Recruit Restaurant Visitor Forecasting

Cynthia Dong, Zhiqi Fu, Zhangyi Pan

December 2019

Abstract

Restaurants holders care about how to arrange raw materials and service people in advance. Thus, they want to know approximately how many people will come to their stores for a particular day. This project is aimed at predicting number of visitors for each store and for each day given the dataset provided by Kaggle competition. We proposed three different models based on regression tree and did a experiment to see which model performs better in which circumstances. What's more, we also compared our model with existing machine learning models like Logistic Regression, SVM, XGBoost.

1 Introduction

From childhood till now, the three of us are interested in various kinds of food. We always hold a dream to open our own restaurant, making profits as well as enjoying ourselves in our interested fields. However, owning and operating a restaurant is a complicated matter. For a restaurant to succeed, it's important to estimate the number of visitors beforehand in order to buy ingredients and prepare in advance to minimize wastes of resources and labors. Our dataset comes from the Kaggle competition "Recruit Restaurant Visitor Forecasting" (see Kaggle Competition Description) which contains the data from 2016 until April 2017. By extracting data from Hot Pepper Gourmet (a restaurant review service), AirREGI (a restaurant point of sales service), and Restaurant Board (reservation log management software), we have access to data like daily reservations, holiday information, actual number of customers, and so on. With these data, we are able to solve this time-series forecasting problem.

The main challenge of this project is that there are a lot of unpredictable factors affect restaurant attendance. It's very hard to predict precisely with a limited set of attributes. Therefore, we will build some new features based on the original dataset and also gain additional infomration from other resources. The other challenge is from the nature of real-world machine learning tasks. Evaluation Metrics like Mean Squared Error(MSE) and Root Mean Squared Logarithm Error(RMSLE) don't have much real-world meaning if we only look at that numerical number in this specific case. In the evaluation part, we want to inspect the raw data more in detail to get a better sense about which model

performs better under what circumstance.

The main goals of this project are:

- Experiment with variations of Regression Tree to get relatively low RMSLE compared to other methods like Logistic Regression, XGBoost, SVM, Random Forest.
- Build a customized prediction model that is suitable for this particular industry, i.e Try not to overestimate much, try to perform well on some extreme cases (number of visitors varies a lot from day to day)

In this report, we will present our data preprocessing work and three models which are based on regression tree, but with customized loss function.

The rest of the report is organized as follows, Section 2 lists related works. In Section 3, we elaborate data preprocessing work that we have done and our proposed models. Our experimental results are presented in section 4. Then we will discuss about the results and possible future work in section 5.

2 Related Work

Since the dataset was from a Kaggle competition, we reviewed the solutions from leader board. There are several things that competitors usually do:

- Since the datasets provide several tables, we need to find a way to select useful features and build new features out of provided raw features. Almost every user do some kind of feature buildings and here is a good example: see Kaggle Notebook or go to the next url: `https://www.kaggle.com/headsortails/be-my-guest-recruit-restaurant-eda`. We will elaborate more details about feature building in section 3.
- Most of the leading competitors use some variant of boosting. See a open source from one of the leading competitor and another open source from one of the leading competitor or go to the url: `https://github.com/anki1909/Recruit-Restaurant-Visitor-Forecasting`, `https://github.com/HaotianMXu/Predict-number-of-visitors-to-restaurants`. Even though boosting is a general approach to lower bias, we decided to experiment with regression tree to see if it can produce relatively high accuracy, and with better interpretability.

We choose regression tree because of these reasons:

- We believe that interpretability is important in business world. Compared to other methods, regression tree is very intuitive and easy to explain to business holders.[1] [2]. In our case, restaurant holders can tell from the regression tree that which attributes can influence the number of visitors. With these information, our results will help them improving their business practices.
- Our features are a combination of categorical and numerical data and our target is continuous value, so we choose regression tree here.[3]

