaves} \sum_{i \in c} (y_i - \hat{y})^2$$

where $\hat{y} = \frac{1}{n_c} \sum_{i=1}^{n_c} y_i$.

Once the regions are defined, for each leaf of the tree, the prediction is simply a constant estimate of the duration. That is, suppose the data points with duration $y_1, .., y_n$ are all the samples that belong to the leaf node, then our estimate is just $\hat{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$ for the sub region, which is the sample mean of the response variable in that cell.

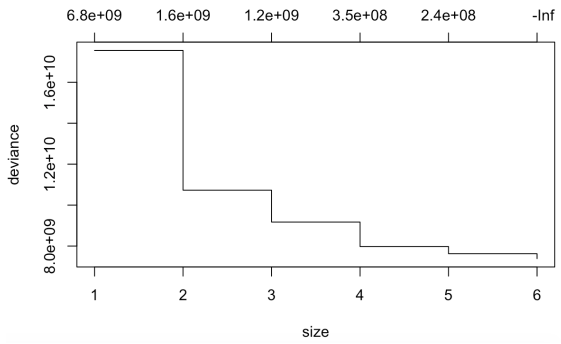


Figure 8: Deviation versus size of Regression Tree Model.

After applying the decision tree model, we get the relationship between the size of the tree and the deviation of the tree, as illustrated in Figure 8, and it clearly shows that the deviance decreases as the tree grows, which is as expected.

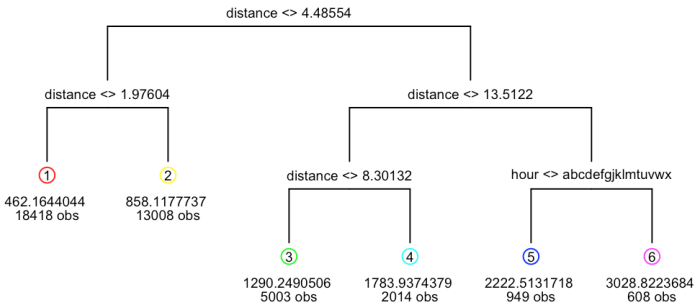


Figure 9: Regression Tree

The Figure 9 shows the plot of the tree, it only involves the feature distance and hour, and abandon all of the rest features, which is not as we expected. Consider that the main reason for using decision tree is to capture the nonlinear effect of the location of pick up point and drop off point, this limit partition prevents us from getting better result than the Linear regression Model, and the MSE of regression tree is 176037.3.

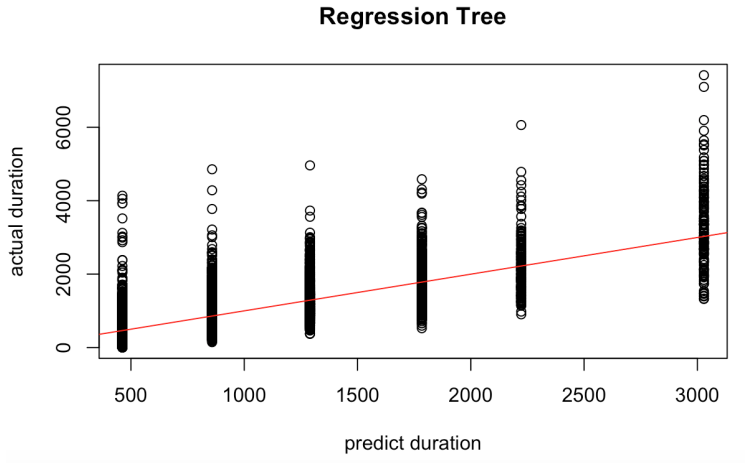


Figure 10: Actual duration versus predicted duration based on Regression Tree Model.

Figure 10 illustrates the comparison between the predicted duration and the actual duration. With only 6 leaves in the tree, there are only 6 values being predicted for all data points. The regression tree model may be underfitting since it applies only 2 feature among the 10 features. We will construct the random forest to prevent the underfitting and generate a strong model based on this single tree model.

