Christina Ho, Ria Pinjani, Zixuan Zhou

December 13, 2019

Messy Data and Machine Learning Final Project

Professor Ravi Shroff

Rating the Raters: Bias on Yelp Reviews

I. CONTRIBUTION

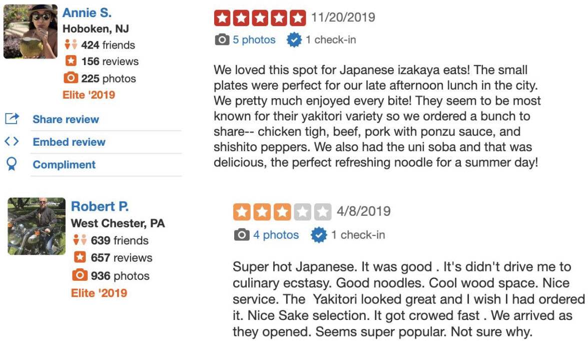
Zixuan Zhou: Zixuan was responsible for combining, cleaning, and merging the different data files -- business, user, and review text files. He also was in charge of the concluding portion of this write-up.

Christina Ho: Christina was responsible for the exploratory analysis, sentiment analysis and introduction and exploration portion of this write-up.

Ria Pinjani: Ria was responsible for the feature engineering and modeling portion of this as well as in the write-up.

II. INTRODUCTION

Yelp is becoming a vibrant bond connecting people with local businesses. Yelp online reviews are a valuable source of information for users to choose where to visit or what to eat among numerous available options – especially in New York City. A common solution to provide a fast overview is to show overall ratings in the form of business stars in a range of 1-5. These reviews and star ratings aid everyday decision making within the business.



Annie S.
Hoboken, NJ
424 friends
156 reviews
225 photos
Elite '2019

Share review
Embed review
Compliment

★★★★★ 11/20/2019
5 photos 1 check-in

We loved this spot for Japanese izakaya eats! The small plates were perfect for our late afternoon lunch in the city. We pretty much enjoyed every bite! They seem to be most known for their yakitori variety so we ordered a bunch to share-- chicken tigh, beef, pork with ponzu sauce, and shishito peppers. We also had the uni soba and that was delicious, the perfect refreshing noodle for a summer day!

Robert P.
West Chester, PA
639 friends
657 reviews
936 photos
Elite '2019

★★★☆☆ 4/8/2019
4 photos 1 check-in

Super hot Japanese. It was good . It's didn't drive me to culinary ecstasy. Good noodles. Cool wood space. Nice service. The Yakitori looked great and I wish I had ordered it. Nice Sake selection. It got crowed fast . We arrived as they opened. Seems super popular. Not sure why.

Fig 1. An example of different standards between two reviewers (bias)

The Yelp text reviews can be subjective and biased towards a user's past experiences and personality as shown in Figure 1. Some people are reluctant to give stars while others are the opposite: even when two people have the same view on a restaurant, one may tend to give a higher star rating while the other gives a lower star rating. In this paper, we attempt to detect bias. Not only does this provide an overview of review text but can cancel out subjectivity.

A biased review tendency is hidden underneath the simple star rating system since we don't know what a costumer's real unbiased opinion is. However, review text is good indication of a costumer's real opinion. Rating a restaurant from one to five takes less than a second, but writing a review costs much more and text conveys much more information, thus we choose to focus on text. In this project, we address this interesting problem by using machine learning techniques to bridge the gap between the personalized rating tendency and the overall rating statistics. From a business and practical perspective, this method could potentially help Yelp to improve the fidelity of its rating system by considering the intrinsic variations of the reviewers' personalities.

As a multinomial classification problem, it is aimed to predict the 5-star rating results based on multiple features. Our features will be generated from the text review alone. We use a combination of two-feature generation methods and four machine learning models to find the best prediction result. Our first approach is to create bag of words from the top frequent words in all raw text reviews and our second is top frequent words/adjectives from results of Part-of-Speech analysis.

III. METHODOLOGY

A. Data

We collected our data from Yelp Dataset Challenge. We use three datasets which contain a total of 192,609 business, 1,637,138 user and 6,685,900 review information. All data files are in json format.

