Capital One Data Science Challenge

NY Green Taxi September 2015 Data Analysis

Qi Li

March 27, 2018

1 Overview

In this report, I analyze the NYC green taxi data for the data science challenge. The whole report is organized as follows. In section 2, I introduce the general software and the essential libraries I have used for this project. In sections 3 and 4, I perform data cleaning and feature engineering, the procedure is explained step-by-step. In sections 5 and 6, I perform various analysis to investigate the distribution of the trip distance and average trip speed in the dataset. Finally, in section 7, I have constructed a tree-based predictive model for the tip rate of the taxi rides.

2 Methods

For this project, I used Python 3.6 under the Anaconda Spyder IDE. Note that the final submitted code package should run with or without the IDE. I have used the "pandas" and the "numpy" libraries for data manipulation and the "pickle" library for fast dataframe save/load. The lognormal/normal distribution curve fitting, statistical indexes, ANOVA were performed with the "scipy.stats" module. The predictive model was constructed with the "scikit learn" library. The September 2015 NY green taxi data was downloaded from `https://s3.amazonaws.com/nyc-tlc/trip+data/green_tripdata_2015-09.csv`. Some important information is extracted from the "NYC Taxi and Limousine Commission (TLC)" website `http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml`.
Note that I have put "random_state = 0" whenever I used a random number generator, to produce consistent results throughout this report.

3 Overview and Preliminary Data Cleaning

After read in the .csv file with pandas, I have checked the size of the dataset. There are 1,494,926 rows and 21 columns in total. In this section, I will explain my strategy for preliminary data cleaning on different features.

3.1 Missing values

Only the "Ehail_fee" and the "Trip_type" columns have missing values. The whole "Ehail_fee" column is "NaN", so I dropped the whole column. There are four missing values in the "Trip_type" column. The value of this columns is either 1.0 or 2.0. Since most of the trips (97.8%) have 1.0, I filled in the missing values in this column with 1.0.

3.2 Invalid values for the price related columns

All the columns related to "prices" are already in float64 format, however, a small portion of the rows have negative numbers, which is probably due to some human error when uploading the data. Therefore, I decided to take the absolute value of all these columns.
The "total_amount" column should always be greater than 2.50, which is the base rate of the NYC green taxi. There are about 0.3% rows with total fare less than 2.50, and I dropped them all.
Next, the "total_amount" column should be equivalent to the sum of the other columns describing different parts of the total amount of the fare. I used this criterion as a sanity check for the "price" related columns of each row. Overall, there are only 0.05% rows that have the inconsistent total fare, and I dropped all these rows.
The "Extra" column is discrete with valid numbers 0.0, 0.5 or 1.0. After performing the steps above, only two rows still have invalid "Extra" value, and I dropped them.

3.3 Columns with categorical data

I cleaned up the "Passenger_count" column by replacing the zero values with 1, which is the most frequent value. Then, I manually encoded the binary categorial column "Store_and_fwd_flag" to 0 and 1.
The "RateCodeID" column should only contain rating from 1 to 6. I substituted the invalid values with 1, which is the most frequent value.

4 Feature Engineering and Further Data Cleaning

In section 3, I performed preliminary data cleaning steps. Only obviously corrupted data are dropped/changed. Throughout this report, I will define the dataset after performing these preliminary steps from section 3 as the new "raw_data". In this section, I have performed feature engineering to select the most important features related to my model of prediction in the modeling section 7. I also extracted some new feathers for further data cleaning to remove the corrupted values from the dataset. To clarify, I define the dataset after performing all the operations from this section as the "cleaned_data".

4.1 New datetime features

Since all the transactions happened in September of 2015, the year and month information in the datetime features is trivial. I extracted 4 new features: "Week" (1st through 5th week), "Day_of_week", "Day_of_month", and "Hour" from the datetime information. Furthermore, I calculated a new column "Time_Span" (in the unit of seconds) describing the overall duration of each taxi trip.