3 Our Approach

3.1 Data Preprocessing

3.1.1 Attributes from Original Dataset

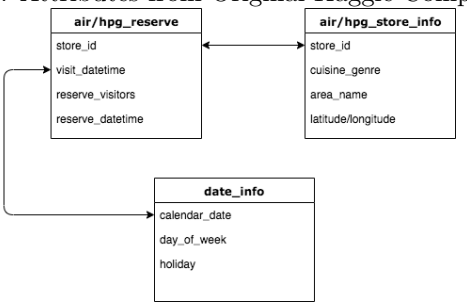
The data were collected from two sites: Hot Pepper Gourmet (hpg) and AirREGI / Restaurant Board (air). Although there are lots of files, we could joint them together using the restaurant id. Figure 1 shows the attributes that we could directly get from original dataset. Note that day_of_week and $holiday$ work as dummy variables: a variable that only takes 2 values.

Number of Last 10_day visitors: In order to show if a store is popular in the recent, we included number of last 10_day visitors as a attribute for prediction. If there were only 6 days among the last 10 days that the restaurant were open, we only took average of those 6 days.

Number of Reservations: Since we are dealing with time series, we are particularly careful about possible information leakage. To calculate the number of reserved visitors, we only count the reserved ones in the previous days. Since restaurants' holders usually decide how many raw materials they should buy in the beginning of a day, so we don't want to include the reservations happen in the same day

Trend: We also want to include the trend of popularity of each store over the past one year. For each store, we took all the data points (number of visitors for each day) and fitted a linear regression line over these data points. The slope of the line was what we used to indicate the trend of popularity.

Figure 1: Attributes from Original Kaggle Competition



3.1.2 Attributes from Other Sources

Other than these variables, we also consider weather as an important factor. It is obvious that people tend to hang out and have dinner on sunny days and choose to stay at home when there are heavy rains. What's more, on cold days, people tend to eat some hot food, while on hot days, people prefer to eat some cooler dishes. In order to get the weather data for a particular day and for a particular store, we found the closest weather station of that store

based on longitude and latitude attributes, and then found the weather data recorded by that station. Fortunately, an expert had done some preliminary steps for us. He provided us with 1663 weather station's data recorded between 01-01-2016 and 05-31-2017 which covered the same date window as our dataset did. The recorded data contains average temperature, high temperature, low temperature, precipitation, hours_ sunlight, and some other features that are not related to our project. He also provided us with the closest weather station for each store based on geographical coordinates. However, not all of the closest stations contain complete information. For example, some of them are missing high temperature. In these cases, we filtered out those stations with incomplete information, and then found the second closest one which contains all the information. We also discarded precipitation and hours_sunlight since a great amount of stations are missing these two values.

Abnormal Temperature: In consideration of that low temperature or high temperature may be not significant features because people will still hang out during December when the temperature is extremely low each day. We built the abnormal temperature feature: Average Temperature - Last 5-day Average Temperature. For example if the temperature drops drastically for one day, people tend to stay at home instead of going out. However, if it's winter and the temperature remains almost the same, people will also hang out for lunch or dinner.

As a result, the final list of attributes that we have is: number of reserved visitors, cuisine genre, day_of_week, holiday, number of Last 10_day visitors, average temperature, abnormal temperature and trend. And we got 37062 observations to train the model.

3.2 Model Building

Our proposed models are based on regression tree. For the regression part, we used Ridge Regression implemented by sklearn with $\alpha = 1$. The baseline model is DaRDecisionTree (Divide and Regression), which we implemented for HW3. For each node, we iterated through each possible combination of splitting attribute and splitting value. We picked the pair which produced minimum of sum variance for left node and right node.

As we discussed before, we want to build a model that is fit for restaurant visitor prediction. In order to achieve this goal, we changed the way to determine the splitting pairs. To be specific, for each possible pair of splitting variable and value, we ran a ridge regression on left node and right node, and picked the splitting pair with minimum sum of error. This error was produced using different evaluation metric in order to push our predicted number of visitors to something appropriate for restaurant holders. Three different evaluation metrics that we tried are listed below:

RMSLE:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2} \tag{1}$$

It does have an advantage over other evaluation metrics like MSE, since it can penalize underestimates more than overestimates by its nature. For example, if we have actually 100 people coming in one day, predicted value 90 will get an error 0.002 and predicted value 110 will get an error 0.0017. This property makes sense with regard to our goal. If restaurants have significantly more guests than they expected, they might not have enough raw materials or waiters/waitresses.