3.3 Support Vector Regression

Support Vector Regression(SVR) is another model we can apply to estimate the travel time. SVR works on similar principles as Support Vector Machine classification. One advantage of SVR is that it permits for construction of a nonlinear model without changing the explanatory variables, helping in better interpretation of the resultant model.

The Support Vector Regression Model has the following objective function:

minimize $\frac{1}{2}\|\omega\|^2$ subject to $|y_i - \omega x_i - b| \leq \epsilon$ for some threshold $\epsilon > 0$.

After implementation, SVR predictor turns out significantly reduced the MSE of predicted travel times. The MSE of support vector regression is 125932.2 with 29732 support vectors. The Figure 11 shows the predicted value versus the actual value of the travel duration. The red line in the graph represents that the predicted duration equals the actual duration. The majority of points doesn't separate far away from the red line. However, there still has a small group of points that they aren't very close to the red line.

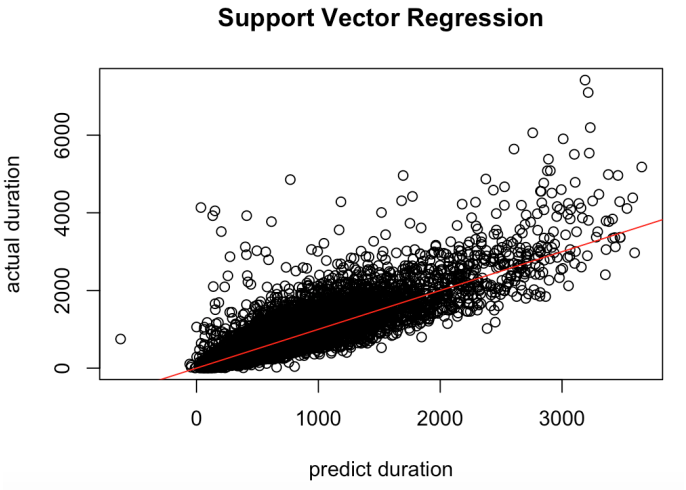


Figure 11: Actual duration versus predicted duration based on the Support Vector Regression.

3.4 Random Forest

In the above regression tree model, the performance is not quite well, and one of the most important reason is that a single tree has a high risk to underfit or overfit. This motives us to apply the Random Forest instead.

In the Random Forest, a large number of trees are used, with each to be a bootstrap sample from the training data, and the prediction of a new observation is made by using the mean of the predictions by all the trees. At each split, only m out of total n splits are chosen to be considered, which efficiently decorrelate the trees. We have chosen $m = \sqrt{(n)}$. As illustrated in Figure 12, the more trees we chose, the smaller the MSE will be. To get the best result, we have chosen the number of trees to be 500.

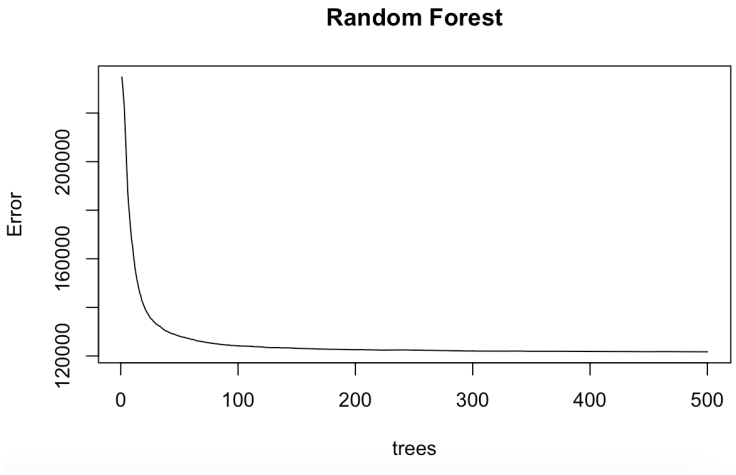


Figure 12: Random Forest

Figure 13 and Figure 14 shows the variable importance for the Random Forest Model on trip duration. As we can see, the variable hour and distance explain the most variance, which is reasonable since the duration mostly rely on how long the journey is and what time to start the travel. Also, the coordinate of the pick up point and drop off point also show their importance with a high InNodePurity and moderate IncMSE, this illustrate that the location of the start and end point are also an important feature to predict the travel time. In contrast, we are not able to extract much information from the $start_timestamp$, which is of low IncMSE and IncNodePurity.

