New York taxi fare prediction with Pyspark
Informatique project 2020

VU Thi Van Anh

ENSAE ParisTech
École nationale de la statistique et de l'administration économique
université PARIS-SACLAY

Abstract

In this project, we will deal with the Big Data problem in predicting taxi fare in New York proposed in an old Kaggle competition in 2018, called *"New York city taxi fare prediction"*. This report will be developed in three axes: the introduction of the problems, the presentation of our implementation in Pyspark, then finally the interpretation of our results.

1 Introduction

The taxicabs of New York City are widely recognized icons of vibrant life in the city. With the explode of ridesharing tendency like Uber or Grab, it is increasingly important for taxi companies to provide visibility to their estimated fare and ride duration, since the competing apps provide these metrics upfront. Predicting fare and duration of a ride can help passengers decide when is the optimal time to start their commute, or help drivers decide which of two potential rides will be more profitable, for example.

The objective of this project is to predict the New York taxi fare, which based on an old Kaggle competition in 2018, namely "New York City Taxi Fare Prediction". Instead of using Python, Spark is employed in this project as it's powerful engine choice delicated to solve Big Data problems.

In order to predict fare, only data which would be available at the beginning of a ride was used. This includes pickup and dropoff coordinates, trip distance, start time, number of passengers and rush hours. Linear regression with model selection, decision tree, random forest and Gradient Boosted tree models, available from algorithms library MLlib, were used to predict fare amount.

2 Data analysis

2.1 Data set

The data used in this study is the set of New York City Taxi's trip data, which contains observations on around 55.5 million taxi rides in New York City between 2009 and 2015. To explore the dataset and then, build the models, a random subset of 2 percent the original dataset were used, containing more than 1.1 million observations.

The original dataset contains features as pickup and dropoff locations, as longitude and latitude coordinates, time and date of pickup and dropoff, ride fare, tip amount, payment type, trip distance and passenger count. The data was processed to extract separate features for year, month, day, weekday, and hour from the date and time of each ride, as well as trip distance as the difference between dropoff and pickup locations.

2.2 Data cleaning

In this section, we study univariates analysis and remove outlier/illegitimate values which may be caused due to some error. In general, we consider all the taxi cars under 8-seats and the number of observations where it's passengers' number is over 7 is negligeable (4 observations), so we decide to drop all these observations. All observations with unreasonable negative fare amount are also dropped.

It is inferred that New York is bounded by the location coordinates (latitude, longitude) of (-90, 90) and (-180, 180), so hence any coordinates not within these coordinates are not considered as we are only concerned with pickups and dropoffs which originate within New York.

Coordinates transform to distance

One additional approach taken to further model the effect of the pickup and dropoff locations was to transform the coordinates. Most of the streets and avenues in New York are aligned in a grid structure With the hypothesis that the avenue or street could explain some of the effect of the location, transforming the coordinates into trip distance.

- Earth radius in kilometers $R = 6371$
- ϕ_1 - radian of pickup latitude
- ϕ_2 - radian of dropoff latitude
- $\Delta(\phi) = \phi_2 - \phi_1$
- λ_1 - radian of pickup longitude
- λ_2 - radian of pickup longitude
- $\Delta(\lambda) = \lambda_2 - \lambda_1$

$$a = sin^2(\frac{\Delta(\phi)}{2}) + cos(\phi_1)cos(\phi_2)sin^2(\frac{\Delta(\lambda)}{2})$$

$$distance = 2R.atan2(\sqrt{a}, \sqrt{1-a})$$

2.3 Data visualization

fare_amount - The figure 1 shows the boxplot and distribution of fare amount. It's obviously, the skewed box plot shows us the presence of outliers where value of outlier is very high, so we focus on studying the distribution of fare amount under 100 USD (accounting for more than 99% subset) and the taxi fare focuses mostly on interval of 0-66 dollars.