1. business.json**:** Contains business data including location data, attributes, and categories.
2. review.json: Contains full review text data including the user that wrote the review and the business the review is written for. Because the bulk of our feature engineering comes from this dataset, here is the columns of the data:

```
{
 ’type’: ’review’,
’business_id’: (encrypted business id),
 ’user_id’: (encrypted user id),
’stars’: (star rating, rounded to half-stars),
’text’: (review text), ’date’: (date, formatted like ’2012-03-14’),
 ’votes’: {(vote type): (count)}
 }
```

3. user.json: contains user level data and information from their yelp profile

We began our analysis in our Data Cleaning.R file. Original data sets were huge and messy (8.67GB). In order to breakdown the huge dataset, we only looked at categories of businesses that contain “restaurant” to focus on restaurant businesses. After our subset, there is information on 4,841 unique restaurants in total. We randomly chose 1000 restaurants and retained reviews from the most recent four years (2015, 2016, 2017, 2018), which broke down our data set to 858 unique restaurants. In our final dataset, we have 43,834 text reviews written about these chosen restaurants.

B. Exploratory Analysis

As part of our exploratory analysis in Data Exploration.R file, we found that the top cities in our restaurant dataset located in Las Vegas, Toronto, Scottsdale, Montreal, and Henderson (Figure 2). We define top-rated restaurants as having an overall business star rating greater than 3.5 and low-rated restaurants as having an overall business star rating lower than 3. We looked at words that appeared in at least 200 reviews. This makes sense since rare words have noisier measurements (few good or bad reviews could shift balance). They are less likely to be useful in classifying reviews or text. We also filter for ones that appear in at least 10 businesses (others are likely to be specific to a particular restaurant).

In order to find top categories (a text column which describes the type of restaurant) for high-rated and low-rated restaurants, we have to convert the text to a corpus to remove stop words, punctuation, and numbers as well as change the text to lower-case. We converted the text to a term document matrix in order to get the top frequencies of each restaurant category. Top categories from top-rated restaurants include bars, sandwiches, fast-food and cafes. Top categories from low-rated restaurants include pizza, american, and chicken. One thing to note is that we have over 80% missing values in the categories section for each restaurant.

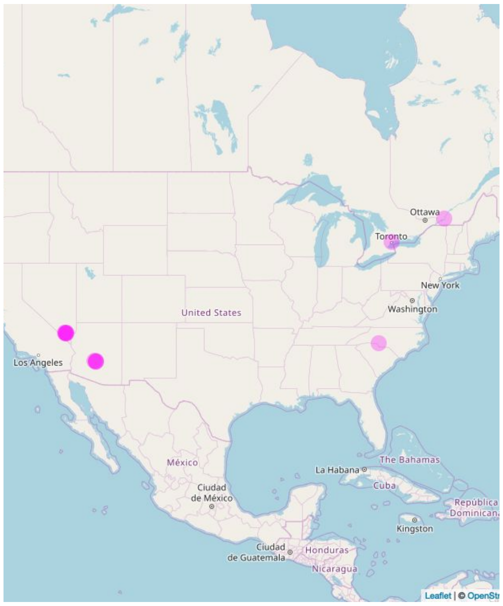


Fig. 2 illustrates the 12 most repeated words in all reviews of our selected 1000 restaurants.

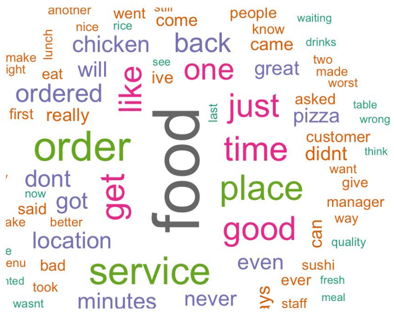


Fig. 3 depicts a word cloud of the top words used in top rated restaurant reviews