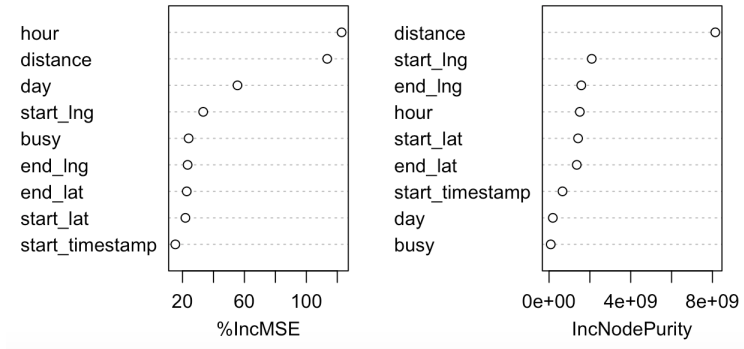


Figure 13: Random Forest variable importance.

```
> importance(rf)
                  %IncMSE IncNodePurity
start_lng        33.35823    2090904942
start_lat        21.86380    1421709099
end_lng          23.28836    1579203737
end_lat          22.71781    1358161755
start_timestamp  15.35700     658765083
distance        113.49590    8141884875
day              55.50326     185447186
hour            122.79495    1507192510
busy             23.94411      83434642
```

Figure 14: Random Forest variable importance, continue

In Figure 15, the plot of predict duration based on Random Forest Model versus actual duration

has been presented. This figure looks much better than the Regression Tree Model solely based on one single tree. Most of the points are close to the red line, where the predicted duration equals the actual duration. The mean squared error decreased to 113148.7 in this random forest model, which is much better than the Regression Tree Model.

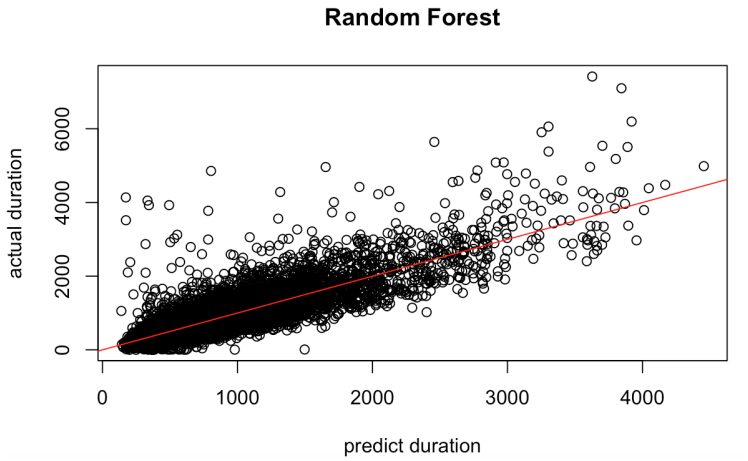


Figure 15: Actual duration versus predicted duration based on the Random Forest Model.

3.5 Gradient Boosting

We have applied the ensembling method Random Forest, and in this section the other ensembling method Gradient Tree Boosting has been used.

The idea of Gradient Boosting is that it assumes an imperfect model F_m at each stage m. The algorithm improves on F_m by constructing a new model that adds an estimator h to provide a better model:

$$F_{m+1}(x) = F_m(x) + h(x) \approx y.$$

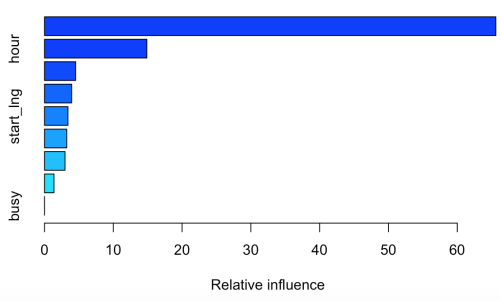
We need to find $h(x)$ which minimize the residual $y - F_m(x)$, which is exactly the gradient of the squared error loss function $\frac{1}{2}(y - F_m(x))^2$. This indicates that each iteration step will be

$$F_{m+1}(x) = F_m(x) + \gamma \sum_{i=1}^{n} \nabla_F (Loss(y_i, F_m(x_i))$$

Therefore, in principle the gradient boosting method is gradient descent, which iteratively minimize the loss function.