4.2 Average speed

Based on the "Time_span" and the "Trip_distance" columns, I created a new "Speed" column (in units of mph). The "Time_span" column has a small portion of invalid values (either close to 0 or as large as a day), so does the "Trip_distance" column. The rows with wrong value on either column can be filtered by setting a mask on the "Speed" column. Most of the taxi trips are in New York City, where the speed limit is 25 mph based on `http://www1.nyc.gov/nyc-resources/service/2508/speed-limit`. Of course, one can go faster on the highways outside the city. I have set an average "speed limit" of 40 mph for this project. As shown in Figure 6.1, the portion of trips with an average speed of 40 mph is already negligible. On the other end, the average human walking speed is 3.1 mph according to `https://en.wikipedia.org/wiki/Walking`. The average speed of a taxi trip should probably be greater than the walking speed to make sense. Therefore, I have set a mask of > 3.1 mph on the average speed to filter out the rows with trip distance data close to 0 or absurdly large time span. After performed the filtering on the average speed, I further removed a few outliners with abnormal "Time_span" or "Trip_distance" values. The distribution of the average speed is shown in Figure 6.1, it has a mean of 12.99 mph and a median of 11.85 mph. The overall distribution has a rough Gaussian shape with a moderate right skew. Further analysis of the distribution is discussed in section 6.

4.3 Average fare per mile

To removed the corrupted data in the "Trip_distance" and the "Total_amount" columns, I created a new "Fare_per_mile" column for the average fare per mile for each trip (in units of dollar per mile). According to `http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml`, The total fare F_{tot} of the standard city rate ("RateCodeID" = 1) can be expressed as:

$$F_{\text{tot}} = 2.50 + 0.50 \cdot \text{floor}\,(5d) + F_{\text{waiting}} + F_{\text{misc}} \tag{4.1}$$

where d is the total trip distance, F_{tot} is the fare for waiting in busy traffic and F_{misc} contains all other surcharge and tax. Equation 4.1 states that F_{tot}/d is strictly greater than 2.50 dollars/mile. On the other hand, I filtered out the trips with the F_{tot}/d ratio greater than 20.0 dollars/mile from the dataset. The sanitized average fare per mile data have a mean of 6.39 dollar/mile and a median of 5.83 dollar per mile, which indicates a right skewness. The distribution of the "Fare_per_mile" column is shown in Figure 4.1. The overall distribution obeys roughly a lognormal function, which will be discussed in more detail in section 5.1.

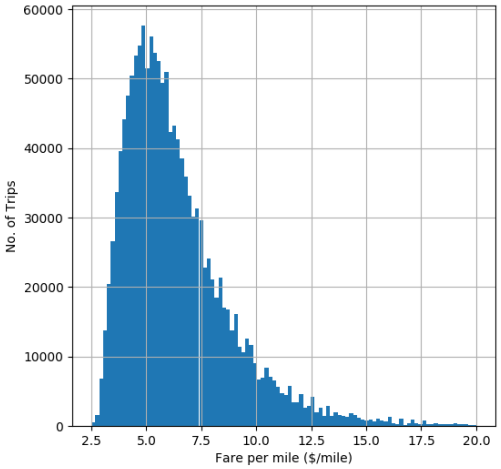

Figure 4.1: Histogram plot of the average fare per mile data.

Note that although the "Speed" and the "Fare_per_mile" features are helpful for data cleaning and understanding the overall distribution of our data, they should not be blindly added into the predictive models because they are merely redundant variables which depend on the other two columns of the data.

5 Distribution of Trip Distance

In this section, I perform analysis to the trip distance. Table 5.1 lists several important statistical indexes. As shown both in Table 5.1 and Figure 5.1, the raw data contains a significant portion of noisy values, the structure of the distribution cannot be read from the raw data readily. The cleaned data have more reasonable statistical indexes, whereas the mean and median trip distance only take negligible change. In the discussions below I will focus on the result of the cleaned data.

5.1 Distribution of the overall trip distance

Table 5.1: Statistical indexes of the trip distance.

