Predicting New Restaurant Success and Rating with Yelp

Aileen Wang, William Zeng, Jessica Zhang

Stanford University
aileen15@stanford.edu, wizeng@stanford.edu, jzhang4@stanford.edu

December 16, 2016

Abstract

This paper aims to predict new restaurant success and ratings, and understand which restaurant features have the most impact on a restaurant's success. Using the Yelp Challenge Dataset, we ran a Chi-squared test and stochastic gradient descent to identify the restaurant features that hold the most weight. Utilizing a variety of binary and multi-class classification algorithms including logistic regression, SVM, Random Forest and Multilayer Neural Networks, we attempt to predict restaurant success and ratings, rounded to the nearest star. We find that the two algorithms that perform best are Random Forest and Multilayer Neural Networks, with accuracies of around 60% for binary classification and 56% for multi-class classification. Finally, we performed sentiment analysis on restaurant reviews and improved accuracy up to 85% using Chi-squared feature extraction with Random Forest and Multilayer Neural Networks algorithms.

I. Introduction

New restaurants fail at a surprisingly high rate: 59% within the first year of opening and about 80% within the first five years [1]. In this highly competitive field, there are several factors that can make or break a restaurant's success. We want to know the cause behind this more-than-half failure rate, and how we can predict restaurant success given its information before it opens. The findings from this project can help predict the success of a new restaurant given its current attributes, and give insight about what attributes can be improved to increase its success.

II. Related Work

Being a highly robust dataset, the Yelp Challenge Dataset has been used in several research projects to predict restaurant success. A paper by Vasa, Vaidya, Kamani, Upadhyay and Thomas also utilized the Yelp dataset to predict restaurant success using binary classification [2]. They investigated only restaurants in the Phoenix area and relied heavily on the type of food that restaurants served to make their predictions, attributing the importance of the food to the demographic layout of the city. They also used various prediction models from regression to decision trees, and determined that their best model was neural networks with a test accuracy of 0.51. We hope to generalize our predictions across many different cities across the country, and discover which features are important for restaurant success independent of demographics. We also aim to focus our predictions towards new restaurants rather than well established ones.

Another paper by Farhan at UCSD attempted to predict individual review rating rather than overall restaurant rating using only restaurant attributes and the time of review, and chose Naive Bayes, neural networks and decision trees as his models [3]. These models produced mediocre accuracy rates and the author attributed the errors to the "highly unpredictable" data and the features he selected. While Farhan predicted the rating for a review throughout the year given the restaurant at-

tributes, our project predicts the overall rating of a restaurant given its attributes and by running sentiment analysis on a few reviews. In addition, while Farhan manually selected his features, our project aims to select an optimal set of features using Chi-squared feature selection.

III. Problem Definition

The goal of our project is to:

- Find the optimal subset of features to produce the best prediction and achieve the maximum accuracy for restaurant success
- Predict restaurant success (binary classification): if it has more than 35 reviews and an average rating of 3.5 or above
- Predict restaurant rating (multiclass classification): measure the extent to which the restaurant is successful
- Identify the important features from text reviews and perform the corresponding sentiment analysis to produce the best prediction for restaurant success

IV. Data Processing

i. Yelp Dataset

For our project, we decided to use Yelp as our source for restaurant information. Yelp is an extremely popular, easy-to-use platform that people use internationally to publish reviews and rate restaurants. As of 2016, there are 3 million claimed businesses, 150 million monthly users, and over 100 million reviews. We will use the Yelp Challenge Dataset, freely provided for academic use, which for each restaurant, contains the number of reviews, rating, and over 33 types of business attributes, such as Price Range, Takes Reservations, Parking, Ambience, Attire, Good For, etc.

There are 26729 restaurants in the Yelp dataset from 4 countries and 10 different cities across the world. In order to reduce the number of restaurants to a manageable size, we only kept restaurants that had a total of 35 reviews or more. In addition, by filtering restaurants by the number of reviews, we minimize the chance of outliers, since restaurants with a greater number of reviews are less likely to be affected by outlying user reviews. We define a successful established restaurant as follows: **A restaurant is considered successful if it has more than 35 reviews and an average rating of 3.5 or above**.