In our model, we have specified the depth of each tree to be 8, and iterative 10000 steps with the learning rate $\gamma = 0.01$. It actually demonstrates that the Gradient Boosting is robust and outperform the Random Forest model with a better MSE to be 110514.3.

Figure 16 shows the relative importance of the features, and similar with the Random Forest model, the two most important features are the distance and hour.




```
> summary(gbm.duration)
                                var      rel.inf
distance                 distance 65.613423752
hour                         hour 14.880295608
end_lat                   end_lat  4.531825134
end_lng                   end_lng  3.953391910
start_lng               start_lng  3.412405144
start_timestamp start_timestamp  3.238017681
start_lat               start_lat  2.984039784
day                           day  1.381239022
busy                         busy  0.005361965
```

Figure 16: Relative Influence of features in gradient boost.

Figure 17 presents the partial dependence plot of duration on distance and hour. It shows the marginal effect of the selected variables by opting out the other variables. As distance increases from 0 to 30km, there is a good linear relationship between the duration and the distance, however, as the distance being larger than 30km, the curve tends to be flat. For the hour, it clearly shows that it takes longer time to drive at busy hours compared with smooth hours.

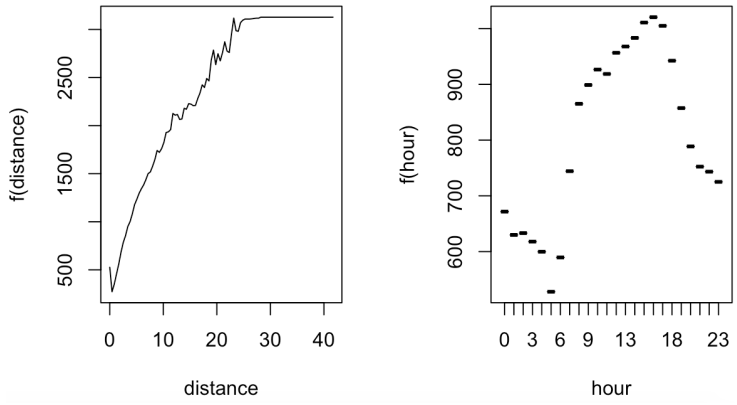


Figure 17: Left and Right are the partial dependence on the distance and hour.

Figure 18 presents the actual duration versus predicted duration based on the Gradient Boost Model. We can see that the performance is pretty well, and the observations evenly distributed around the $y = x$ line. Most of the predicted durations don't have huge time difference with the actual durations.

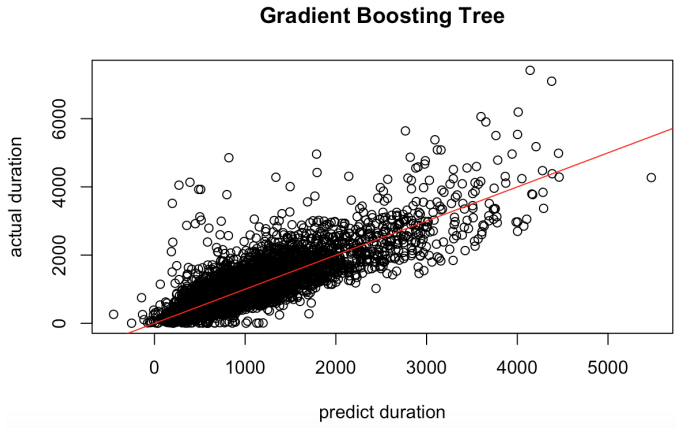


Figure 18: Actual duration versus predicted duration based on the Gradient Boost Model.