Weighted RMSLE:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n} w_i(\log(p_i+1)-\log(a_i+1))^2} \tag{2}$$

where

$$w_i = log((p_i - min(a))^2)$$

RMSLE only cares about the relative difference between the true and the predicted value. In other words, it will treat small differences between small true and predicted values approximately the same as big differences between large true and predicted values. In our case, the situation that we have actually 10 people coming in one day and we predict 20 people coming in compared with the situation that we have actually 10000 people coming in one day and we predict 20000 people coming in one day. These two situations will generate almost the same error. However, We don't think it's a fair evaluation metric which gives the same weight to those two situations. Thus we introduce weighted RMSLE, which also takes the scale of number of visitors into account.

Weighted RMSLE with Penalty Term:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n} w_i(log(p_i+1)-log(a_i+1))^2 + \alpha \times max\{\frac{p_i - a_i}{a_i}, 0\}} \tag{3}$$

where

$$w_i = log((p_i - min(a))^2)$$

α is based on cuisine genre

The other thing that we care about is the wastes. For restaurants, we want to avoid the cases that we overestimate too much. If we overestimate too much, based on our prediction, restaurants may prepare more than sufficient raw materials, which cause wastes. Also, different cuisine types will have different tolerance rate to wastes. For some of the restaurants, they cannot reuse the remaining materials, we will penalize more for overestimating, while for restaurants that can reuse remaining materials, we will penalize less for overestimating.

In implementation of all these three models, we did group by before running our models. First we grouped the whole data set based on different cuisine types based on empirical experience. The resulting groups are listed below:

- Genre 1: Italian/French/Western
- Genre 2: Japanese Food
- Genre 3: Okonomiyaki/Yakiniku/Korean food
- Genre 4: Asian/International
- Genre 5: Creative cuision
- Genre 6: Bar/Izakaya/Cocktail
- Genre 7: Karaoke/Party/Cafe/Other

We also grouped the data set by store, since each store will have its own pair of splitting variable and value. In other words, we assume that number of visitors in different stores is influenced by different factors.

4 Experimental Results

First, in order to compare the three models we proposed and the baseline model, we use genre 2 (Japanese Food) as a example for analysis. We listed RMSLE for each model in Table 1. Notice that v1 is our baseline model, v2 is RMSLE, v3 is Weighted RMSLE, v4 is Weighted RMSLE with Penalty Term.

Model	v1	v2	v3	v4
Genre 2 RMSLE	0.3494	0.3146	0.3044	0.3217

Table 1: Comparison between four models with total test size 2800

We can see that Weighted RMSLE produced minimum error, and all three proposed models (v2, v3, v4) outperform our baseline model (v1). We also inspected predicted number of visitors and actual number of visitors in details to get a better idea of our models' performance.

Figure 2: Model v3 Compared with Model v2

store_id	true_y	pred_y_v2	pred_y_v3
air_84f6876ff7e83ae7	[15]	[19.05581629]	[12.17902219]
air_84f6876ff7e83ae7	[22]	[16.76885016]	[17.52227177]
air_84f6876ff7e83ae7	[8]	[16.74206914]	[12.93904427]
air_84f6876ff7e83ae7	[14]	[7.2910506]	[10.2946804]
air_84f6876ff7e83ae7	[14]	[16.96449104]	[14.26968369]
air_84f6876ff7e83ae7	[6]	[11.50298472]	[10.22796426]
air_84f6876ff7e83ae7	[34]	[28.05447365]	[31.18346678]
air_84f6876ff7e83ae7	[23]	[24.11734677]	[26.47791393]
air_84f6876ff7e83ae7	[11]	[15.09717123]	[12.92889182]
air_84f6876ff7e83ae7	[7]	[16.70244579]	[15.22924728]

This snapshot(Figure 2) is extracted from a store whose number of visitors vary from day to day a lot. For example, in some days, it had 20 or 30 customers, while on the other days, it only had less than 10 customers. In this kind of cases, Weighted RMSLE generally performs better than RMSLE. That weight term is relatively more influential for this kind store, and this term pushes our predicted y values towards the actual y values.