Datasets were prepared for both binary classification and multiclass classification. For binary classification, a restaurant's feature vector was mapped to whether it was successful or not, given the definition above. For multiclass classification, a restaurant's feature vector was mapped to the restaurant's average star rating, rounded to the nearest star. Thus, there were 5 possible classes for the feature vector, from one star to five stars.

ii. Feature Extraction

A Feature Value Conversion

There are many different type of features, including binary (offers take out), string enumeration (ambience), and multi-option (types of parking). To reduce the number of features and simplify the classification process, we converted all feature values to integers as follows. We attempted to preserve the inherent value of the feature in its integer representation wherever possible. For example, a loud noise level would have a greater value than average, which would in turn have a larger value than quiet.

- **Binary Value**: 1 is true, 0 is false.
- **String Enumeration Value**: For example: "Attire" has values: casual, dressy, or neither. Thus, casual = 1 , dressy = 2, neither = 0.
- **Multi-Option Value**: treat as binary number, then convert to integer. For example, "Parking": "garage": true, "street": false, "validated": false, "lot": true, "valet": false will be treated as binary number 10010. Convert 10010 to the equivalent integer, 18.

B Choosing Optimal Features

Restaurants have a variety of attributes. What is the optimal subset of features to choose to produce the best prediction? To retrieve the optimal subset of features, we will perform the following steps:

- Perform a chi-squared test on the samples to extract the best set of features with the highest chi-squared score. We will invoke *fit_transform* method of **SelectKBest** from **SKLearn** package to select features according to the k highest scores. The subset selected by chi-squared feature selection will boost prediction performance and allow us to achieve the maximum accuracy.
- Use stochastic gradient descent (SGD) to determine the weights of the chosen subset of features. Features associated with any negligible weights will be ignored.

V. Approaches

i. Baseline

We used the baseline as the lower bound check for our classification accuracy. We implemented a simple stochastic gradient descent with a learning parameter of $\eta = 0.01$ to determine the weights of each feature. We split the restaurant data into two groups, trained the classifier on 5988 restaurants and tested the classifier for 3992 restaurants. For our baseline we found a train accuracy of 0.4097 and a test accuracy of 0.5646.

ii. Oracle

We individually attempted to assess the success of 30 restaurants based solely on their feature set using our own human intuition. We got a 76.67% classification accuracy for whether a restaurant was successful or not. This doesn't necessarily serve as an upper bound for our algorithm's performance, since machine learning can definitely perform better than human intuition, especially when given thousands of restaurants.

iii. Classification Algorithms

All classification algorithms described below were implemented in binary and multiclass classification, as described in the data processing section. Binary classification predicted whether a restaurant was successful or not, while multiclass predicted a star rating rounded to the nearest star.

A Logistic Regression

We used the logistic regression method in the **linear_model** class of **SKLearn**, with a regularization parameter of $C = 0.1$. While logistic regression is primarily binary, we used the multiclass variant of the method for predicting the star rating. Logistic Regression assumes that features are roughly linear and the problem is linearly separable. Additionally, it is robust to overfitting and noise. We varied the strength of regularization, finding that found that more regularization - a smaller C-value - yielded higher test accuracy for our model. We observed a train accuracy of 0.6067 and a test accuracy of 0.5934 for binary classification, and a train accuracy of 0.5571 and test accuracy of 0.5534 for multiclass.

B Decision Tree

We used the decision tree class in the **tree** class of **SKLearn** for both binary and multi-class classification. One advantage of using decision tree is that its performance is not affected by nonlinear relationships between features. However, decision trees tend to be sensitive to changes in data and overfit data. We observed a train accuracy of 0.9340 and a test accuracy of 0.5746 for binary classification, and a train accuracy of 0.9222 and test accuracy of 0.4947 for multiclass. Furthermore, the optimal number of features for decision trees is 9.