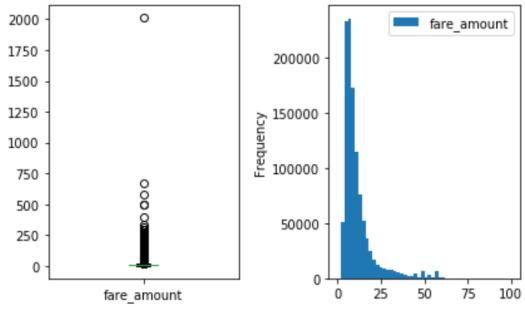

Figure 1: Distribution of fare amount

distance - It's argued that one of the most important features affecting the taxi fare is the trip distance. The distance between the pickup and dropoff locations is assumed to exhibit a linear correlation with fare amount and in order to study further on this variable, we divide all observations in sub-dataset into 4 groups depending on its distance including:

- distance_bins = 0 for $distance \in [0, 0.25]$
- distance_bins = 1 for $distance \in [0.25, 10]$
- distance_bins = 2 for $distance \in [10, 50]$
- distance_bins = 4 for $distance > 50$

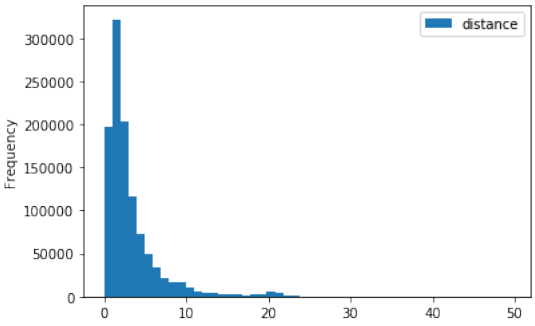

Figure 2: Distribution of distance lower than 50km

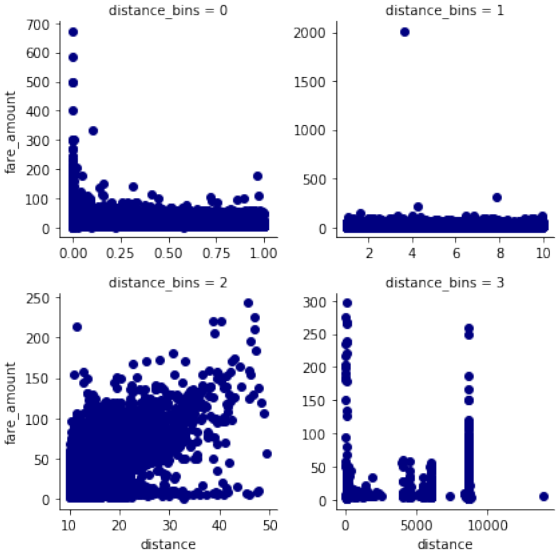

Figure 3: Scatterplot of fare and distance by groups

The figure 3 shows the relationship between taxi fare and trip distance by distance groups. In the first group where trip distance ranging from 0 to 0.25 km, the fare focuses mainly on range under 100 USD while there are several abnormal outliers close to 0km but fare is much high (over

200 USD). Even though these high fare can be explained by waiting charges but it seems to be unreasonable while waiting fare is only 2.5 - 3 dollars per hour comparing with total fare bill of over 100 USD for only 1km trip.

The same pattern is also seen on the next second group where distance is under 10km, and the total is more stable in the budget under 100 USD. This can be also explained by its volume while this group accounts for largest proportion of data set (80%).

In the third group where distance ranging from 10 to 50km, it's obvious that there is linearly increasing relationship between trip fare and distance. On the other hand, in the last group, it seems unexplainable for abnormal observations with distance over than 50km and much high fare (over 200 USD). Therefore, these abnormal outliers will be removed or modified by fare calculating formula in the section of data engineering.

year - By studying all trip with fare amount inferior to 100 USD over years, it's can be seen that the averaged fare amount increases by years.

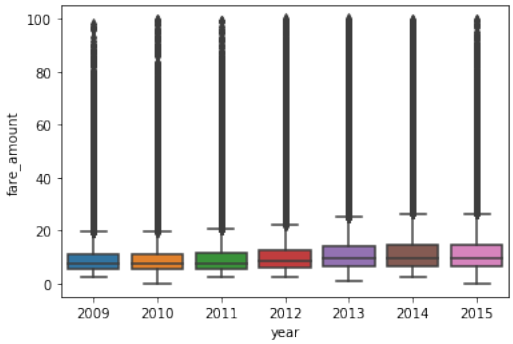

Figure 4: Boxplot of fare by years

Other features such as hours, weekdays and month - In studying taxi fare depending on its time span features such as hours, weekend and peak-hour intervals, it seems there no much differences on fare amount among different groups of time. The following figures show the example of taxi fare by weekdays-weekends and by rush hours.

