STAT 628 Module 3 Summary

Group members: Yicen Liu, Hua Tong, Enze Wang

Introduction and Background

Yelp is an Internet company which provides a platform for users to post reviews of businesses. Nowadays, customer reviews play a pivotal role in doing business. For this project, we will analyze more than 67000 reviews for more than 1500 **pizza** business on Yelp and provide three kinds of helpful suggestions to business owners in order to improve both their business and reviews star ratings on Yelp. Welcome to our main code on Github: `https://github.com/sai-liu/628Yelp` and Shiny app: `https://ewang36.shinyapps.io/m_3_app/` for any detail.

Data Cleaning

The original data are four JSON files, so we first write code to transform them into CSV files by unfolding integrated columns. We focus on business from four cities: Madison, Cleveland, Pittsburgh and Urbana-Champaign. Therefore, we remove the business which are too far away from the center of four cities (more than 55 km). Then we select pizza businesses which are open, and remove some useless columns. For all remaining attributes, we check their values and do some transformation to make sure that each of them has appropriate factor levels for statistical analysis. For all reviews connected with pizza business, we detect some unreadable reviews (not English or meaningless expressions) and remove them. Therefore we get about 67100 reviews and 1560 pizza business for our data.

Exploratory data analysis:

Before doing complicated modeling and tests, we explore the data and make some straightforward graphs to get familiar with the datasets. We generate histogram to find out the distribution of stars. It seems that for all four cities, the star distribution is polarized in 1, 4 and 5 stars. We also plot the occurrence in reviews star rating for positive, negative and some foods words. It seems that for most positive words, they are more frequent in 4 or 5 stars reviews such as ”delicious” and for negative words, they are more frequent in 1 and 2 stars such as ”rude”.

After EDA, we decide to give suggestion for pizza business in three aspects: **Business suggestion:** which business attribute are highly related to pizza business star rating, such as offering alcohol or not, what should business do? **Menu suggestion:** how to expand pizza menu, which food are highly related to review star rating and whether add or remove it? **Review suggestion:** For every business, what food or service are customers mainly complaint or praise? How they mention in review and what can business do next?

Word Importance Calculation and Word Selection

For materials for menu and review suggestion, we first rank each word by two different methods, TF-IDF and regression method. TF-IDF is a popular method for text mining. It tends to increase if a word appears more frequent but not as frequent as common word like ‘and’, ‘is’ and ‘are’. Therefore, TF-IDF is an advanced alternative for word frequency. Regression is a traditional way to deal with this problem. Here we try to fit linear and tree regression model for the occurrence of words and review star ratings. The top 3 example results of two methods are listed below:

TF-IDF			Tree Regression	
Word	Frequency	TF-IDF	Word	Importance
pizza	97915	0.014674	delicious	0.0731
good	40169	0.011126	great	0.0726
food	34945	0.010568	not	0.0660

Most of nouns concentrate in the top of TF-IDF rank because most nouns occur more frequently and most of adjective concentrate in the top of tree regression rank because they are highly connected with review star rating. Therefore from TF-IDF rank we select by hand **Important Noun** list including "food", "pizza" and "delivery" which we can give suggestion in review suggestion part and **Food Noun** including "pasta", "beer" and "chicken" which we can give suggestion in menu suggestion part. From tree regression rank we select **Positive Word** including 200 words such as "good", "delicious" and **Negative Word** including "bad", "rude" to build our review attitude function. From the two lists we select **Important Positive Word** including 50 words such as "delicious", "affordable" and **Important Negative Word** including "rude", "expensive" which we can raise suggestion because it is difficult to give suggestion for broad word such as "good" and "bad". Then we can do analysis for our three suggestion part.