C Random Forest

We used the random forest classifier in the **ensemble** class of **SKLearn**. Since Random Forest uses averaging to improve accuracy and

control over-fitting, it performed more optimally than Decision Trees. We observed a train accuracy of 0.9223 and a test accuracy of 0.5999 for binary classification, and a train accuracy of 0.9087 and test accuracy of 0.5361 for multiclass. Furthermore, the optimal number of features for random forest is 19.

D Multi-layer Neural Network

We used the MLPClassifier (multi-layer perceptron) class in the **neural_network** package of **SKLearn** with a quasi-Newton solver 'lbfgs'. Neural networks use multiple hidden layers to learn a non-linear function. We observed a train accuracy of 0.6391 and a test accuracy of 0.6391 for binary classification, and a train accuracy of 0.5903 and test accuracy of 0.5624 for multiclass.

E Support Vector Machines

We used the SVC class in the **svm** package of **SKLearn**. SVMs treats feature vectors as high-dimensional points in space, which it tries to separate with a hyperplane in order to create the largest margin between the points and the decision boundary. We observed a train accuracy of 0.5695 and a test accuracy of 0.5531 for binary classification, and a train accuracy of 0.5316 and test accuracy of 0.5263 for multiclass. Furthermore, it used significant more computation time compared to the other algorithms, taking 20 seconds to run compared to less than one second more most algorithms.

F K-Means Clustering

We used the KNeighborsClassifier class in the **neighbors** package of **SKLearn**. K-means is an unsupervised learning algorithm that groups the features into a set number of clusters. We set the number of clusters to 2 for binary classification and 9 for multiclass classification. We observed a train accuracy of 0.7719 and a test accuracy of 0.5689 for binary classification, and a train accuracy of 0.6425 and test accuracy of 0.5506 for multiclass.

G AdaBoost Classifier

We used the AdaBoostClassifier class in the **ensemble** package of **SKLearn**. Adaboost applies classifiers to additional copies of the dataset in order to focus more on the difficult classifications. We observed a train accuracy of 0.6436 and a test accuracy of 0.6278 for binary classification, and a train accuracy of 0.3385 and test accuracy of 0.3359 for multiclass.

H Naive Bayes

We used the GaussianNB class in the **naive_bayes** package of **SKLearn**. Naive Bayes looks at each feature independently and assigns probabilities to each feature value based on the y value. We observed a train accuracy of 0.5033 and a test accuracy of 0.5065 for binary classification, and a train accuracy of 0.0746 and test accuracy of 0.0729 for multiclass.

iv. Evaluation Metrics

We are using *accuracy_score* method in **SKLearn** package to measure both the training and testing performance of the above classification algorithms, which is calculated by dividing the number of correct predictions by the total number of test data examples.

VI. Sentiment Analysis of Text Reviews

In addition to the core restaurant attributes, customer review text are also important indicators of restaurant success. In this section, we will identify important features from text reviews that were not part of restaurant attributes, such as "great", "good", "nice" from positive reviews (Figure 1) and "bad", "nasty", "terrible" from negative reviews (Figure 2). Then we perform the sentiment analysis on these important features. Our sentiment analysis is divided into the following steps:

- **Preprocessing:** Using the business id, group all related text reviews of the same restaurant into one text review with space separators.

Figure 1: *Word Cloud of Positive Review*

Figure 2: *Word Cloud of Negative Review*

- **Feature Extraction:** Extract the words from the grouped text reviews except the stopwords and calculate the word frequencies. We invoked the *fit_transform* method of **TfidfVectorizer** in the **SKLearn** package to convert a collection of grouped text reviews to a matrix of features.
- **Identifying Optimal Features:** Use chi-squared test to extract the best set of features with the highest chi-squared score in the same process as in Section IV.
- **Prediction** Apply algorithms in Section V on the extracted features above to predict restaurant success.

