628 Module 3 Summary

1. Introduction

In this project, we used the merchant information dataset from the Yelp platform for statistical analysis. We selected restaurants that provide breakfast as the research object. We explored all the reviews of these businesses to calculate the Food_stars, Service_stasr and Price_stars. Based on the three stars, we can give recommendations for the restaurant in the aspect of food, service and price. In addition, we will use ShinyApp to visualize the analysis outcomes, making it more convenient for merchants and users with non-quantitative backgrounds to use.

2. Data Pre-Processing and EDA

We used three datasets in our project: business.json, review.json and user.json.

Find Breakfast_Restaurant from all Businesses:

Firstly, we want to find the information of all Breakfast_Restaurants. We search the business whose “categories” contains “breakfast” as Breakfast_Restaurant. Finally, there are 6265 Breakfast_Restaurants found.

Find Breakfast_Review from all Reviews:

Based on the business_id from Breakfast_Restaurant, we can find out the corresponding reviews. The amount of reviews is about 850 thousand.

Find Important_User from all Users:

When we explore user.json, we find that there are 2 million users, while 1.6 million users have 0 fans! In our opinion, a user with 0 fans is not important and he may give a misleading review, ranking the business high or low deliberately. So, we find out 0.4 million users whose fans are larger than 0. We call them Important_User.

Find Important_Breakfast_Review from all Breakfast_Review:

After we get Important_User, we can find the corresponding Important_Breakfast_Review from Breakfast_Review based on user_id. The amount of reviews is about 410 thousand.

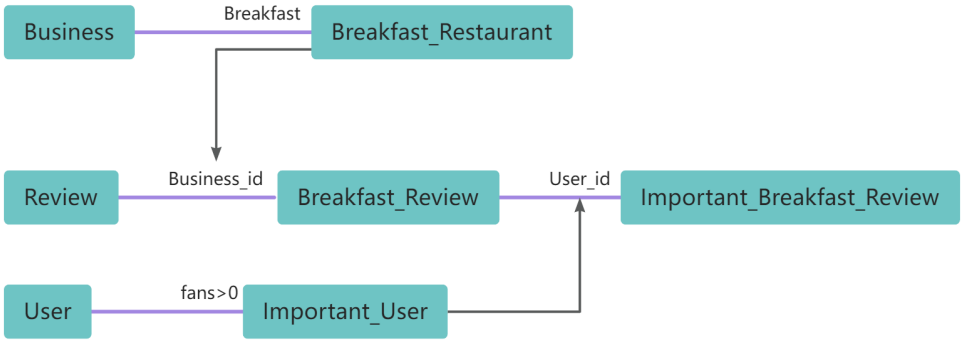

Table 1 Steps of Data Pre-Processing

Standardize the Ratings:

From the historical ratings of each user, we can see that different users may have different rating standards. For example, some users always give a restaurant 0-3 stars, and some users always give it 3-5 stars. In this case, we can not compare the ratings of different users, since 3 stars mean the highest rating of one user and the lowest rating of another user.

To solve this problem, we plan to standardize the "stars" so that the scoring standards of different users are unified. Specifically, we find out Important_Review from Review based on the Important_User's user_id, get the mean and variance of the user's historical ratings and then, for the stars of Important_Breakfast_Review, we scale it by

$$stars\ adjust_{user-i} = \frac{stars_{user-i} - mean_i}{sd_i}$$

In the later analysis, to be simpler, we scale the adjust_stars from $(-\infty, +\infty)$ to {0,1,2,3,4,5} with breaks (-2,-1,0,1,2), named as scaled_stars.

We can compare the original_stars count plot and the scaled_stars histogram plot of all restaurants:

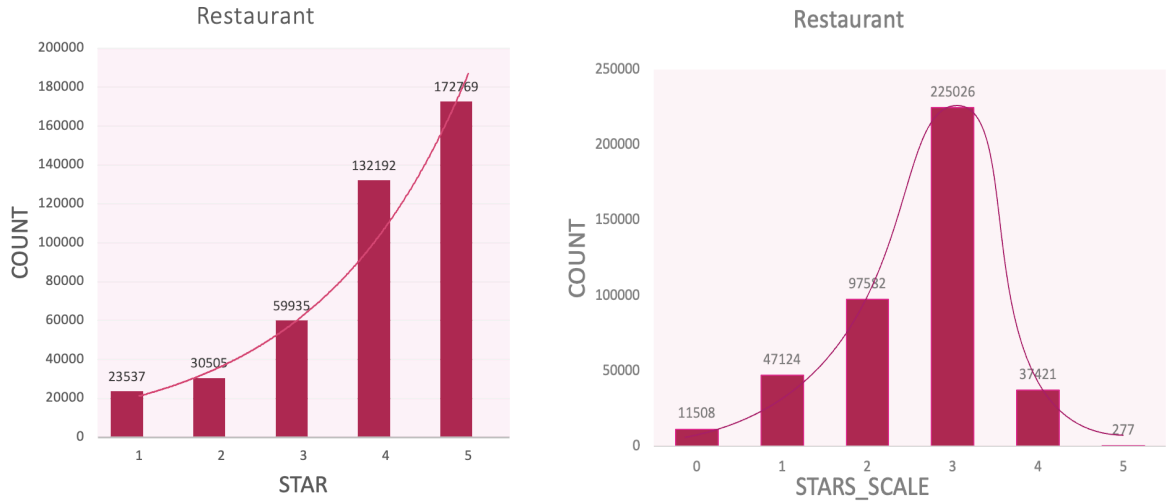

Figure 1 Comparison of Original_stars and Scaled_stars

The original_stars plot shows that people would usually give a high score. It is not credible since users may have different rating standards. While the scaled_stars plot shows that most people give a pertinent score, and more users would give a bad score than a good score.

3. Model Building: Text Analysis

Based on Important_Breakfast_Review, we use NLP to analyze the text of reviews. The goal is to divide reviews into three categories: Food_Related_Review, Service_Related_Review and Price_Related_Review by the keywords of reviews.

Firstly, we find words with high frequency in reviews using R. Then, we divide these words into three kinds of keywords: Food_keywords, Service_keywords and Price_keywords:

word categories	key words
Food_keywords	food, delicious, coffee, cheese, fresh, meal, chicken, eggs, sandwich, tasty, drinks, sauce, salad...
Service_keywords	service, friendly, staff, parking...
Price_keywords	price, expensive, cheap, affordable...

Table 3 Categories of Keywords

Next, we form a word frequency matrix via create_dtm (R's text2vec). It has 418,938 rows (review count) and 48 columns (keyword count). If review_i has keyword_j, then we let x_ij =1.

We know that each review will give stars to the restaurant, but we don't know which aspect disappointed users if the business got a negative review. So we divided reviews into three categories: Food_Related_Review, Service_Related_Review and Price_Related_Review by the keywords of reviews. For example, If a review contains food_keywords, we mark it as Food_Related_Review.

In this case, we can divide stars into Food_star, Service_star and Price_star. For a certain restaurant, we can compare the average food_star of this business with that of all restaurants. If it is lower, this restaurant should improve its food quality. Moreover, we can further analyze which aspect of food disappointed users by dividing Food_Related_Review into more kinds of reviews based on the keywords.

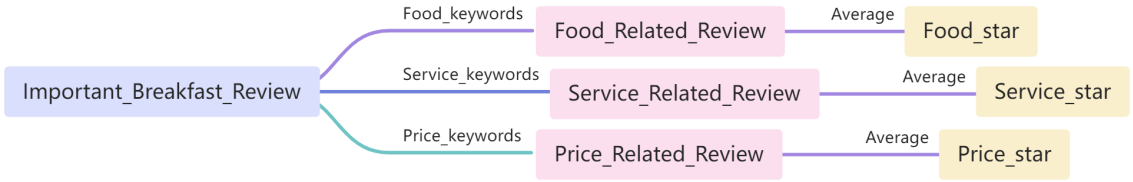

Table 4 Divide Important_Breakfast_Review

4. Recommendation for Business

We will use Taco Bell as an example to show how we come up with suggestions for a specific breakfast business. The Taco Bell locates in Carmel, IN. Its business ID is S5a0iHy8KDVm- PFnZ1Sze-g and it has 105 reviews with 1.5 stars (by business.json).