Menu Suggestion Analysis - Linear and Lasso Regression

From the **Food Noun**, we generate a word matrix which rows are each review and columns are star rating and food nouns, the i th row and j th columns is the frequency of j th food nouns in i th reviews. In this way, it is straightforward that we could do regression to find the connection between occurrences of words in reviews and review star ratings. Here we try both linear and lasso regression models. Since the word in model is few, the linear regression assumptions do not meet well here and the R^2 is quite low because there are not too many words, see part of coefficients:

Variable	Coefficient	Std.error	T-value	$Pr(>\|T\|)$
beer	0.0543	0.0107	5.069	4.01e-07
wine	0.0956	0.0162	5.886	3.98e-09
potato	-0.0109	0.0279	-0.390	0.6962
Multiple R^2:	0.02058		Adjusted R^2:	0.01877

All our selected food words explain about 0.02 of variation of review star rating. With 1 more times occurrence of beer, for example, the mean review star rating will increase 0.05. Next we fit the lasso regression model. It is shown that there is no significant difference with linear on sign of coefficients but significant variables. Considering that our regression problem is on a huge sparse matrix, we decide to use the lasso results. We will give suggestion based on positive and negative of coefficients. For example, "beer" and "wine" have positive regression coefficients, which means that the more frequent occurrence of "beer" and "wine" in a review may increase mean review star rating. This result coincides with our later analysis of alcohol offer.

Review Suggestion Analysis - Review Analysis, ANOVA and T Test

If we are going to raise suggestions for business from reviews, we have to tell whether reviews are positive or negative. What's more, if someone writes "the pizza delivery is delayed", we should suggest business to make delivery faster. To meet the goal, we develop review attitude function and review split process.

Review Attitude Function: From **Positive Word** and **Negative Word** list build before, for each review, we try count the number of words from the two list. Scanning each word, if it is from positive list with no negative adverb ('no', 'not') in front of it, we will count positive +1; if there is negative adverb in front of it, we will count negative +1. Similar with negative words. In this way, we could judge a review as **positive (+1)** if positive element is more than negative, as **negative (-1)** if negative more and as **neutral (0)** if they are equal. Because many reviews intersect with neither positive nor negative word list, so we have a lot of neutral reviews. The threshold is not so strict.

Review Split: It is never a good idea to judge a whole review because we have no idea what food or service review attitude towards. After we clean review by use lowercase and lemmatization. For every clean review and every target word such as "cheese". We first cut a whole review into pieces by ".", "?", "!", etc, who usually cut the meaning of a whole sentence. For any small review contains the target word "cheese", we merge them as a new review. Then we use the output of review attitude function with this new review as the attitude of the initial review towards target word. Therefore, we can generate a matrix whose rows are all our reviews, and columns are all words in **Important Noun**, the (i, j) entry +1, -1, 0 is positive, negative or neutral attitude of i th reviews towards j th nouns, and nan if review does not mention this word.

Therefore, for every word in **Important Noun**, we extract review star rating of its positive, neutral and negative reviews. By ANOVA, we found that there are significant differences among the mean review star rating of three kinds of reviews of almost all nouns. Then by T test with H1: the mean review star rating of positive reviews is larger than negative, we find 110 food and service whose review star rating of positive is significant larger than negative at $\alpha = 0.05$ level, the following is part of T test results. For those nouns, if business has more positive reviews they will have higher review star rating. Therefore the problem how to increase review star rating is transferred to how to increase positive reviews and avoid negative reviews.

Word	Mean rating of positive reviews	Mean rating of negative reviews	T-test p-value
pizza	4.1	2.4	<2e-16
food	3.9	2.0	<2e-16
cheese	3.9	2.8	<2e-16

Business Suggestion Analysis - ANOVA and T Test

For more than 10 business attributes from business.json. Which of them such as offering alcohol or not are highly related with business star rating? To answer this question, we conduct ANOVA and T tests for mean business star rating of different attributes option with H0: mean business star rating of different options is the same, H1: they are not the same. Then we use ANOVA or T test to find whether difference of business star rating of option in attributes is significant and calculate their mean difference to find high rating option. We found 9 significant business attributes and their high and low rating option and following is part of results:

Attribute	High rating option	Low rating option	Rating difference	T-test p-value
Alcohol	Offer alcohol	Do not offer alcohol	0.18	4.083e-05
Reservation	Offer reservation	Do not offer reservation	0.21	6.796e-06

Data-Driven Business Plan and Suggestion on Shiny Application

Now we can give suggestion based on both business reality and analysis above.

