Predicting Restaurants' Rating And Popularity Based On Yelp Dataset

Yiwen Guo, *ICME,* Anran Lu, *ICME,* and Zeyu Wang, *Department of Economics, Stanford University*

***Abstract*—Every business wants to know whether it can succeed in the future. For restaurants, rating on Yelp is one of the most important indicators. It not only reveals restaurants' quality and services, but also helps to attract more customers. This project focuses on predicting ratings and popularity change of restaurants. With data from Yelp, we uses several machine learning methods including logistic regression, Naive Bayes, Neural Network, and Support Vector Machine (SVM) to make relevant predictions. While logistic regression performs better than the others, predictions from all the methods are far from perfect. This implies the potential improvement of more data and more suited methodologies.**

***Keywords*—Yelp, rating prediction, trending prediction, machine learning.**

I. Introduction

As a great platform for choosing restaurants, Yelp is one of the most popular apps nowadays. It allows people to get a holistic view on a particular restaurant based on its basic information, pictures, reviews, and so on. The rating of restaurants on Yelp also becomes a very important indicator of whether a restaurant is successful and popular. On one hand, high ratings on Yelp show that the restaurant provides quality food and services. On the other hand, high ratings can attract more people to the restaurant, making it even more popular. Thus, it is of great interest for restaurant owners to have a rough idea about how their restaurants perform on Yelp. In this project, we would like to predict ratings of restaurants on Yelp and popularity change based on restaurant features, such as available services, price level, locations, opening hours, etc. This project can not only shed lights on what customers value the most about a restaurant, but also provide suggestions on what feature combinations one should choose when opening a new restaurant, and how likely this restaurant can succeed.

To be more specific, we take restaurants in Toronto, a city representing a variety of cuisine choices as our main target. The input to the algorithm is related feature variables about a restaurant, such as price range, noise level, service available, food type, location, opening hours etc. We then use linear regression, logistic regression, Naive Bayes, Neural Network and SVM to output a predicted Yelp rating of the restaurant, and a prediction on the restaurant's popularity change. Here the popularity change is a binary prediction - increasing or decreasing popularity. The rating prediction is a multinomial prediction ranging from 1 star to 5 starts.

II. Related Work

Due to the rich information contained in the Yelp dataset, many past research and projects tried to use it to predict ratings of restaurants and to evaluate the future development. For example, Kong, Nguyen and Xu [2] classified restaurants based on cultural categories and analyzed international restaurants success mostly with Gaussian Discriminant Analysis (GDA). Several other previous papers focused on the sentiment analysis with text content from Yelp. Xu, Wu and Wang [5] combined the customer reviews and ratings together to conduct sentiment analysis, while Gingerich and Bochkov [1] mainly used matrix factorization to analyze text information and predict Yelp ratings. Linshi [3] worked on user-based text analysis on Yelp rating prediction. He showed that how Yelp user experience can be improved from rating prediction. Other than Yelp review, Tang, Qin, Liu and Yang [4] introduced neural network to predict movie reviews. They claimed that matrix-vector multiplication would be more effective than vector concatenation when considering text analysis. So far, most research works on text analysis of customer reviews, but leaves out other features in Yelp Dataset Challenge [6]. In this project, we apply non-text features to predict restaurants ratings and aims to work on a region-based analysis instead of a user-based analysis in order to provide suggestions to Yelp restaurants.

III. Dataset And Features

The data comes from Yelp Dataset Challenge [6]. It is a small subset of Yelp data, including information about local businesses in 12 metropolitan areas across 4 countries. The dataset contains restaurant information such as location, opening hours, price level, food type, service provided etc. It also has review data available, including the review text, time and rating. For each restaurant, 74 features are included at last.

In this project, we mainly focus on restaurants in Toronto area. In all there are 6750 restaurants in the city Toronto. We randomly choose 5000 of them as the training dataset, and the other 1750 restaurants as the testing dataset. Figure 1 is the Yelp rating distribution of the 6750 restaurants. There is enough variation in the rating for prediction purpose.