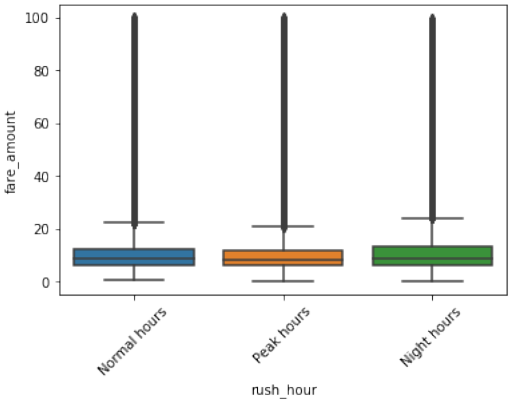

Figure 5: Boxplot of fare by groups of peak hours

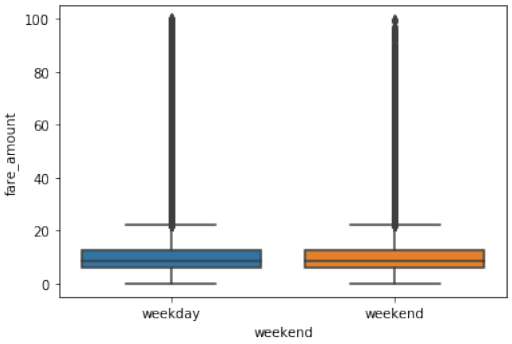

Figure 6: Boxplot of fare by groups of weekdays

3 Methodology

3.1 Feature engineering

By referring the taxi fare information[1], there is not much different between taxi fare on weekdays and weekend (including weekdays nights) in New York except for the base fare. Based on these information, several steps in features engineering are taken place, including:

- Removing observations where distance = 0 and fare amount lower than 2.5
- Modifying distance for all observations where its distance ranging from 0 to 0.25 km and the fare is much larger than expected (we can understand that there are several cases taxi drivers count the waiting time); or where distance is over 300km. The distance is recalculated by following formula:
  - distance = (fare_amount - 2.5) / 1.56 if on weekday
  - distance = (fare_amount - 3) / 1.56 if on weekend or night time

[1] https://www.taxi-calculator.com/taxi-rate-new-york-city/259

- Modifying fare amount for all observations where fare amount is 0 or over 2000 USD by using the modified distance in the above formula.

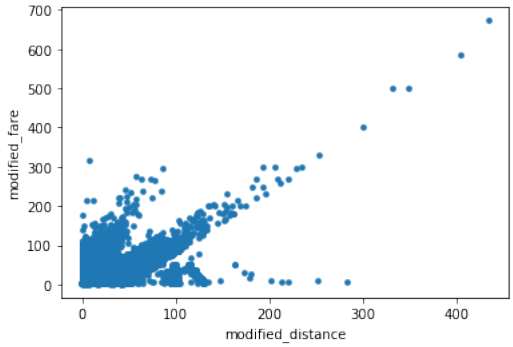

Figure 7: Scatterplot of fare and distance

3.2 Implementation models

Linear regression model

As a baseline prediction, the mean fare from the training set were used to predict a constant value for the validation set. From feature selection, the linear regression with all covariates available at the pickup and dropoff locations was predicted to be the best fare prediction. The linear regression model finds the set of θ coefficients that minimize the sum of squared errors

$$y^{(i)} = \theta_0 + \Sigma_j \theta_j x_j^{(i)}$$

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

Selection tree and Random Forest

As traffic is clustered and aggregated more densely to different locations at different times, the location of the ride will clearly have an affect on the trip duration and trip fare. Although there is no straightforward way of considering all locations between the start and end points of a ride, the pickup and dropoff locations are available in the dataset and can be used to model some of the effect of traffic and conjunctions.