Data	mean	std. dev.	median	skewness	kurtosis
Raw	2.97	3.07	1.98	8.45	1028.75
Cleaned	2.97	2.77	2.01	2.21	6.14

The distribution exhibits a clear right skewness with a peak at around 1 mile. The deviation from the standard normal distribution suggests that a simple random variation cannot explain

the trip distance distribution. According to Ref. [1, 2], a series variable is more likely follow a normal distribution if it is determined by an additive process (e.g., the number of heads through a sequence of coin tosses). On the contrary, it tends to follow a lognormal distribution if it is determined through a series of multiplicative processes (e.g., the size of a city, farm, population distribution of bacteria growth). Based on the observation on the histogram and the theorem, I bring up two hypotheses:

1. The deviation from the normal distribution suggests some important correlation within the data.

2. The trip distance data follow a lognormal distribution.

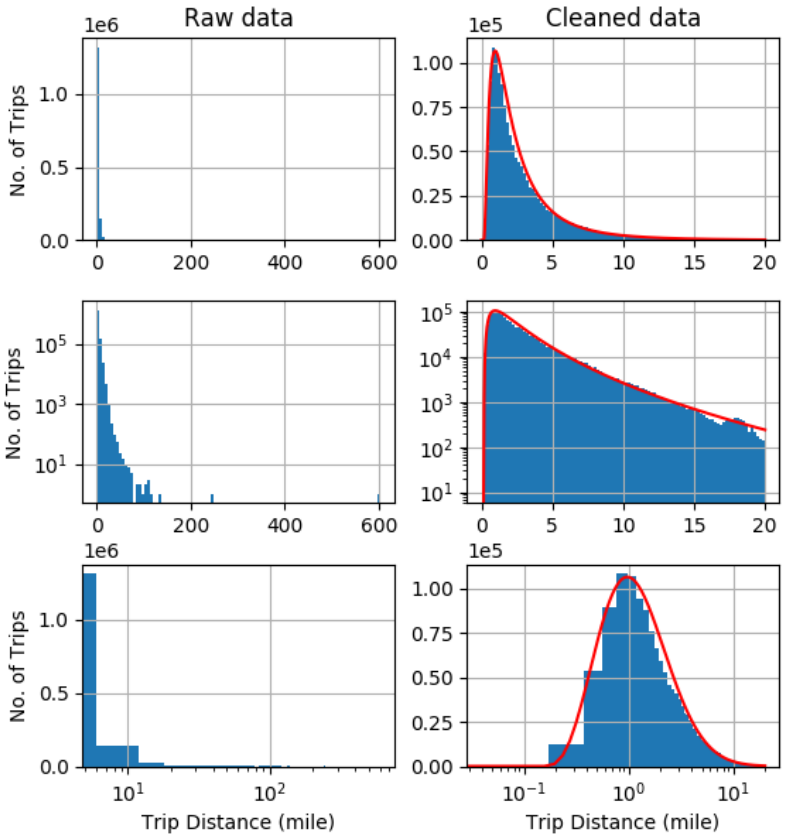

Figure 5.1: Histogram plot of the distribution of the trip distance. The subplots on the left are plotted from the raw data, while the ones on the right depict the distribution of the cleaned up data using strategies in section 4. Top, middle, bottom panels are plotted with normal, semilogy, and semilogx scale, respectively. The solid red lines on the clean data subplots illustrate the probability density function (pdf) of a fitted lognormal distribution.

As shown in Figure 5.1, the trip distance distribution falls onto a fitted lognormal probability density function (pdf) curve nicely. The semilogx plot on the lower right panel shows that the logarithm of the trip distance follows a normal distribution. Although the multiplicative process determining the trip distance is not as apparent as the other examples, we can probably understand it as the following way. If a variable depends on the "historical" information of itself (e.g., the size of a city depends on its previous size, any change in size tends to be adding/subtracting a portion of that previous size.), we can regard that it is determined by multiplicative processes. Since many people take taxi rides on a regular basis (e.g. taking taxi to commute), the "history" of the trip distance highly impacts the future rides. The "fat" tail on the lognormal distribution might be a result of people taking a taxi for long distance commute.