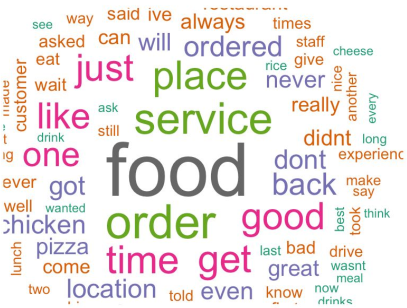
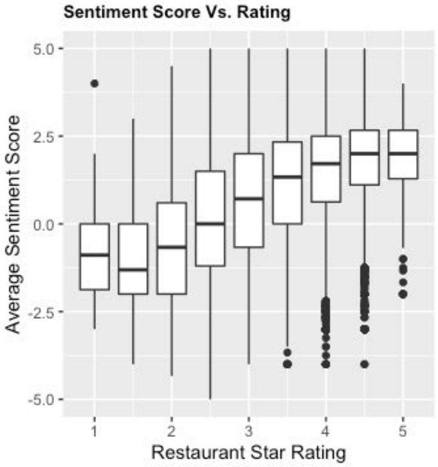


Fig. 4 depicts a word cloud of the top words used in low rated restaurant reviews

Figure 3 and Figure 4 depicts word clouds of review text. We looked at the 43, 834 review texts to see the top words used in reviews, filtered by top rated and low rated restaurants as shown in the figures. For top-rated restaurants, most common words such as “food”, “good”, and “place” have the highest frequency amongst all. For low-rated restaurants, common words such as “food” “good” and “service” have highest frequency amongst all.

C. Sentiment Analysis



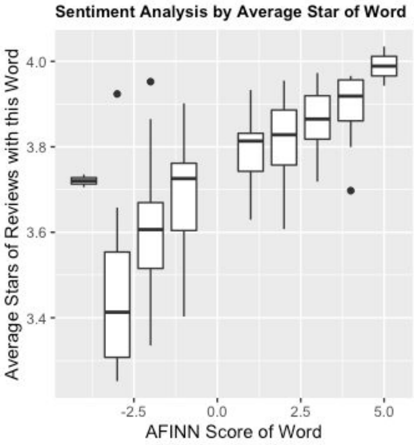


Fig. 5

Fig. 6

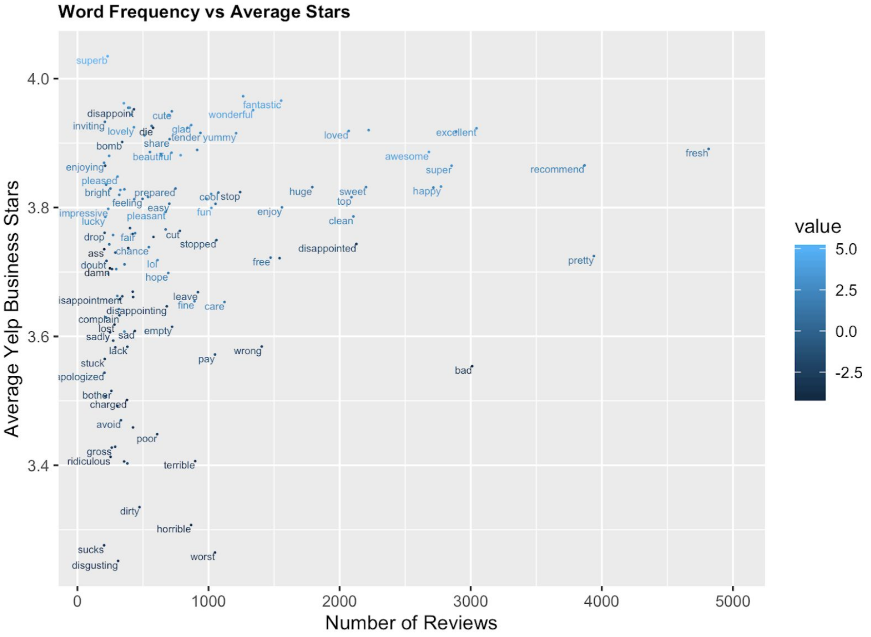


Fig.7

We perform sentiment analysis and compare to a pre-existing lexicon, AFINN. Figure 5 and 6 shows that there is a very clear trend -- the AFINN sentiment analysis works to some extent. Both figures show the higher the average star ratings and sentiment score, the higher the AFINN score of the word. Figure 7 shows the AFINN score of each word plotted against the frequency and average yelp business star the word is from. how the There are some common words that are positive (e.g. " pretty", "fresh") and others that are pretty negative (e.g. "bad", disappointed"). This is a good plot to look at initial misclassifications of the AFINN lexicon and to possibly take out neutral words. We did not carry on with this analysis, though this is good to take into account in the future.

