Yelp Review Analysis for Burger Business

Jiawei Huang, Yiqun Xiao, Shushu Zhang

November 28, 2020

1 Introduction

Yelp, a platforms for customers to write reviews and ratings of the businesses, published a subset of their data set, containing four JSON files about the reviews and the information of businesses and users. In this report, we intend to provide useful, actionable suggestions to business owners in order to improve their ratings based on the information contained in reviews, and the information of the businesses. Owing to the enormous differences among different categories of businesses, we mainly provides suggestions to "burger businesses" in this report. The rest of the report is organized as follows. We introduce the details of data analysis, including preprocessing and model building od the data, in section 2. We provide overall analytical insights and data-driven action plan based on the aforementioned data analysis in Section 4. We state the strengths and weaknesses of the model in Section 5, and our contributions in Section 6.

2 Statistical Analysis

In this section, we provide detailed narratives of the process of data analysis. We first clean and extract useful information for the text data in Section 2.1, and then use lasso to reduce the high dimension of the text data in Section 2.3, and use regression to fit proper model in Section 2.4.

2.1 Data Cleaning

In order to extract useful information from the reviews, we need to preprocess the text data. We (1) keep those whose "categories" contains "burger"; (2) merge "review.json" and "business.json" by "business_id" in order to reflects the information of reviews on

the properties of their corresponding businesses. For example, if we find that the word "parking" in the review is of great importance to the ratings, and the business has no parking spot, then we suggest the bussiness owner to provide parking spots; (3) remove all the punctuations; (4) tokenize negative words, such as "no", "not" into the adjacent words to convey the real feelings of the reviewers, such as converting "not clean" to a single word "no_clean"; (5) remove stop words; (6) lemmatize; (7) count frequency of each words; and (8) remove all the words occurring only once or twice, since they are mostly typos or nonsense.

2.2 TF-IDF (Term Frequency-Inverse Document Frequency)

We first obtain a sparse matrix based on the frequencies of the occurance of a word for each text review, where rows for the text reviews and columns for words that occurs three or more times in all reviews. We have 65313 reviews in total for all "burger" restaurants, and 21848 words that occurs more than twice, resulting a sparse matrix of 65313*21848 dimension. In this case, however, if a word occurs more often than others in general, it is more likely to occur more frequently in a review, such as "burger" in our case, which is unfair to other words. Therefore, we use TF-IDF (Term Frequency-Inverse Document Frequency) to offset these effects. TF-IDF can be mathematically characterized by

$$\begin{aligned} TF(\omega, t) &= \frac{\#\omega \text{ in t}}{\#\text{words in t}} \\ IDF(\omega, t) &= log(\frac{\#\text{words in t}}{\#\text{reviews that contains } \omega}) \\ TFIDF(\omega, t) &= TF(\omega, t) * IDF(\omega, t) \end{aligned} \tag{1}$$

where ω is a word and t represents a review.

2.3 Lasso

After taking the TF-IDF of the original matrix, we have a scaled sparse 65313*21848 matrix. Although $n > p$ which can not be considered as high-dimensional regime, we still need to reduce the dimension of the variables (i.e., words). In a penalized regression problem such as the lasso, the goal is to identify a set of variables that will have nonzero weight in the model. In this case, we estimate the model parameters $\hat{\beta}$ and then define the selection set $\hat{S}$ as follows:

$$\begin{aligned} \hat{\beta}^{\lambda} &= argmin_{\beta \in \mathbb{R}^p} ||y - X\beta||_2^2 + \lambda ||\beta||_1 \\ \hat{S}^{\lambda} &= k : \hat{\beta}_k^{\lambda} \neq 0 \end{aligned} \tag{2}$$

where $X \in \mathbb{R}^{n*p}$ is the design matrix, $y \in \mathbb{R}^n$ is a vector of outputs, and λ is a regularization parameter that controls the size of the selection set.

There are several existing methods for model selection, such as cross-validation, AIC/BIC scores, hypothesis testing, knockoffs, and stability methods. In our context, we use *stability selection*[1], whose goal is to provide an algorithm for performing model selection in a structure learning problem while controlling the number of false discoveries. Based on Lasso, we use Algorithm 1 to seven times using seven equally spaced thresholds (i.e., 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9), take the intersect of the return variable set $\hat{S}^{stable}$, resulting in selecting 765 words out of 21848.

Algorithm 1: Stability Selection Algorithm

Require: data set $Z_1, \ldots, Z_n$.

1: Define a candidate set of regularization parameters Λ and a subsample number N.

2: For each value of $\lambda \in \Lambda$, do:
(a) Start with the full dataset $Z_{full} = Z_1, \ldots, Z_n$
(b) For each i in $1, \ldots, N$, do:
i. Subsample from Z_{full} without replacement to generate a smaller dataset of size $\lfloor n/2 \rfloor$, given by Z_i.
ii. Run the selection algorithm on dataset Z_i with parameter λ to obtain a selection set $\hat{S}^{\lambda}_{(i)}$.
(c) Given the selection sets from each subsample, calculate the empirical selection probability for each model component: $\hat{\Pi}^{\lambda}_k = P(k \in \hat{S}^{\lambda}) = \frac{\sum_{i=1}^{N} I(k \in \hat{S}^{\lambda}_{(i)})}{N}$ The selection probability for component k is its probability of being selected by the algorithm.

3: Given the selection probabilities for each component and for each value of λ, construct the stable set according to the following definition:
$\hat{S}^{stable} = k : \max_{\lambda \in \Lambda \hat{\Pi}^{\lambda}_k \geq \pi_{thr}}$
where π_{thr} is a predefined threshold varied from 0.6 to 0.9.

4: **return** $\hat{S}^{stable}$

2.4 Model Fitting

After stability selection, we fit regression model with review scores to be response, and the aforementioned 765 words to be predictors. The results of top ten variables based on p value is summarized in table 1.

Table 1: Regression Summary (Top 10 variables)

words	occurence	pval	tscore
great	29060	0	47.4054425559631
best	10642	2.75845242066736e-243	33.4542380847868
worst	2238	2.90066315434987e-215	-31.4349060901696
amazing	6170	3.22304171253052e-190	29.5173425939315
delicious	9390	9.48495435969436e-166	27.5191839459281
terrible	1990	7.16046601520904e-143	-25.5136751342918
awesome	4776	4.47266428430108e-130	24.3217684663504
excellent	4329	8.93741984766674e-122	23.5170115699077
horrible	1539	6.56706045138237e-119	-23.2327980044903
good	39469	7.49159746934729e-105	21.7863536884955

3 Diagnostics

4 Insights & Action Plan

5 Strengths and Weaknesses

6 Contributions

JH: Data Cleaning of R code, Section 4 of the Summary,
YQ: Model Diagnostics of R code, Section 3&Section 4 of the Summary,
SZ: Model building of R code, Section 1& Section 2 of the Summary,

References

[1] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.