5.2 Trip distance grouped by the hour of the day

Figure 5.2 shows the total trip distance grouped by the pickup hour of the day. The median is lower because of the overall right skewness of the distribution. There is a clear peak around 5 am in the morning, and a second peak at around the midnight. My hypotheses here are:

1. The 5 am peak is associated with trips for long distance home to work (H→W) commute. From 5 am to 8 am (end of the morning commute), the average trip distance decreases almost linearly because people who live closer to work tend to leave home later in the morning. Since the hour of day indicates the pickup time, and the time people leave their office in the afternoon does not have a strong correlation with their H→W distance, such a peak would not show up correspondingly around the afternoon rush hour. In fact, there is a minimum average trip distance around the rush hour. When the traffic is the worst, people probably only want to taxi for short trips (e.g., gather together to a place close to their office for dinner after work), because no one wants to get stuck on the street for hours! Moreover, If we group the trip distance by the "drop off" hour of the day, we could probably see a peak at around 9 pm that corresponds to the 5 am peak from people commute a long distance.

2. The midnight peak is probably associated with some occasional trips. For example, taking a taxi back home from a party, taking a taxi from/to the airport. Such trips tend to be longer than the majority of commute taxi trips.

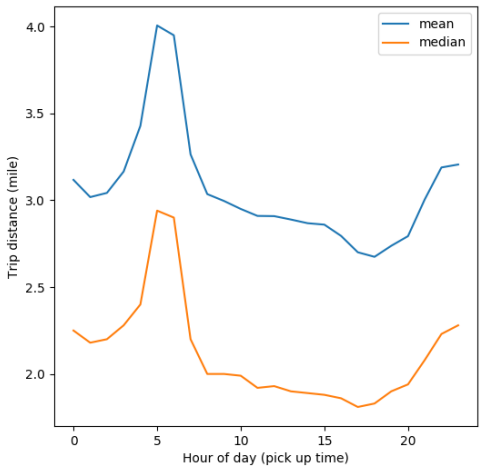

Figure 5.2: Illustration of the trip distance grouped by the taxi pick up hour of day.

5.3 JFK airport trips

According to `http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml`, the "RateCodeID" column indicates the types of the trips. The from/to JFK airport trips have a "RateCodeID" of 2. An alternative approach is, of course, to filter by the longitude and latitude of the trips. There are 4435 JFK trips in the original data, 4417 in the preliminarily cleaned data and 1897 in the cleaned data. The reason why more than half of the JFK trips were dropped is: 1) over 1,800 of the JFK trips in the raw data have zero trip distance as indicated in Figure 5.3, 2) A significant portion of the trips have long trip distance ($>$ 20 miles) which were filtered out in section 4. As illustrated in Figure 5.3, the trip distance has a clear peak at around 18 miles, which is roughly the distance between the JFK airport and Manhattan. The total fare

is always greater than 52 dollars which is the base rate of JFK airport trips. In the raw data plot, it shows a clear peak at near the base rate. It also has two sub-peaks at roughly 58 dollars and 71 dollars, which are probably associated with two different routes with different tolls. Note that although I filtered out about a half of the JFK trips, the portion is only 0.16% of all the total trips. However, if we are investigating the total fare of the JFK trips alone, the raw dataset should give a more accurate description.
Table 5.3 lists the statistical indexes of the total fare of the JFK trips, and the distribution deviates far from normal. The mean of the total fare in the raw dataset is 59.42 dollars. It is higher in the cleaned dataset, probably because many trips with zero distance and a total fare close to the base rate (52 dollars) are filtered out.

Table 5.2: Statistical indexes of the total fare of the JFK trips.

Data	mean	std. dev.	median	skewness	kurtosis
Raw	59.42	7.54	58.34	0.77	-0.53
Cleaned	64.15	6.49	63.36	0.03	-0.83