VII. Results and Analysis

We first use the *train_test_split* method in the **SKLearn** package to randomly select 5988 for the train dataset and 3992 for the test dataset from the dataset with 9980 elements defined in Section IV. Then we run the classification algorithms in Section V on this dataset configuration and obtained the following results:

Figure 3: *Optimal features for Binary Classifier*

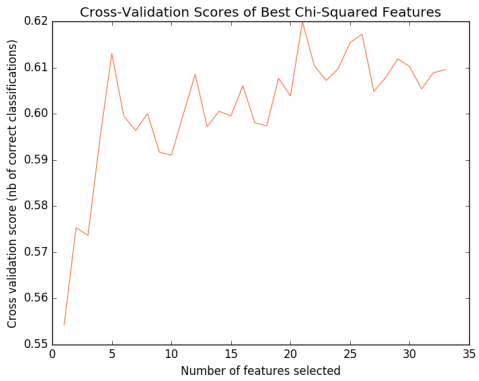
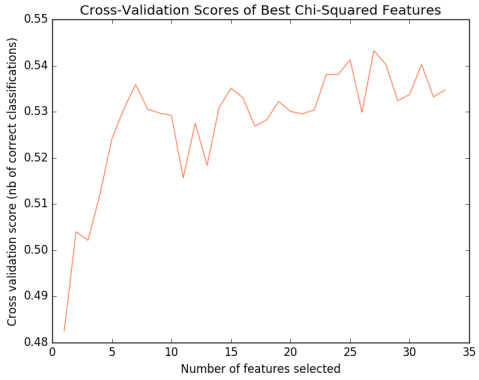


Figure 4: *Optimal features for Multi-Class Classifier*



i. Optimal Features Selection

After running Chi-squared as the scoring function, we determined that the optimal number of features to be used were 21 for binary classification and 28 for multiclass classification (see Figure 3 and 4). After running SGD algorithm (Figure 5), we found that the features that were most highly weighted were parking, good for, attire, ambience, and ages allowed. Other features that were generally correlated were smoking, noise level, BYOB/Corkage, happy hour, and good for groups.

ii. Classifier Accuracy Comparison

After running our binary and multiclass classifiers on the optimal selected features, we observed that Random Forest and MLP per-

Figure 5: *Weights of Restaurant Features*

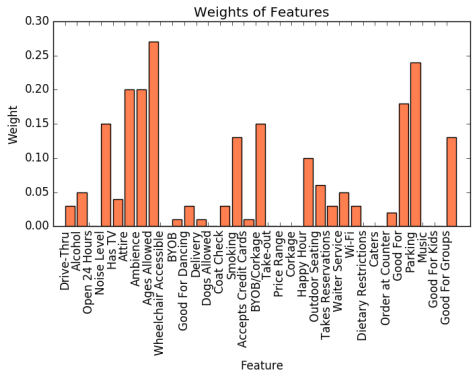
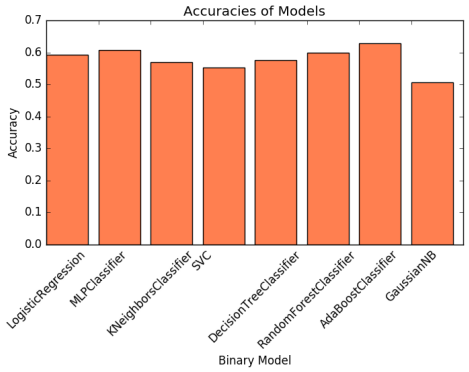


Figure 6: *Binary Classification Results*



formed the best (See Figure 6 and 7). Since both these algorithms do not expect linear features or features that interact linearly, they perform more optimally than other models that assume linearity and are sensitive to outliers in data. We can expect that a restaurant's features will not be interact linearly, since oftentimes certain groupings of features are better than others. For example, a restaurant can have a high rating with a classy ambience and low noise level, but a restaurant with a casual ambience and a high noise level can also be successful. In addition, we observed that Random Forest performed better than Decision Tree, since Random Forest uses averaging to improve accuracy and control over-fitting.