From Table 1, we also observed that v4's RMSLE is larger compared with v2 and v3. Thus, we looked into the comparison of v4 with v3 to see what's going on there.

Figure 3: Model v4 Compared with Model v3

store_id	true_y	pred_y_v3	pred_y_v4
air_84f6876ff7e83ae7	[35]	[19.569472]	[18.73380688]
air_84f6876ff7e83ae7	[33]	[20.00220447]	[19.03912852]
air_84f6876ff7e83ae7	[21]	[20.179903]	[19.33748558]
air_84f6876ff7e83ae7	[21]	[19.35229334]	[19.38536832]
air_84f6876ff7e83ae7	[33]	[18.90727918]	[18.92315505]
air_84f6876ff7e83ae7	[15]	[19.16019766]	[18.68843998]
air_84f6876ff7e83ae7	[23]	[19.15162586]	[18.45500431]
air_84f6876ff7e83ae7	[33]	[19.25157805]	[18.59021911]
air_84f6876ff7e83ae7	[40]	[19.41348389]	[18.79800978]

Figure 3 is a snapshot of comparison between v3 and v4. We observed that although there are cases where version 4 tries to underestimate, the performance is not stable. What's more, on cases where version 3 already underestimates, version 4 tries to push the predict y far off from the actual y. Thus, for now, the best model we can get is Regression Tree with Weighted RMSLE as a loss function. Table 2 presents RMSLE for all genres with a comparison between v1 and v3.

Genre	Genre 1	Genre 2	Genre 3	Genre 4	Genre 5	Genre 6	Genre 7
v1 RMSLE	0.4051	0.3494	0.3619	1.2596	0.4839	0.4647	0.3987
v3 RMSLE	0.4028	0.3044	0.3791	1.1006	0.3982	0.4472	0.3877

Table 2: Comparison between v1 and v3 for all genres

We can see that v3 performs better than v1 in almost all genres. Now we want to compare our model with other prevailing machine learning techniques like Logisitic Regression, XGBoost, SVM, Random Forest.

From Table 3, we observed that Logistic Regression performed worst and it's not surprising, because Logistic Regression is not appropriate to predict continuous y values. Our model also performed better than XGBoost. One

Method	Our Model	SVM	Logisitic Regression	XGBoost
Total RMSLE	0.4079	0.3825	0.6219	0.4579

Table 3: Comparison between different methods

possible reason could be XGBoost is overfitting to the training set. Since for predicting restaurants' visitors, each day may vary a lot from each other and we have a special holiday in our test set. Thus, it's possible for XGBoost to perform bad in our test set. Our model was worse than SVM in terms of RMSLE. However, SVM is lack of interpretability. In our case, it's important that we can explain the model to restaurant holders and they can clearly see which factors are influencing number of visitors mosts.

5 Discussions

Our work enables the restaurant holders to know the features that have significant impact on the number of visitors and thus pay more attention to those features. Since we incorporate the group by idea, holders could find the group that best fits their restaurants and predict within the group, thus achieving a relatively more precise predictions. However, there are some potential future works remaining to be done. We think in the future it might be possible for us to create a collaboration system in which the restaurants could share their information and thereby allow some restaurant with limited data to find some data from other similar ones to make better prediction. We think it will also be helpful for us to include some features about local competitions of similar restaurants.

References

[1] Liang-Tsung Huang, M. Michael Gromiha, and Shinn-Ying Ho. iPTREE-STAB: interpretable decision tree based method for predicting protein stability changes upon mutations. *Bioinformatics*, 23(10):1292–1293, 03 2007.

[2] Polly Phipps, Daniell Toth, et al. Analyzing establishment nonresponse using an interpretable regression tree model with linked administrative data. *The Annals of Applied Statistics*, 6(2):772–794, 2012.

[3] Aram Karalic. Linear regression in regression tree leaves. In *In Proceedings of ECAI-92*. Citeseer, 1992.