To further test the model and interpret the prediction results, we also select about 5000 restaurants near Toronto as another set of testing sample. These restaurants are chosen according to the following distance threshold.

Let $lat_i,\ long_i$ be the latitude and longitude of restaurant i. Define the following terms.

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

$$long_{\text{Tor}} = \frac{\sum_{i \text{ in Toronto}} long_i}{\sum_{i \text{ in Toronto}} 1}.$$

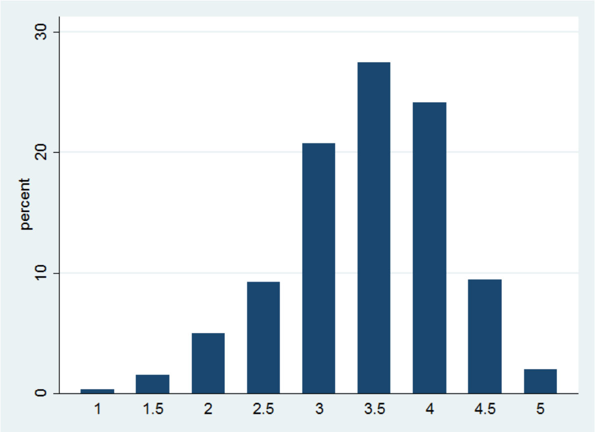

Fig. 1. Rating distribution

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

Any restaurants with $dist_i <= 0.2$ and i not in Toronto are chosen as the second set of testing samples. Intuitively, we first find the location of the "center" of Toronto: lat_{Tor} and $long_{\text{Tor}}$. Then we choose nearby restaurants based on Euclidean distance.

Another data pre-processing before the analysis is about the popularity change. As mentioned earlier, one of the project's goal is to predict a restaurant's popularity change. However, this is not some information we can get directly from the raw data. Thus, we approximate the popularity change in the following way.

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

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

If $trend_i$ is bigger than 1, we say there is a downward trending. Otherwise, there is an upward trending. Intuitively, we compare the number of reviews a restaurant receives in its first half of "life" and the second half, to have a rough idea on the popularity change. Notice that we also discount the number of reviews received by the total number of reviews on Yelp in that year. This is to offset the increasing popularity of Yelp as an app, and any environmental effects that will affect all the restaurants, for example the financial crisis in 2008.

The last piece of data pre-processing is the feature extraction from the variable "category" in the raw dataset. "Category" is a string variable that contains a lot of tags about the business. For example, as long as a business has the tag "restaurant", we consider it as a restaurant and include it in our sample. Other than the "restaurant" tag, it also has information about the food type, for example, Chinese, Korean, fast food, breakfast, barbeque etc. To better represent this information in our analysis, we create corresponding dummy variables for each of these hot tags.

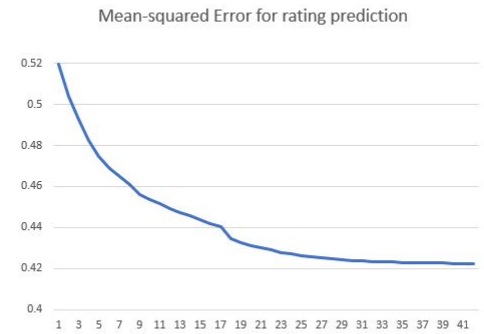

Fig. 2. MSE for rating prediction

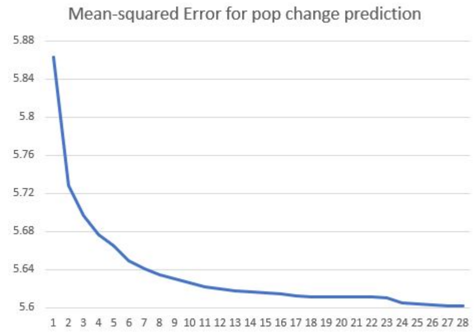

Fig. 3. MSE for popularity change prediction

IV. METHODS

As mentioned above, we want to make predictions on both the Yelp rating and popularity change. Thus, most of the methodologies here are applied to both predictions (except SVM, which only applies to binary predictions). In addition, for all the methods discussed here, we train the model with the 5000 restaurants in city Toronto, and then test it with both the 1750 restaurants in city Toronto, and 5700 restaurants near Toronto. The first test set is to test how well our model performs when the test set is similar to the training set, while the second test set is to test how well our model applies to other regions.