4 Evaluation of the Models

In this section, we mainly compare the performance of the models that we presented in last section. Several evaluation metrics have been chosen, such as root of mean squared error(RMSE), mean absolute error(MAE) and mean relative error(MRE).

$$RMSE = \sqrt{\frac{\sum_i |y_i - \hat{y}_i|^2}{n}}$$

$$MAE = \frac{\sum_i |y_i - \hat{y}_i|}{n}$$

$$MRE = \frac{\sum_i |y_i - \hat{y}_i|}{\sum_i |y_i|}$$

where $\hat{y_i}$ is the travel time estimation of trip i and y_i is the actual duration,

The following Table 3 shows the result of comparison between the models: Linear Regression (LR), Regression Tree(RT), Support Vector Regression (SVR), Random Forest(RF), Gradient Boosting(GB). The result indicates that the ensemble methods RF and GB outperform the rest methods. The method with the best performance can achieve RMSE 332.4s on the test dataset, which is 5.54 minutes, and the mean relative error is around 25.7%. This result is pretty good, considering that the features provided in the data set is very limited.

Table 3: Model Performance Evaluation

Model	LR	RT	SVR	RF	GB
RMSE Train (s)	412.7	429.7	363.0	348.2	274.5
RMSE Test (s)	401.4	419.6	354.9	336.4	332.4
MAE Train (s)	274.8	292.0	219.2	219.0	184.7
MAE Test (s)	271.6	288.2	217.1	216.7	213.0
MRE Train (%)	32.6	34.69	26.04	26.02	21.94
MRE Test (%)	32.8	34.79	26.21	26.15	25.71

5 Discussion

Based on the last section, there is still 5.54 minutes error even for the best model. There is still space to improve the performance by a more sophisticated model or by providing more information to build the model. The following is the future work that we can consider:

- More location information

  We only have the start longitude, start latitude, end longitude and end latitude provided as path information. Those cannot give us accurate location information, so we cannot map each taxi trip start point and end point in the New York City. For certain locations, all roads are crowded, like downtown, will take more time to arrive the destination than other part in the city. We also don't have location information about the travel route. Different routes will also lead different duration. Those missing information cause significant errors. Even the best model is Gradient Boosting, it still has mean relative error above 20% on both training dataset and testing dataset.

- K-nearest neighbors model

  The other potential suitable model for predicting the taxi travel time is the K-nearest neighbors (KNN) model. We can use the K nearest start points and k nearest end points to estimate the duration. However, the main challenge is that how to define the distance between two different points. We also have categorical variables Day, Hour, Weekend and Busy Hour, which play an important role in duration estimation. Transforming those four categorical variables into numeric variables is also a big challenge for us, since the KNN algorithm doesn't work with categorical variable.

- Outliers

  Outliers are always the tricky part in the data cleaning process. In this paper, we define outliers just based on our life experience. Super short travel time or travel distance doesn't make sense in real life. It is the same for super long travel time and travel distance. However, in this paper, since we don't have enough location information, we don't remove those potential outliers. Those potential outliers might cause high bias during the model construction process.

  Another way we can do to remove the outliers is by constructing a regression model between duration and distance, which has a strong correlation with each other. If an observation is away from expectation too much, we may safely group it as an outlier.

- Clustering

  Based on Figure 6, there is area which is the urban are and quite busy and which is remote area which locates only a few travel rides. We may need to cluster the regions based on the traffic density. After that, we can develop different models based on the density. The unsupervised learning algorithm can be considered for this case, such as clustering.

References

[1] Christophoros A., Delara F., and Antoine F. A., *Fare and Duration Prediction: A Study of New York City Taxi Rides*. 2016.

[2] Thomas Hoch, *An Ensemble Learning Approach for the Kaggle Taxi Travel Time Prediction Challenge*.

[3] Hongjian Wang, Zhenhui Li, Yu H. Kuo, Dan Kifer, *A Simple Baseline for Travel Time Estimation using Large Scale Trip Data*.

[4] Chun H. Wu, Jan M. Ho, D.T. Lee *Travel Time Prediction with Support Vector Regression*.

[5] Leone P.M., Marco A.C., Marcelo T.M.C, *Travel Time Prediction using Machine Learning*.

[6] Arnab Kumar Laha, Sayan Putatunda *Travel Time Prediction for GPS Taxi Data Streams*.