D. Feature Engineering

In order to fit data into models, we formed feature vectors by the following steps. First, we used the same text mining procedure above -- tokenizing text reviews of all restaurants was derived using the tidytext package. Words were also filtered such that any stop words as well as numbers were removed. Then, we picked the top k = 500 frequent words. We ran our models with k = 50, 100, 500, and 1000. We got the lowest RMSE for k = 500 frequent words. Thus, we picked the top 500 words for our analysis. We calculated the frequency of each top k word in all reviews of each business. We also calculated the total number of times the word occurred in the entire dataset. Lastly, we calculated a proportion; number of occurence for word *i* in business *j*, divided by the number of occurence for word *i*. The resulting feature matrix had dimensions equal to k X N where k is the top k words and N is the number of unique restaurants.

Two feature vectors were created by two methods, (i) baseline; (ii) feature engineering II

1. *Baseline: Using Top-K Frequently Used Terms*

   Using the raw text data, we chose the top K frequently used terms in all the 43,834 reviews. The array of words becomes our feature vector. We went through the reviews of each 858 restaurants and counted the number of times each member of the feature vector was used in the reviews for that particular restaurant. We divide the number of occurence with the total number of occurence in all of the top K words to calculate the frequency of our feature vector members with this formula:

$$\mathrm{freq(i)} = \frac{\mathrm{x_i}}{\sum_{\mathrm{i=1}}^{\mathrm{K}} \mathrm{x_i}}$$

   Where *xi* is the number of times the *i-th* top K frequent word appeared in the review of a restaurant. We used K = 1000.

2. *Feature Engineering II: Part-of-Speech Tagging*

   The motivation of doing this feature engineering was that we thought adjectives could give us more information than other kinds of words. Part-of-Speech (POS) tagging is essentially assigning a tag to every word to define if it corresponds to a noun, a verb etc. using the WordNet lexical database. For this model we used the same steps as the baseline model; except we subsetted words to only include adjectives and adverbs using the parts of speech library from tidytext. This led to the removal of two businesses from our analysis; as their reviews did not contain any adjectives or adverbs. We used the same training - testing split ratio as before. To predict ratings with our new set of features, we used the svm function from the e1071 package. We got a rmse higher than the one in our baseline feature matrix, equal to 0.682. The value for rmse decreased to 0.664.

IV. Results

Feature Selection Method	**Learning Model**	**RMSE**
Baseline (Top Frequent Words from Raw Data)	Naive Bayes	**1.251**
	Multinomial Logistic Regression	**0.695**
	Support Vector Regression	**0.772**
	Support Vector Regression-n	**0.779**
Feature Engineering II(Top Frequent Adjectives after POS)	Naive Bayes	**1.354**
	Multinomial Logistic Regression	**0.593**
	Support Vector Regression	**0.781**
	Support Vector Regression-n	**0.771**

Fig. 8

Naive Bayes model was used because it is easy and fast to predict class of test data set and it also performs well in multi-class prediction, where as Multinomial logistic regression is a particular solution to classification problems that use a linear combination of the observed features. We implemented Support Vector Regression because it avoided the difficulties of using linear functions in the high-dimensional feature space.

To evaluate our results, we divided our feature matrix into 90% training and 10% testing datasets. For training dataset, we used both text review and business' star. For testing, we used our models to predict the business rating and then compared it with the actual rating that we had to evaluate the accuracy of our model. Our plan was to try four regression machine learning algorithms to get a solid understanding of how each variable correlated with the number of stars. We chose these four learning methods because they are more interpretable and transparent than blackbox methods. This would also provide enough results to see why one model performed better or worse than another. In order to predict ratings from our feature matrix, we first used the svm function from the e1071 package in R. We then moved on to calculate accuracy from the predicted ratings and got a rmse equal to 0.649. The RMSE decreased to 0.571 when we used normalized features in our svm model.