Since we extract variables as many as possible from the raw dataset, we are not sure if these models will runs into the problem of overfitting. Therefore, we first conduct a forward feature selection before the formal analysis, using the function "sequentialfs" in MATLAB. The loss function we choose here is the sum of regression error from the linear regression model (the details will be discussed later in the linear regression subsection). Figure 2 and Figure 3 show how the mean-squared error improves when we add more features into the model. We can also see that at the end, adding more features will not improve the predictions anymore. Overall, we choose 42 features for rating predictions and 28 features for popularity change predictions.

The followings are all the models we try to predict the rating and popularity change. Note that the rating prediction is multinomial, which can be classified into 9 levels from 1 to 5 stars, with 0.5 increment. The popularity change is binary, with either upward or downward trending.

A. Linear Regression

To begin with, we first run a simple linear regression with the selected features from the forward feature selection. One problem with linear regression is that the dependent variable should be continuous, instead of discrete. Due to the specialty of our setting, this conflict can be solved. Though the predicted variable is discrete, the discrete variable actually comes from a continuous underlying variable. For rating of restaurants, the underlying variable is the average rating of all the reviews. Rounding the underlying variable to the nearest half star, we can get the Yelp rating of restaurants. For popularity change, the underlying variable is the review number growth rate, $trend_i$. With the Yelp data available, we can back out both underlying continuous variables. Then running the linear regression is straightforward. To calculate the prediction error, we can first predict the continuous variable based on the linear regression model, and then get the discrete prediction based on the discretization rule. In this way, we can compare performance of linear regression with other machine learning methods.

B. Multinomial Logistic Regression

Next in this project, we perform standard multinomial logistic regressions for both predictions using ”mnrfit” in MATLAB. Multinomial logistic regression predicts the probabilities of the dependent variable falling into each of the category and classifies the prediction into the class with the highest probability. With the parameter θ, it applies softmax function to compute the probability for each category:

$$P(y=i|x;\theta)=\frac{e^{\theta_i^T x}}{\sum_{j=1}^{k} e^{\theta_j^T x}}.$$

To estimate the parameter θ, maximum a posteriori estimation (MAP) is often used, which is an extension of maximum likelihood estimator with regularization.

C. Naive Bayes

In this subsection, we conduct the Naive Bayes method through function ”fincnb” in MATLAB. The basic idea is the following: From Bayes’ Theorem, we know that

$$P(y|x_1,...,x_n)=\frac{P(y)P(x_1,...,x_n|y)}{P(x_1,..,x_n)}.$$

Assuming conditional independence, we can rewrite the equation as the following:

$$P(y|x_1,...,x_n)=\frac{P(y)\prod_{i=1}^{n}P(x_i|y)}{P(x_1,..,x_n)}.$$

Then the class with the highest conditional probability becomes the prediction. In our case, we choose multinomial distribution for the distribution of X, since most of the dependent variables are discrete. For the few continuous variables, we discretize them in this subsection.