Business Suggestion: For all 9 business attributes, we propose that restaurants should keep high rating option and make it better if they already have them. Otherwise, we suggest that business owners should try to add high rating option. For example, if one business offers alcohol, we will suggest keeping it and offer better alcohol service, or we will suggest that they should offer alcohol if possible. For every 1560 businesses and all 9 business attributes we give suggestion by this way.

Menu Suggestion: By coefficients of Lasso regression, we can divide all 52 significant foods into two groups: positively correlated with the ratings or negatively correlated with the ratings. For every 1560 business and every significant food noun, we scan their review to find whether food occur or not, then give four sets of suggestions:

1. If a kind of food is positively correlated with the ratings and not occur in one business reviews, we will suggest that the business should **add** it;

2. If a kind of food is positively correlated with the ratings and occur in one business reviews, we will suggest that the business should **keep** it;

3. If a kind of food is negatively correlated with the ratings and occur in one business reviews, we will suggest that the business should try to **fix** it;

4. If a kind of food is negatively correlated with the ratings and not occur in one business reviews, we will suggest that the business should try to **avoid** it.

For example, our previous analysis shows that occurrence of beer has positive effect on the ratings. Then if we detect that the word 'beer' is mentioned in the reviews of a restaurant, we would say that they should keep offering it and make it better. However, there are many reasons a food noun occur in reviews so we suggest business think twice before adding or removing them.

Review Suggestion: For every 110 food and service noun found in review suggestion analysis, we count the top 3 important positive words and negative words by list we build before from all reviews, the following is results for "delivery":

Word	Attitude	Top1	Count	Top2	Count	Top3	Count
delivery	positive	delicious	73	free	55	fresh	55
delivery	negative	rude	68	lazy	21	unprofessional	8

Therefore, for "delivery", we can give suggestion such as "It is a good idea to perfect your delivery food to be more delicious, fresh and economical to win more positive reviews." for **positive** sides and "Customers don't like unpolite, slow and untrained delivery, please offer your delivery faster and mannerly" for **negative** sides. We edit and save all our 220 suggestions in table. For every business and every 110 target words, if we find that there are more than 5 reviews mention the target noun, we will count the number and give visualization of positive, neutral and negative reviews for them, such as "There are total 9 reviews mention delivery, 2 of them are positive, 6 of them are neutral, 1 of them are negative.", along with keyword suggestion on **positive** sides, if they have negative reviews, we will also give suggestion on **negative** sides. For every 1560 businesses their reviews count are customized but their food and service suggestion are shared, because as you can see, there are totally 73 "delicious" in all 67000 reviews and most of reviews don't contains "delicious", so it is difficult to customize this part for every business for different top 3 important positive or negative words.

Summary, Advantages and Disadvantages

In this project, we apply Python and R to extract useful information from the business and reviews data, and then try to give appropriate suggestions to business owners in order to help them improve their ratings by regression, ANOVA and T test. Most of our suggestion is customized and our Shiny application and suggestion are constructive and informative. On the other hand, our review attitude function is not perfect. It is also not a perfect way to use linear and lasso regression on word count matrix. In the future, we can use more advanced method to build our model and further study reviews to make more detailed and informative customized suggestion in review suggestion part.

Contributions

Yicen Liu:

- GitHub and Code: Editor of word importance by regression, debug other code.
- Summary: Fix report.
- Shiny Application: Fix problem.
- Presentation: Main editor of presentation mp4. Speak in data clean and exploratory data analysis part.

Hua Tong:

- GitHub and Code: Editor of business suggestion, debug other code.
- Summary: Main editor of whole report.
- Shiny Application: Fix problem.
- Presentation: Speak in Shiny application part.

Enze Wang:

- GitHub and Code: Main editor of data cleaning, EDA, key words selection, review function, analysis and suggestion, menu suggestion code.
- Summary: Editor of review function and split part.
- Shiny Application: Main editor of Shiny application, maintain and responsible.
- Presentation: Main editor of presentation pptx file. Speak in review and menu suggestion part.