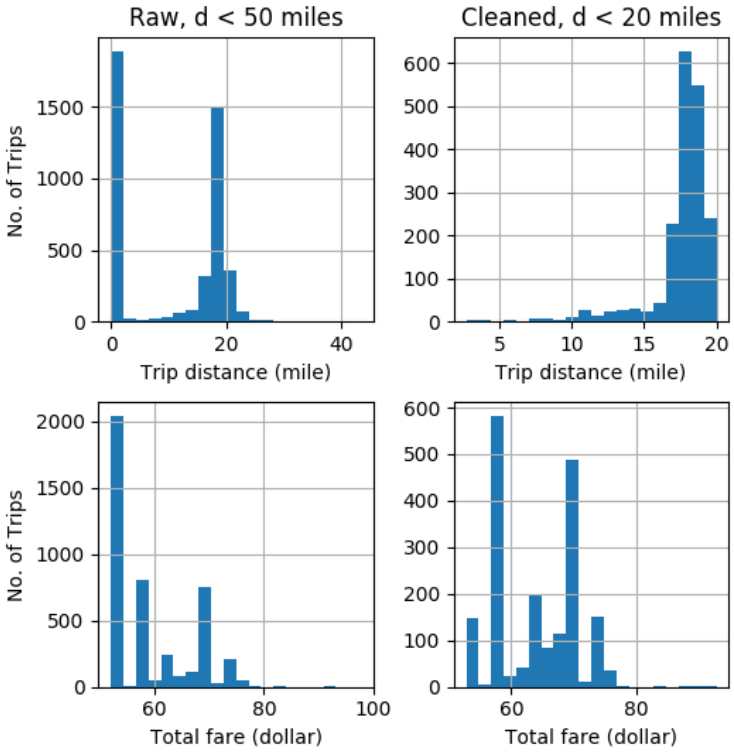

Figure 5.3: Histogram plot of the distribution of the trip average speed.

6 Distribution of the Average Speed

In this section, I will discuss the distribution of the average trip speed. This section covers the "Question 5 – Option A" in the challenge t.

6.1 Overall distribution of average trip speed

The average trip speed has a mean of 13.03 mph and a median of 11.86 mph. The overall distribution of the average trip speed is shown in Figure 6.1, where I used the red line to represent the lognormal fit and the black line to represent a fitted normal distribution with an adjusted center at 11.0 mph. The actual distribution lies roughly "in between" the two fitted curves. The reason for this observation could be that most of the trips are inside the city, with speed limit and the traffic apply an additional constraint over the average speed.

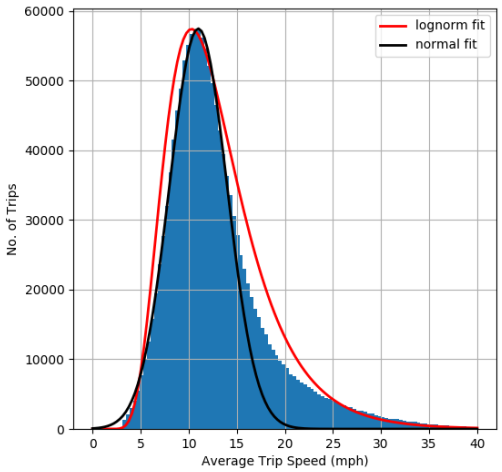

Figure 6.1: Distribution of the average trip speed, the red line is a lognormal fit and the black line represents an adjusted normal fit.

6.2 ANOVA on the average speed in different weeks

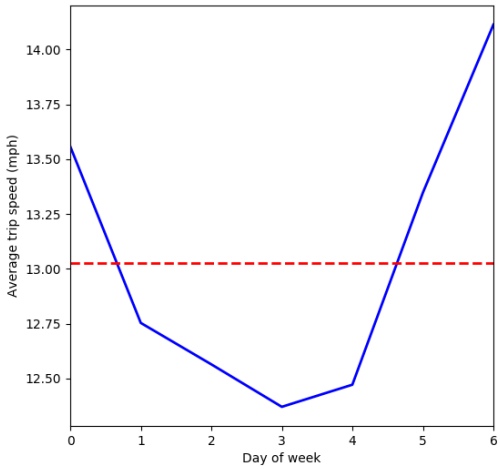

Figure 6.2: Average trip speed grouped by day of week. The dashed horizontal line indicates the mean trip speed of the whole month.