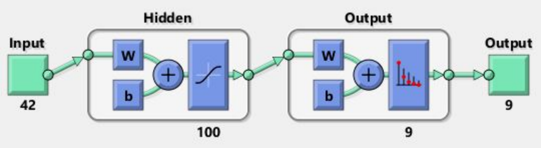

Fig. 4. Flow chart of Neural Network, Rating Prediction

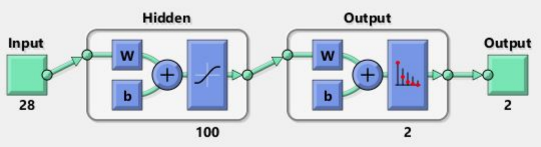

Fig. 5. Flow chart of Neural Network, Popularity Change Prediction

D. Neural Network

We also perform neural network for both predictions because there can be some hidden factors behind the selected features that influence our prediction results. Here we construct the neural network as a three-layer model with hidden layer of size 100. For activation functions of the hidden layer, we choose sigmoid function. For activation functions of the output layer, we choose softmax, which is actually sigmoid for popularity change prediction. Figure 4 and Figure 5 demonstrate the basic neural network structure for both predictions. Notice that the only difference between the two structures is the number of inputs and outputs.

We use the pattern recognition network tool in MATLAB to perform the analysis. 80% of the training data are used for model training and the rest 20% are for cross validation. The tool trains the model through scaled conjugate gradient backpropagation. The training stops when generalization stops improving, indicated by an increase in the cross-entropy error of the validation samples.

E. Support Vector Machine

The last machine learning method we use in the project is SVM, by function ”fitcsvm” in MATLAB. Since SVM only works for binary classification, we only apply SVM for the popularity change predictions. The problem in our case is clearly non-separable, so we use a soft margin. The minimization problem in our case is the following:

$$\min_{\beta,b,\xi}\left(\frac{1}{2}\beta'\beta+C\sum_j \xi_j\right)$$

such that

$$y_j f(x_j) > 1-\xi_j$$

$$\xi_j \geq 0$$

$$f(x)=x'\beta+b$$

The kernel function used here is linear function with polynomial order of 3.

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 I. 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 II. POPULARITY CHANGE (BINARY) PREDICTION RESULTS.

V. RESULTS AND DISCUSSION

To have an overview of how different methods perform, we summarize the train error rates and test error rates for all the previous methods in Table 1 and Table 2.

For the rating prediction, we can see that linear regression and logistic regression perform slightly better than the Native Bayes and Neural Network. One possible explanation is that the first two methods apply to a wider range of problems, and are more robust to problematic model specifications. For example, in the Naive Bayes model, we need to assume conditional independence among the independent variables, which clearly cannot be the case. Another possible explanation is that we don't have enough input features to make a decent prediction. A more complicated method might involve more noise and thus yield low accuracy when the information is limited. With more features that can capture the factor affecting ratings, and also more training samples, the last two methods should improve their performance significantly.

The error rate is just a rough overview of how good the predictor is. In our setting, even though a prediction is wrong, we also want to know how much deviated the prediction is: for a restaurant with a 4.5-star rating, classifying it as a 1-star restaurant is a totally different story from classifying it as a 4-star restaurant, even though both predictions are wrong. Therefore, here we need the help of confusion matrix. For the limit of space, we will only include the confusion matrix for our "best" predictor - logistic regression in Figure 6, Figure 7, and Figure 8.

From the confusion matrix, we can see that even when the prediction is wrong, our predicted value is still clustered around the true value. This implies that even if our prediction is not 100% accurate, it still provides a good estimate of the true value. Also, we can see that the first two graphs are very much the same, indicating similar prediction accuracy between training sample and testing sample in Toronto. However, the third graph is more spread out compared to the first two graphs, showing that the prediction for the restaurants near Toronto is not that accurate.