In addition to running the classification algorithms on all the restaurants, we ran our

Figure 7: *Multiclass Classification Results*

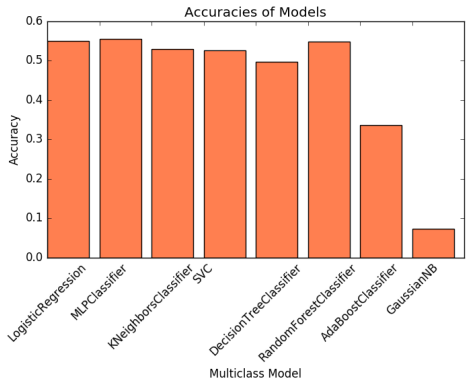
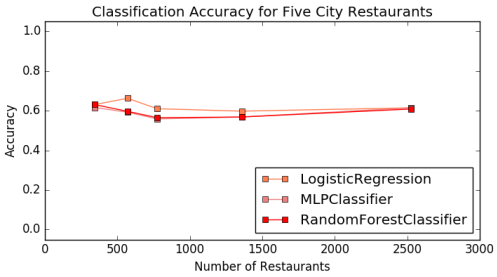


Figure 8: *Classification Comparison on 5 Cities*



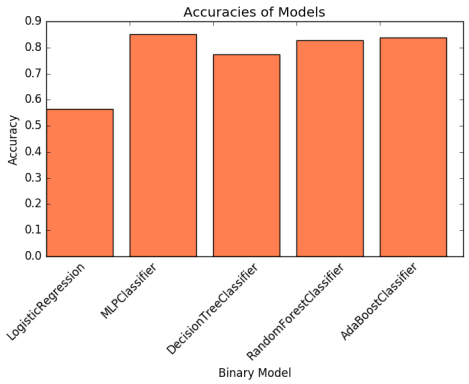
binary and multiclass classifiers separately on restaurants from 5 different cities within the US. From Figure 8, we observed that the prediction accuracy for Random Forest, MLP, and Logistic Regression are stable and consistent.

iii. Sentiment Analysis Result

After applying the sentiment analysis algorithm in Section V, we can extract thousands of text review features for each restaurant. To reduce the runtime memory consumptions, we only select the top 20 features that have the 20 highest chi-squared scores. Then we apply five algorithms from Section V to predict the restaurant success. After running five algorithms on restaurants from the US, Canada, UK, and Germany, we observed that

- MLP, Random Forest, and AdaBoost are the best and have achieved up to 85% accuracy for binary classification.

Figure 9: *Binary Sentiment Analysis on US Restaurants*



- The top 20 highest chi-squared features are airport, amazing, bad, buffet, buffets, chinese, delicious, denny, falafel, gyro, horrible, hummus, ihop, manager, minutes, pita, terrible, waitress, wings, worst.
- Compared to the binary classification accuracy based on the restaurant core features, these three sentiment analysis algorithms have improved accuracy by 42%. This implies that sentiment features from customer text reviews contains more accurate information about the restaurant success.
- We also tried to increase the number of k best chi-squared features during the classification process. But we didn't see any significant improvement on the binary classification accuracy. For example, if k = 200, the accuracy only increased to 90%.

We also ran five algorithms for the multiclass scenario. From Figure 12, we observed that there is no improvement in terms of the classification accuracy. This is because the sentiment features from customer text reviews usually contains information about the restaurant success and doesn't have the clear corresponding information about restaurant rating.