To test if the average trip speed is constant among different weeks of September 2015, I run an analysis of variance (ANOVA) on 1,000 randomly sampled trip speeds grouped by the

weeks. The null hypothesis is that: all the weeks have the same average speed. The result of the ANOVA analysis has a p-value of 9.2e-5. Therefore I reject the null hypothesis and regard the average trip speed data are significantly different among different weeks of September 2015. My hypotheses of this finding are as follows:

1. As shown in Figure 6.2, the average speed as a clear dependence on the day of the week. In general, there is less traffic on the weekends compared to weekdays. Hence the average trip speed is higher. On Fridays, people tend to leave office early, so the rush hour traffic is not as bad. Saturdays have exceptionally high trip speed. Compared to the full weeks (week 2, 3, 4), week 1 consists of Friday, Saturday and only three weekdays. Therefore week 1 should have the highest average trip speed. Week 5, on the other hand, only has one "fast day" (Sunday) and three weekdays. As a result, it should have a lower average trip speed compared to the full weeks.

2. Some other irregular events (e.g., holidays, sports events like the US open) might cause additional variance.

6.3 Average trip speed grouped by the hour of the day

Figure 6.3 illustrates the change in the average trip speed over a day. Similar to the analysis of the trip speed versus day of the week, the trip speed depends mainly on the traffic. From late night to early morning (10 pm - 6 am next day), there is least amount of traffic and correspondingly highest trip speed. From 6 am to 8 am there is a sharp drop in the trip speed, due to the start of the daily commute. The average trip speed stay around 12 mph from 9 am to 2 pm. During the "extended" rush hour (3 pm - 7 pm, because some people tend to leave office earlier/later to avoid the worst traffic), the average trip speed reaches a "bowl" shaped region with a minimum at 5 pm. After 8 pm, the traffic gradually clears out, and the trip speed rises steadily.

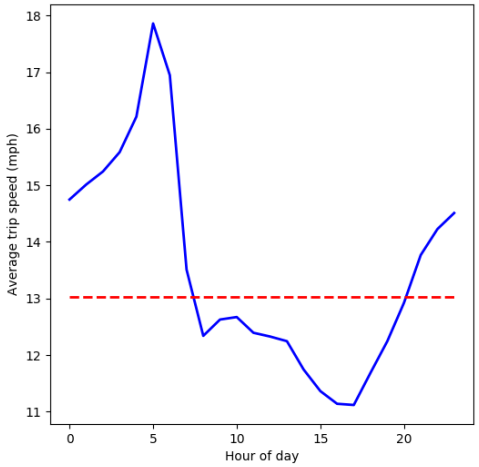

Figure 6.3: Average trip speed grouped by hour of day. The dashed horizontal line indicates the mean trip speed.

7 Predictive Model for the Trip Rate

In this section, I construct a predictive model for the tip percentage. Due to the limitation of my computing power (running on my laptop), I took a sample of 10,000 trips for the training purpose. I have selected three tree based regressors (Decision Tree, Random Forest, Gradient

Boosting Tree) to fit the "Tip_rate" variable. The reason for this choice is: 1) Tree-based models are powerful and accurate in nonlinear problems, 2) One of the major weaknesses of tree-based models, vulnerable to overfitting, can be controlled in this problem because of the large size of the dataset and our relatively aggressive strategy on removing the outliners in sections 4.

7.1 Feature selection and hyper-parameter tuning

The features I have selected in this model contains:

1. Categorical: "VendorID", "Hour", "RateCodeID", "Day_of_week", "Payment_type"
2. Numerical: "Tip_amount", "Trip_distance", "Time_span"

My general strategy is to include all the most relevant features, while to avoid keeping highly correlated features at the same time. For example, the "Extra" column ($ 0.5 for trips between 8 pm and 6 am, $ 1.0 for trips during rush hour) can be determined by the "Hour" and "Day_of_week" features. Therefore, it is excluded from the predictive feature list.
All the categorical features are encoded together by the "OneHotEncoder" function. The numerical features are used directly because feature scaling is unnecessary for tree-based models.
I have used the "GridSearchCV" to fine-tune the hyper-parameters of the different regressors. I only selected several important hyper-parameters for the grid search due to the limited computing power. Note that this step is commented out in the code I uploaded to save time. Table 7.1 lists the optimized hyper-parameters for each regressor. The Gradient Boosting Tree takes a smaller tree depth compared to Decision Tree or Random Forest regressors for best performance.

Table 7.1: Optimized hyper-parameters for each tree based regressor.