For the popularity change prediction, we can see that the linear regression model performs poorly. Therefore, the linear model might not be a good fit for the data in this case. All the other methods have a decent performance, while logistic regression, again, becomes our best predictor. The reason mentioned above when analyzing results for the rating predictions can also apply to this prediction. With more

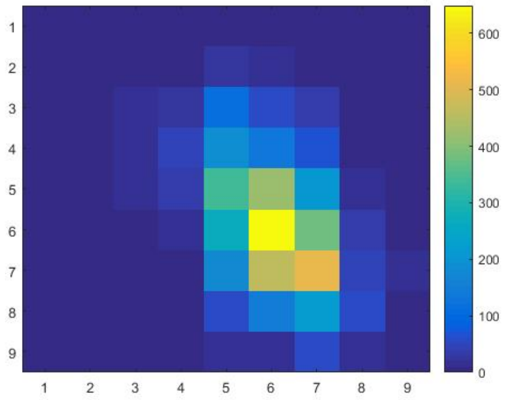

Fig. 6. Confusion Matrix of Logistic Regression, training

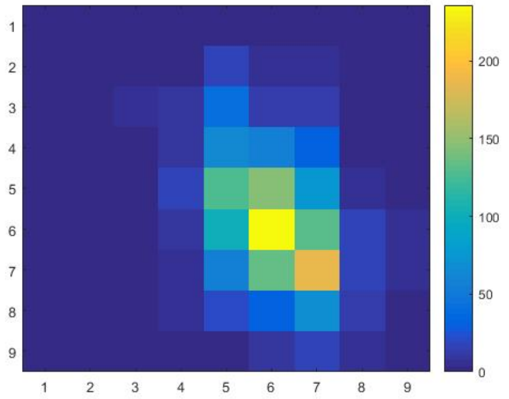

Fig. 7. Confusion Matrix of Logistic Regression, test in Toronto

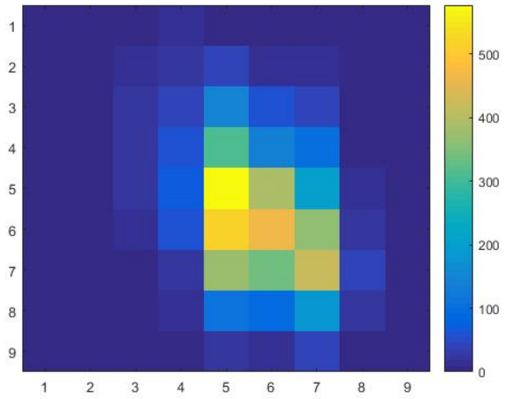

Fig. 8. Confusion Matrix of Logistic Regression, test near Toronto

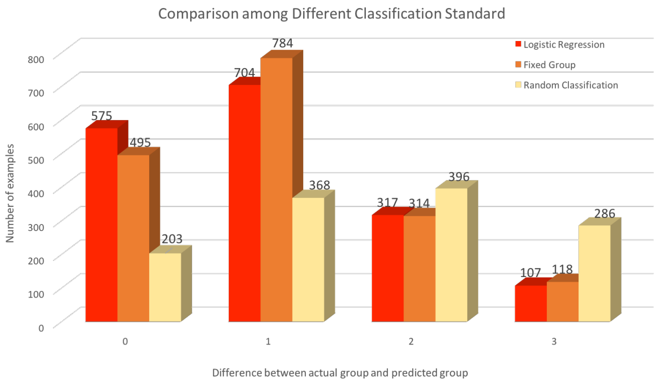

Fig. 9. Comparison between logistic regression predictor, random-number predictor, and constant-number predictor

relevant input features, the last three methods should perform better. Taking a look at the confusion matrix, we find that the prediction is much more accurate when the restaurant experiences downward popularity change.

Overall, from Table 1 and Table 2, we can clearly see that the difference between training errors and test errors in Toronto is almost negligible across both predictions and all methods. This suggests that the forward feature selection we conduct at the very beginning is quite successful in solving the problem of overfitting. However, the error rate is increasing significantly when we apply our model to the restaurants near Toronto. Since we know that this cannot be the problem of overfitting, the only logical explanation is that the model we estimate in this project is very local and can only apply to the city Toronto. If we want to predict the restaurant rating in other regions, we need to retrain the model with new dataset.