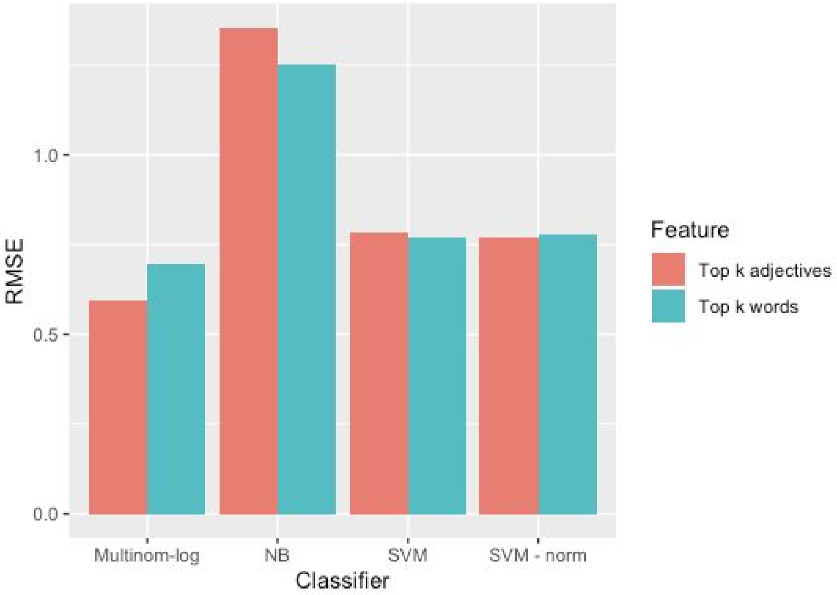


Fig. 9

A. Bias Calculation

With SVM classifier trained on data set, we could compute each reviewer's bias br by:

$$b_r = \frac{\sum_{i=1}^{N_r}(star_i - star_{i,predict})}{N_r}$$

where *Nr* indicates the total number of businesses in our dataset. We compute each restaurant's bias (*br*) by using the real star rating of the i-th business and star i, predict is the predicted star. We divide this difference by *Nr* which indicates the total number of businesses. With our definition of bias we get the statistical distribution of restaurant bias.

We calculated bias for two of our models; the one with the highest rmse (Naive Bayes using top-k adjectives) and the one with the lowest rmse (Multinomial logistic regression using top-k adjective). When predicting on the entire data set, we got an average bias (for all restaurants) equal to -1.01 for the former model and -2.49 for the latter.