Fig.2 shows the distribution of original_stars of reviews in this business. From the analysis above, it is not credible since different users may have different rating standards. So we have Fig.3, which shows the distribution of adjust_stars. It gives us more information than Fig.2 that this business is not so bad, since there are no 0 scaled_stars and little 1 scaled_stars.

For this restaurant, the percentage of 1-2 scaled_stars takes about 60%. In order to give the owner practical suggestions, we analyze the three kinds of stars In Fig.4. The red points are the average score among all the breakfast restaurants. By comparison, Taco Bell falls short of all three aspects.

In order to provide informative advice, we looked closer into the word frequency matrix. Fig.5 shows the top food-related words in lower(1-3) scaled_star reviews. Customers complained most about cheese and chicken. Similarly, we can have suggestions on service and price. The suggestions are the following:

1. Taco Bell should care more about the quality of cheese and chicken.
2. Taco Bell should cook faster to avoid users' long waiting.
3. The staff of Taco Bell should be more friendly.
4. Taco Bell should care more about the price.

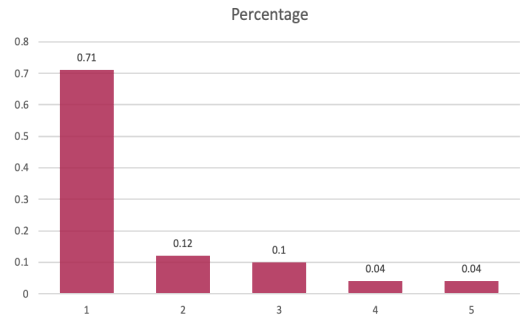

Figure 2 Original_stars count

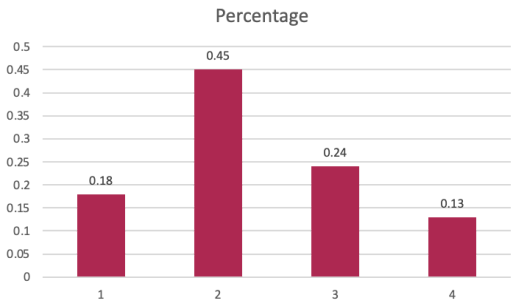

Figure 3 Scaled_stars count

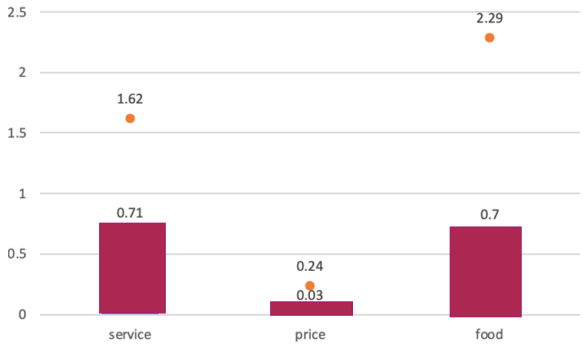

Figure 4 Stars on Three Aspects

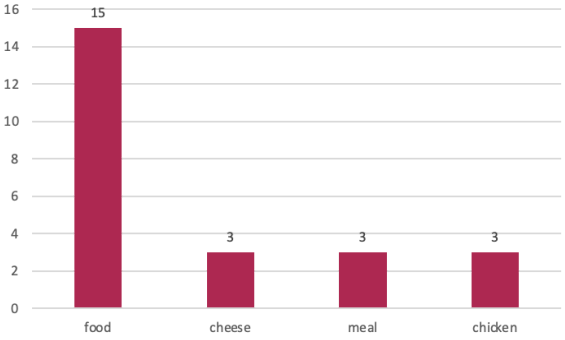

Figure 5 Further Analysis on Food

5. Conclusion

To sum up, we focus on data from the Breakfast Restaurant. We find out Important Users whose fans are larger than 0 and then filter out Important Breakfast Reviews. We standardize users' stars so that they are comparable. By using NLP, we find the keywords and divide the review into three categories: Food_Related_Review, Service_Related_Review and Price_Related_Review. We can get the corresponding stars and give suggestions to the business based on them.

6. Contribution

Name	Contribution
Xu Zou	Responsible for the coding and summary
Cuizhuo Lu	Responsible for the shinyapp and summary
Yunqing Shao	Responsible for the github repo, summary and presentation slides