Regressor	Hyper-parameter	Optimized value
Decision Tree	[max_depth, min_samples_split]	[14, 3]
Random Forest	[max_depth, n_estimator]	[14, 60]
Gradient Boosting Tree	max_depth	6

7.2 Model performance

I performed a 10-fold cross validation for each regressor to measure the accuracy, the performance and the elapsed time for the three machines are listed in Table 7.2. All three machines perform quite well in predicting the tip rate. The Gradient Boosting Tree has exceptionally high accuracy and low variance, however, it is also the most time-consuming approach.

Table 7.2: Optimized hyper-parameters for each tree based regressor.

regressor	mean accuracy	std. of accuracy	elapsed time (second)
Decision Tree	0.9875	0.0055	<1
Random Forest	0.9919	0.0071	11
Gradient Boosting Tree	0.9936	0.0055	28

Figure 7.1 plots the histogram of the distribution of the error between the predicted tip rate and the actual value. The distribution has a symmetric structure, most of the tip rates in my sample can be predicted without any error by the Gradient Boosting Machine. The most extensive error is less than 3%. Overall, my trained model does a good job in predicting the "Tip_rate" variable.

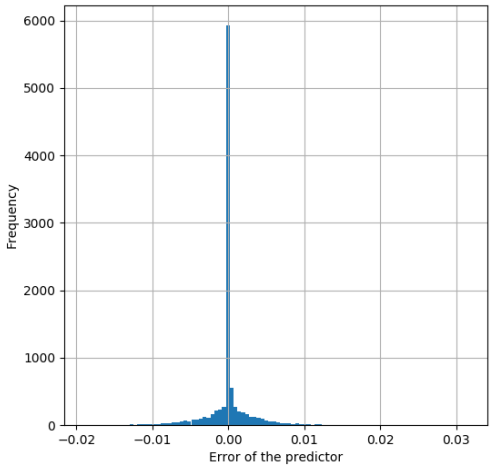

Figure 7.1: Error distribution of the predictions from the Gradient Boosting Machine.

8 Summary and Future Work

To summarize, I have used python 3.6 to analyze the NYC green taxi data in September 2015. Due to the complex correlation within the data, the distribution of the trip distance and many other variables follow a lognormal distribution. I have also identified several interesting structures hidden in various features of the dataset, and constructed qualitative hypotheses to explain these phenomena. As the last step, I have trained several tree-based regression models that perform quite well in predicting the tip rate of the trips.

The NYC green taxi data I analyzed in this report is an interesting dataset with lots of profound knowledge to be discovered. Besides the work I have done in this report, there are many other ideas that I want to try as listed below. However, those directions are out of the scope of this report.

1. The ride-sharing idea (Option C of the challenge) is compelling. Although I did not choose it in this project, I have a general plan for solving the problem and make it a real-time application. Since we cannot predict the future, one can only share a ride with someone who ordered a taxi nearby a short while ago. Given the longitude/latitude and the time for a person who wants to share a ride, first I select all the rides happened in the past short amount of time t (e.g., 2 minutes) from the "sea" of transactions. Since all the transactions are recorded as a time series data, this step should be fairly easy to perform. Suppose there are $n(t)$ rides within this window, we can 1) sort all of them and select the first k rides, time complexity $\sim O(N \log N)$ or 2) using a heap of size k, with time complexity $\sim O(N \log k)$, to find the $k-$nearest rides. **Please do not grade my work based on this paragraph, I have selected the speed distribution challenge (Option A) as my answer to Question 5.**

2. There are some other features, such as the origin/destination borough of the rides, could generate interesting information, or have a positive impact on the predictive models.

3. Visualize the decision trees (with for example "Graphviz") to have a better understanding about how the features affect the prediction.

References

[1] Allanson, P. Farm size structure in england and wales 1939-89. *Journal of Agricultural Economics 43*, 2, 137–148.

[2] Limpert, E., Stahel, W. A., and Abbt, M. Log-normal distributions across the sciences: Keys and clueson the charms of statistics, and how mechanical models resembling gambling machines offer a link to a handy way to characterize log-normal distributions, which can provide deeper insight into variability and probability normal or log-normal: That is the question. *BioScience 51*, 5 (2001), 341–352.