There are still problems with our model, since the error rate is high and there is much space for improvement. In Figure 9, we compare our best predictor-logistic regression with a random-number predictor, and a constant-number predictor. A random-number predictor makes prediction based on a random number generator. A constant-number predictor always predicts the restaurant to be 3.5 stars, since this category has the largest share of restaurants. The X-axis is the difference between the actual classification and the predicted classification. Bars in 0 mean the number of accurate predictions. Bars in 1 mean the number of predictions that are only off by 1 etc. From the graph, we can see that the logistical regression performs significantly better than the random number predictor, but only slightly better than the constant-number predictor: the logistic regression predictor has a higher accurate rate, but when considering the rate of predicting within error of 1, the two predictors perform basically the same.

VI. Conclusion and Future Work

This project performs both multinomial classification in terms of rating prediction and binary classification in terms of popularity change prediction. Our overall prediction accuracy is around 26 to 32 percentage for the multinomial prediction and around 70 percentage for the binary prediction. The best performed method is logistic regression possibly because it is more robust. The result can be utilized to provide restaurant improvement suggestions for business owners in the city Toronto, and to a less extent, the business owners near Toronto.

With all the methodologies tried in our project but still no ideal results, it might be the case that the data and features in our dataset is not enough for accurate predictions. Thus, to make further progress, we might need to collect more relevant data, for example about taste, waiting time, servers etc. Also, we mostly use standard machine learning techniques, without customizing into this setting. If time allowed, we can try variations of these models.

Lastly, in this project we choose not to use text of restaurant reviews in our prediction, since the text of reviews directly contains very relevant information about the review rating. If we think the problem from the prediction perspective, we should have zero information about customers' review when making review rating predictions. However, since reviews can contain much more diversified information than discrete variables, it can be a very interesting practice to use these information to provide improvement suggestions to the business owners, other than focusing solely on the prediction.

VII. Contributions

So far, our group has conducted data preprocessing, feature selection, machine learning method implementations, and result analysis for this project.

All the team members work on the data preprocessing part. Zeyu converts the given Yelp Challenge JSON dataset [6] to csv format through Python and Stata. He also cleans up the dataset and creates dummy variables for cuisine categories. Anran and Yiwen, together, discretize continuous restaurant features into categorical features, so that the dataset can be used for later experiments.

For feature selection, Zeyu conducts forward feature selection in order to choose reasonable features. Furthermore, Zeyu implements multinomial logistic regression for both popularity change and rating prediction, as well as SVM for popularity change. Moreover, he summarizes the results for different methods and writes results analysis in both poster and final writeup.

Yiwen proposes the motivation of this project, which can be seen as a fundamental step for us. Then, she works on Naive Bayes implementation and fits a multinomial Naive Bayes model for restaurants rating predictions. In order to see how our model works in different scope, she performs test on dataset both inside and around Toronto.

Anran performs the linear regression on the preprocessed data as a first step. In addition, she implements neural network with Python, which is later improved by Zeyu with MATLAB. Given the test prediction error, she works on error analysis and tries to avoid overfitting. Also, she comes up with the idea to compare the results among our prediction, and then with a fixed-number predictor and a random-number predictor.

Overall, our group members collaborate closely and really enjoy working together.

REFERENCES

[1] Gingerich, Travis, and Yevhen Bochkov. Predicting Business Ratings on Yelp. Stanford University. 2015.

[2] Kong, Angela, Vivian Nguyen, and Catherina Xu. Predicting International Restaurant Success with Yelp. Stanford University. 2016.

[3] Linshi, Jack. Personalizing Yelp Star Ratings: a Semantic Topic Modeling Approach. Yale University. 2014.

[4] Tang, Duyu, Bing Qin, Ting Liu, and Yuekui Yang. User Modeling with Neural Network for Review Rating Prediction. IJCAI. 2015.

[5] Xu, Yun, Xinhui Wu, and Qinxia Wang. Sentiment Analysis of Yelps Ratings Based on Text Reviews. Stanford University. 2015.

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