Figure 10: *Binary Sentiment Analysis on Canada Restaurants*

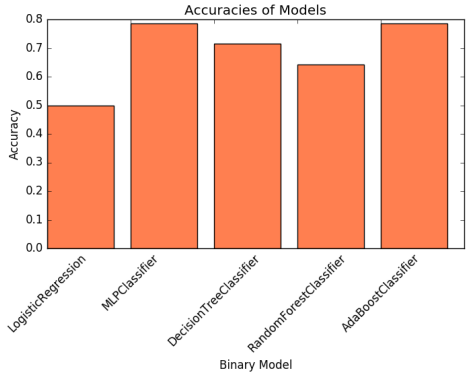


Figure 11: *Binary Sentiment Analysis on Germany Restaurants*

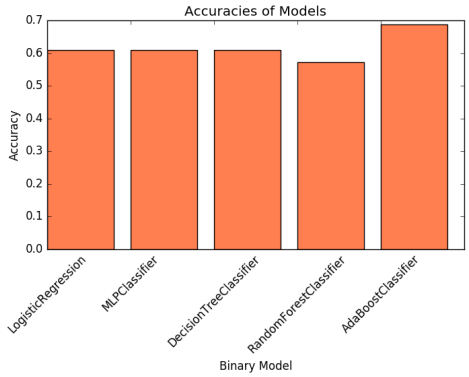


Figure 12: *Binary Sentiment Analysis on UK Restaurants*

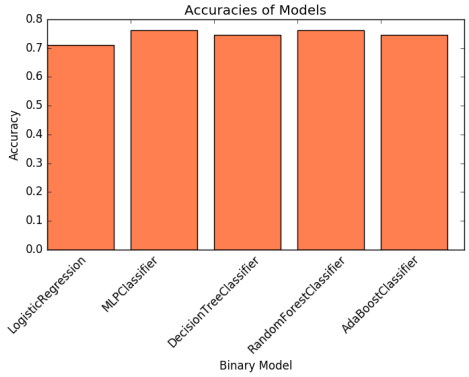
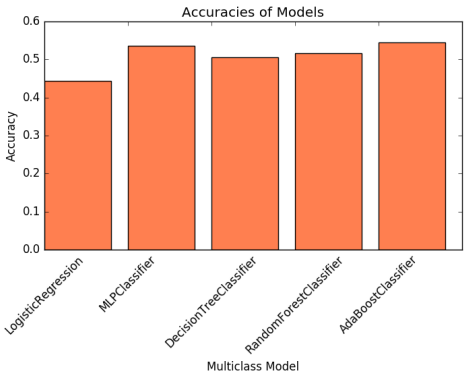


Figure 13: *Multi-Class Sentiment Analysis on US Restaurants*



VIII. Discussion

i. Error Analysis

One of the most important factors that determines a restaurant's rating and whether a user will return is a restaurant's food quality. Unfortunately, this is not captured in a restaurant's attributes. We attempted to make up for this by implementing sentiment analysis, which analyzes the restaurant reviews themselves and helps gain some insight into this attribute. We found that the addition of sentiment analysis increased the accuracy of our classification by 32%, demonstrating that food quality and other factors within user's review text definitely affect a restaurant's average rating.

ii. Limitations

One of the largest challenges was determining the best choice of features and how to represent them. Running Chi-squared and standard gradient descent reduced the number of features, thus preventing overfitting by only focusing on the features that most strongly correlate with restaurant success. Initially, we created a separate feature for each option of a non-binary feature, i.e. Noise Level: quiet and Noise Level: average as two separate features. However, this created way too many features, and didn't relate the values of a single attribute, since they were split up into several features. While the integer representation represents one attribute as one feature, it cannot precisely translate the attribute values into accurate integer equivalents. For example, parking, with values for garage, street, validated, lot, and valet, cannot easily be represented as an integer that reflects the value/weight of each parking option.

Even though sentiment analysis improves the classification accuracy, new restaurants likely won't have too many reviews. Thus, it may not always be possible to use sentiment analysis to improve our predictions of restaurant success and ratings. However, even without sentiment analysis, an accuracy of 56% for multiclass is substantially beneficial.

iii. Future Work

Another section of the Yelp Challenge Dataset that we could possibly utilize is the check-ins. On Yelp, users can check in when they visit a restaurant, generally for some sort of reward which encourages the feature's use. Thus, check-in data could be interpreted as representing how often a user returns to a restaurant, which is likely correlated with how good the food or service is. Thus, this could provide an additional feature to the current feature vector.

Another feature we can use is the type of cuisine, i.e. Chinese, Mediterranean, Italian, etc. which is specified in the Yelp Challenge Dataset. Since some cuisines are on average rated higher than other ones, the type of cuisine is likely an important factor for helping to determine a restaurant's rating.

References

[1] King, Tiffany, Njite, David, Parsa, H.G., and Self, John T. (2005). Why Restaurants Fail *Management*, 46:304–322.

[2] Vasa, Neel., Vaidya, Aditya., Kamani, Shreya., Upadhyay, Manan., Thomas, Mark (2014). Predicting Restaurant Success *University of Southern California*.

[3] Wael, Farhan. (2014). Predicting Yelp Restaurant Reviews *University of California, San Diego*.