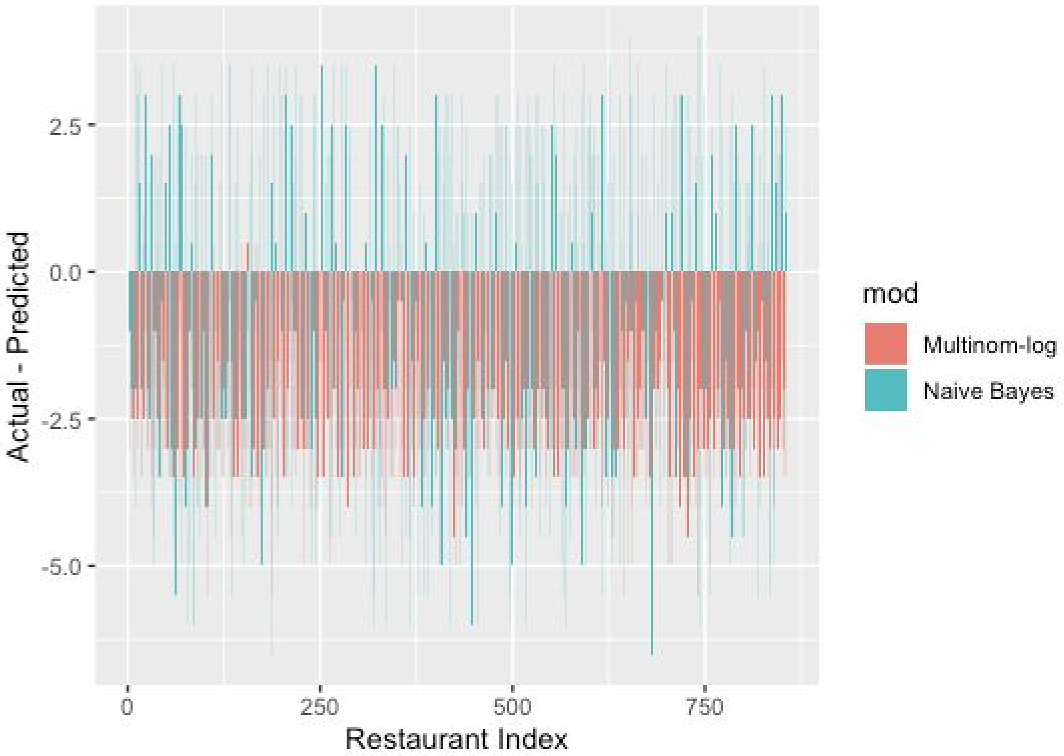


Fig. 10

The plot above displays differences between actual and predicted classes for ratings; Multinomial logistic regression leads to predicted ratings being higher than the actual ratings for all restaurants. Whereas Naive Bayes has predicted ratings both higher and lower than actual ratings. This explains the highly negative bias for Multinomial logistic classifier and and a lower bias for Naive Bayes.

V. DISCUSSION AND LIMITATIONS

In this paper, two different feature generation methods were implemented- top frequent words from raw data and top frequent adjectives after parts-of-speech. Four learning models, including Multinomial Logistic Regression, Naive Bayes, Support Vector Regresion, and Support Vector Regression Normalized, were used on our chosen data.

As shown in the table (*Fig.8*), there is no significant difference in RMSE of different feature generating methods. The baseline feature selection method has higher figures of RMSE in support vector regression compared to the feature engineering II, where the RMSE of multinomial logistic regression was smaller. Some limitations of the baseline method is that it would treat conjugated words as different words where in fact they mean the same thing; No important information of data was lost in the transforming. When utilizing part-of-speech tagging, informative features and words are erased and 500 the most frequent words are selected.

We expect the RMSEs derived from feature engineering II smaller than the baseline method, however, there was not much of a change. The reason for this might be that adjectives do not actually carry more information than other types of words, or the number of adjectives was not large enough to be representative of the text.

Multinomial Logistic Regression performs the best in general (Figure 8). Multinomial logistic regression is a particular solution to classification problems that use a linear combination of the observed features, hence we inferred that the features and business stars probably were linearly correlated. The classes for the response variable in our data set are skewed to the right; with more people giving reviews greater than three. This implies that are classes are dataset is not balanced; i.e. classes are not represented equally. This may impact the bias in our models. We also found that normalization of features is helpful for Support Vector Regression model.

VI. Conclusion

An abundance of diversity and quality of reviews is embedded in our review texts, which allows us to firsthand explore what a costumer's real unbiased opinion is. Two feature engineering methods and three Machine Learning Algorithm, including Multinomial Logistic Regression, Naive Bayes and Support Vector Regression, were implemented. The final results demonstrated that Multinomial Logistic Regression with top frequent words extracted from raw data had the best performance. Due to time limitations, we focused our analysis on restaurants. Thus, generalizing our model by considering all business categories, such as beauty, theaters and haircuts, would be something we want to do in the future once we get better results on Yelp restaurant reviews.

For future work, we plan to further enhance our feature extractor and incorporate high-level features such as word dependencies incorporating word vectors or semantic parsing. One method is using a polarity score value between -1 and 1 on word or sentence level in the following transformation: ((1 - (1/(1 + exp(polarity)))) * 2) - 1.

Text understanding is still actively researched and we plan on using new technologies and methods to get better results on detecting the derived bias so that a business can receive a fairer rating so that customers can have a clearer expectation of it.

VII. References

M-clark.github.io. (2019). *An Introduction to Text Processing and Analysis with R*. [online] Available at: https://m-clark.github.io/text-analysis-with-R/part-of-speech-tagging.html [Accessed 9 Dec. 2019].

GitHub. (2019). *trinker/sentimentr*. [online] Available at: https://github.com/trinker/sentimentr [Accessed 11 Dec. 2019].

Yu, M., Xue, M. and Ouyang, W. (2015). Restaurants Review Star Prediction for Yelp Dataset.

Fan, Mingming & Khademi, Maryam. (2014). Predicting a Business Star in Yelp from Its Reviews Text Alone.

Rpubs.com. (2019). *RPubs - Naive Bayes Classification for Sentiment Analysis of Movie Reviews*. [online] Available at: https://rpubs.com/cen0te/naivebayes-sentimentpolarity [Accessed 11 Dec. 2019].

Yan, Q., Ji, J. and Li, H. (2019). [online] Cs229.stanford.edu. Available at: http://cs229.stanford.edu/proj2015/334_report.pdf [Accessed 02 Dec. 2019].