In the linear regressions, the locations' effect on trip duration is modeled simply by the magnitude of the longitude and latitude coordinates. As traffic is clearly not varying solely based on the magnitude of the coordinates, the linear models fail to account for the nonlinear effect the locations have on traffic and hence trip duration (and also fare amount). An algorithm that can better account for these nonlinearities is the random forest.

Comparing to algorithm of selection tree, random forest algorithm aggregates many decision trees built on bootstrapped samples of the training data in order to reduce the high variance of a single decision tree and improve prediction accuracy. Each of these decision trees aims to divide the predictor space - the possible values for all features in a number of distinct and non-overlapping regions; and the predictor space is divided into high-dimensional rectangles of regions that minimise the RSS.

When building each tree, a top-down approach is taken. Beginning with all points in the same region, the algorithm successively splits the predictor space into two halves, stopping when there are no more than five points in a region. Once the regions are defined, a prediction by a single tree is made by averaging the responses of the training observations in the region to which the test observation belongs.

On the other hand, in the random forest, a large number of trees are fit, each using a bootstrap sample from the training data, and a prediction of a new observation is made using the mean of the predictions by all the trees. At each split, only m of the total n predictors are randomly chosen to be considered. This approach is taken to de-correlate the trees, as considering all predictors might yield very similar trees when one or a few predictors are particularly strong. As averaging many uncorrelated trees leads to a larger reduction in variance, this approach often yields better prediction results.

Gradient Boosted tree regression

Like random forests, gradient boosting is a set of decision trees. Random forest build trees in par-

allel - fully grown decision trees (low bias, high variance), while in boosting (high bias, low variance), trees are built sequentially i.e. each tree is grown using information from previously grown trees unlike in bagging where we create multiple copies of original training data and fit separate decision tree on each. That's why it generally performs better than random forest. Though both random forests and boosting trees are prone to overfitting, boosting models are more prone.

3.3 Evaluation metrics

In order to evaluate the results among applied models, we use the metrics of RMSE.

4 Result and conclusion

Before we start predictions using the tree-based regression models, the cleaned subset of New York taxi trip is splitted into train and valid sets with the proportion 0.95 : 0.05, with the number of observations of 1,051,423 : 55.092, respectively.

Models	Train	Val
Linear regression	8.094	8.004
Generalize linear regression	8.111	8.092
Selection tree	7.226	7.039
Random forest	5.696	5.389
GB tree regression	4.732	4.553

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

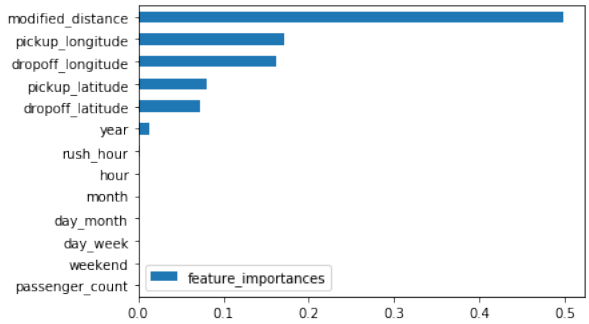

Figure 8: Important features in Random Forest

The random forest model result is much better than the linear model and selection tree however, Gradient boosted tree regression outperform all other models used, as it manages to model the non-linearities of traffic and location effect. Although the model accounts for the effect of pickup and dropoff locations, it has no way of modeling the effects of the locations along the route.

Although using several time span features improves the models, rush hour or weekend does not yield any significant improvement in prediction accuracy.

In conclusion, in this project, Pyspark has been employed to solve the problem of regression in a large dataset. Obviously, Spark is the most comprehensive suite of execution engines specifically tailored to data engineering, however, compared with scikit-learn in Python, in Machine learning Spark MLlib, there are fewer algorithms present. It lags behind in terms of a number of available algorithms. Although Spark has many limitations, it is still trending in the big data world.

References

[1] Vanajakshi, L., S. C. Subramanian, and R. Sivanandan. *Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses*. 2009.

[2] New York Taxi Fare Prediction - Can you predict a rider's taxi fare? https://www.kaggle.com/c/new-york-city-taxi-fare-prediction. 2018.