Predicting Restaurants' Rating and Popularity based on Yelp Dataset

Yiwen Guo[1], Anran Lu[1], Zeyu Wang[2]

[1] ICME, Stanford University, [2] Department of Economics, Stanford University

Abstract

Restaurants' rating on Yelp becomes an important indicator of their future. In this project, we focus on predicting ratings and popularity change of restaurants. With data from Yelp, we use several machine learning methods including logistic regression, Naive Bayes, Neural Network, and Support Vector Machine (SVM) to make relevant predictions. While logistic regression seems to perform better than the others, predictions from all the methods are far from perfect. This implies the potential improvement of more data and more suited methodology.

Data and Features

The data comes from Yelp Dataset Challenge [1]. It is a small subset of Yelp data, including information about local businesses in 12 metropolitan areas across 4 countries. The data contains information such as location, opening hours, price level, food type, service provided etc. It also includes review data, including text, time and rating.

From the raw dataset, we select 74 related features. Due to different cultures across cities, we only focus on restaurants in Toronto and surrounded areas in this project.

Term Definition

We predict both rating and popularity change. Rating is straightforward from the raw data. Popularity change is approximated in the following way.

Let len_i be the "age" of restaurant i. Let $rev_{j,i}$ be number of reviews received in year j for restaurant i.

$$trend_i = \frac{\sum_{j=1}^{j\leq(len_i+1)/2} \frac{rev_{j,i}}{\text{total \# of reviews in year j}}}{\sum_{j\geq(len_i+1)/2}^{len_i} \frac{rev_{j,i}}{\text{total \# of reviews in year j}}}$$

If it is bigger than 1, we say there is a downward trending. Otherwise, there is an upward trending.

We also want to know how broadly our model can apply. Thus besides training and testing within the city Toronto, we also apply our model to Toronto surrounded areas for test errors. Let lat_i, $long_i$ be the latitude and longitude of restaurant i.

Term Definition (Continued)

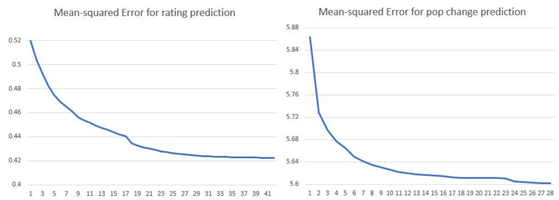
$$lat_{\text{Tor}} = \frac{\sum_{i \text{ in Toronto}} lat_i}{\sum_{i \text{ in Toronto}} 1} \quad long_{\text{Tor}} = \frac{\sum_{i \text{ in Toronto}} long_i}{\sum_{i \text{ in Toronto}} 1}$$

$$dist_i = (lat_i - lat_{\text{Tor}})^2 + (long_i - long_{\text{Tor}})^2$$

We consider the restaurant i surrounded if $dist_i <= 0.2$ and i not in Toronto.

Feature Selection

Since we extract variables as many as possible from the raw dataset, we are not sure if we run into the problem of overfitting, so we want to perform feature selection first. By function "sequentialfs" in MATLAB, we choose 42 features for rating predictions and 28 features for popularity change predictions. The dependent continuous variable used here is the same as the one used in the linear regression.



Linear Regression

We begin by running a linear regression with the selected features. Although in the setting, the dependent variable is discrete, we know it comes from a continuous underlying variable. For rating, it is the average rating of all the reviews. For popularity change, it is the review number growth rate. Thus, we back out the underlying continuous variable from the data and perform the linear regression. For prediction error, we can predict the continuous dependent variable first, and then categorize into the corresponding class to calculate errors.

Multinomial Logistic Regression

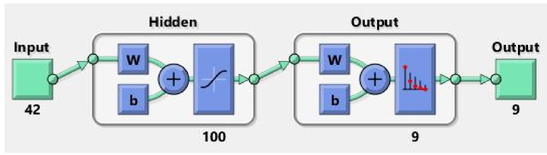
Here we preform standard multinomial logistic regressions on both dependent variables using "mnrfit" in MATLAB. For the rating prediction, there are 9 classes. For the popularity change prediction, there are only two classes.

Naive Bayes

Here we preform standard Naive Bayes on both dependent variables using "fitcnb" in MATLAB. Since most variables are discrete, we discretize all the rest features in this part so that we can choose multinomial distribution as the data distribution.

Neural Network

Here we construct the neural network as a three-layer model with hidden layer of size 100.



Note that this is the graph for rating predictions. The other is almost the same except the number of inputs and outputs (28 and 2). We use "patternnet" in MATLAB to perform the ananlysis. Here, 80% of the data are used for training and the rest 20% are for cross validation.

Support Vector Machine

We use "fitcsvm" in MATLAB to perform SVM analysis. Since SVM only works for binary classification, we only perform SVM for the popularity change predictions. We choose linear kernel function with polynomial order of 3.

Results

The following two tables summarize the training errors and test errors of previous methods.

Rating Prediction	Train Error	Test Error In TOR	Test Error near TOR
Linear Regression	0.6872	0.6851	0.7114
Logistic Regression	0.6738	0.6714	0.7208
Naive Bayes	0.7166	0.7503	0.7841
Neural Network	0.7148	0.7206	0.7377

Table 1: Rating (Multinomial) Prediction Results.

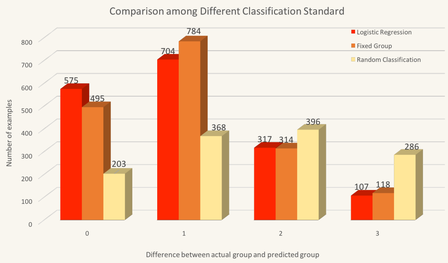
Trending Prediction	Train Error	Test Error in TOR	Test Error near TOR
Linear Regression	0.4219	0.4220	0.4563
Logistic Regression	0.2721	0.2795	0.3586
Naive Bayes	0.2927	0.3052	0.3932
Neural Network	0.2953	0.2813	0.3773
SVM	0.2951	0.2807	0.3782

Table 2: Trending (Binary) Prediction Results.

Discussion

We can see that logistic regression preforms better than the other methods. One possible explanation is that the assumptions for other models are problematic, and logistic regression is more robust to problematic model assumptions.

We also notice that the difference between training error and test error in Toronto is small, suggesting that we did good in feature selection and avoiding overfitting. The test error is significantly bigger for surrounded areas though, implying that our model estimation might be very localized.



However, the prediction needs further improvement. We compare our best predictor-logistic regression with a random-number predictor, and a constant-number predictor. As we can see, the logistic predictor is only slightly better than the constant-number predictor.

Future Work

It is clear that the data in Yelp is not enough for accurate prediction. In the future, we need to collect more data, for example about taste, waiting time, server etc. Also, we mostly use standard machine learning techniques, without customizing into this setting. If time allowed, we can try variations of these models.

Reference

[1] https://www.yelp.com/dataset_challenge

[2] J. Huang, S. Rogers and E. Joo. "Improving Restaurants by Extracting Subtopics from Yelp Reviews". Social Media Expo, 2014.