Recruit Restaurant Visitor Forecasting

Cynthia Dong(wdong7), Zhiqi Fu(zfu32), Zhangyi Pan(zpan27)

I. 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 and buy ingredients and prepare in advance to minimize wastes of resources and labors. In this project, we care about the prediction of number of visitors for some certain days, with some observations from Hot Pepper Gourmet (hpg)(similar to Yelp, here users can search restaurants and also make a reservation online) and AirREGI / Restaurant Board (air)(similar to Square, a reservation control and cash register system). With features given in those data sources, we hope we could have an accurate prediction.

II. Literature Review

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 building 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.
- We don't think the evaluation metric given in the competition makes a lot sense, so we want to modify this evaluation metric a little bit, and improve accuracy based on the new evaluation metric. Details about evaluation metric will be discussed in Section 4 part A.
- 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 another approach other than boosting to see if it can produce relatively high accuracy. Details about proposed model will be discussed in Section 4 part B.

III. Feature Building

A. Original Data set

1) Data Sources: The data are 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.

2) Attributes in hpg/air:

- Restaurant id
- the time of reservation
- the time the reservation was made
- the number of visitors for that reservation
- genre
- area
- latitude/longitude
- visitor date
- number of visitors

3) Dummy variables: Dummy variables work as indicator variables, which help us to include categorical variable taking only two values in our analysis. The current two dummy variables we considered to include are Japanese Holiday and Day of the week.

- Japanese Holiday:
  We include a dummy variable for the Japanese holiday. We do this because the restaurant reservation and visiting during holidays may be atypical. For example, people may be more likely to go to a restaurant like Valentines day and less likely to go to a restaurant during New Year.
- Day of the week:
  We include dummy variables for each day of the week to control for average differences in guest visiting that might be day-of-the-week specific.

B. Attributes based on existing data

1) Time series: Time series analysis is prominently used in analyzing behaviors and trends in corporate performance and sales data because it allows dealing with issues such as autocorrelation, trends, and seasonal variation.In our case, it is useful for forecasting guest visiting, taking into account the strong market development patterns of the restaurants.

2) Categories: The whole dataset could be divided into different categories in which each category may have its own characteristics. Therefore, we expect variables to respond differently to guest visiting. For example, different cuisine may have different amounts of guest visiting. Our analysis focused on, but not limited to cuisine type, geology

and etc.

3) Data Transformation: Instead of using the variable itself, we may make some transformation. For example, we may use the log value of the variable as our attributes, which help us to get a normal distribution and also eliminate the effect of outliers.

C. Introduce new attributes

1) Weather Data: After introducing new variables based on the original dataset, we also brainstormed other factors that will influence the number of guests visiting on that day. One of the factor is the weather. 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.

IV. Evaluation Strategy

A. Evaluation Metric

In the original Kaggle Competition, submissions are evaluated on the root mean squared logarithmic error:

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

where n is the total number of observations, p_i is the prediction of visitors, a_i is the actual number of visitors, and $\log(x)$ is the natural logarithm of x. We will refer to this evaluation metric as RMSLE in the following discussion. RMSLE is actually taking the logarithm of $\frac{p_i+1}{a_i+1}$ and get mean of squared of these logarithm percentage. That means the evaluation function only cares about the relative difference between the true and the predicted value. In other words, RMSLE 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 same error. We don't think it's a fair evaluation metric which gives the same weight to those two situations. Thus, we want to modify the evaluation metric to make it more fair.

However, RMSLE does have an advantage over other evaluation metrics like MSE, since it can penalize underestimates more than overestimates. 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. However, if they have less guests than they expected, the worst thing is just to waste the food they prepared. Thus, we want to preserve this property of RMSLE in our new evaluation metric.

Our plan for now is to do a linear combination of MSE and RMSLE to take advantage from both metrics. We will experiment with different possible scalars for this linear combination to see if there is a good evaluation metric that preserve then ranking that we want. For example, for actual value 20, predicted value 30 should do better than 10 in a good evaluation metric. Besides, predicted value 30 for actual value 20 should do better than predicted value 300 for actual value 200 in a good evaluation metric. Also we want to add a penalty term for underestimates, so that for actual value 20, predicted value 10 will get a greater error than predicted value 30.

If we cannot find a good evaluation metric through this linear combination method, we will use another strategy. Since we want to avoid overwhelming loss caused by the large number of visitors but we do want the difference in size to be effected in the metric, we want to assign a weight for each term in RMSLE. The formula will generally be:

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

and we define the weight w to be the normalized logarithm of the number of visitors, which will be:

$$w_i=\frac{log(y_i)-log(min(y)}{log(max(y))-log(min(y))}$$

In this formula, we significantly reduced the effects of large target while still take them into considerations.

B. Proposed Model

There might be some "clusters" of features in the datasets, which means some subsets of features are highly correlated within the subset while almost independent from the others.

Our first idea to deal with such problem is to apply "group by" filter to the datasets. We might want to group the data by some specific features. We also propose the usage of decision tree to first cluster the datasets into some groups. To deicde the level of the tree, we will try and compare trees with different levels.

After we separate the datasets into some sub-groups, we will use regression models to predict the number of visitors. In a way, we will use regression tree model to predict the number of visitors.

If the proposed method results in relatively low accuracy, we will try boosting on those sub-groups in order to improve performance.

We will compare the performance of all of these methods after all.

.