1 Group Members and Contributions:
Zijun Feng: Slide 11-15
Daiyi Shi: Slide 1-10
Hanlin Tang: Slide 16 (Shiny)

2 Introduction (DS worked on this part)
Our analysis focuses on the restaurants in PA, OH, WI, IL that are categorized as **fast food**. Among these restaurants, our specific goals are to find out which advantages and disadvantages they have from the reviews and their related star ratings by customers on Yelp and then give them corresponding suggestions of improving their star ratings.

3 Data Cleaning (ZF worked on this part)
First, all the four files we download are in json format. In order to analyze them more easily, we use python to convert the four files to csv format. After we look into the category column in business_city table, we decide to analyze the fast food restaurants, which "fast food" is included in the category. It's because that we are very familiar with fast food restaurants in our daily life, so we believe the reviews are easy to understand and explain. Then we combine business_city table and review_city table together by their business id, and we get all the fast food restaurants' reviews. There are total 33262 reviews. The sample size is reasonable.
To find out the relationship between the reviews and stars, we decide to split the reviews into words and select some significant words which are meaningful. We split each review into standardized words by 7 steps:
1. Converts all the characters in the review to lowercase.
2. Tokenize the review into a list of token words.
3. Use the Stanford-standard POS Tagger in "nltk" package to get the part-of-speech for all token words.
4. Use the WordNet Lemmatizer in "nltk" package to lemmatize all token words. For example, the comparative form of verb and the past form of adjective will be reduced.
5. Check for negative words like "no" & "not" in each sentence, then label the words after negative word in this sentence as "negative".
6. Remove stop words which are nonsense and words which contain non-English characters.
7. Set up a dictionary to count the frequency for words in all reviews. Remove words with frequency lower than 30.

Now we get many word lists, each list corresponds to a review. 7 reviews that correspond to empty word lists are **removed**. These reviews may be too short or not written in English. Then we can transform these lists into an embedded word matrix (33255×3171). In this matrix, each column represents a word in the dictionary with frequency higher than 30, and each row represents a review, each cell represents the count of this word in this review. Note that if a word in a review is labeled as "negative", then we need to minus 1 instead of plus 1 when counting it.

4 Exploratory Data Analysis (HT worked on this part)
Now we can do some exploratory analysis on fast food restaurants.
We finally get 33255 reviews about these restaurant, the average score for all reviews is 2.69. The left panel in Figure 1 shows that the distribution of review scores is concave, which means people tend to give 1 or 5 compared with 2, 3, 4. The right panel shows that as the length of review grows, the score will be decrease.
In Figure 2, we can find that the positive word "excellent" is correlated with a high score; The neutral word "the" doesn't correlated with score; The negative word "bad" is correlated with a low score. The result is straightforward and easy to understand, but this gives us an insight that some words used by customers in their reviews is correlated with the scores they give to the restaurant. The problem here is "excellent" can be used to praise any aspect of the restaurant, and "bad" can be used to criticize any aspect of the restaurant, so these words are actually meaningless, and we need to find some more meaningful words. In the fourth panel, word "delicious" can be a good example. This is a positive word that is specifically used to praise the food, and we can find it is positively correlated with score. Therefore, if many reviews for a restaurant contains "delicious", it means the food in this restaurant tastes good. In reverse, if many reviews doesn't contain "delicious", it may suggest that the food tastes bad. In the fifth panel,

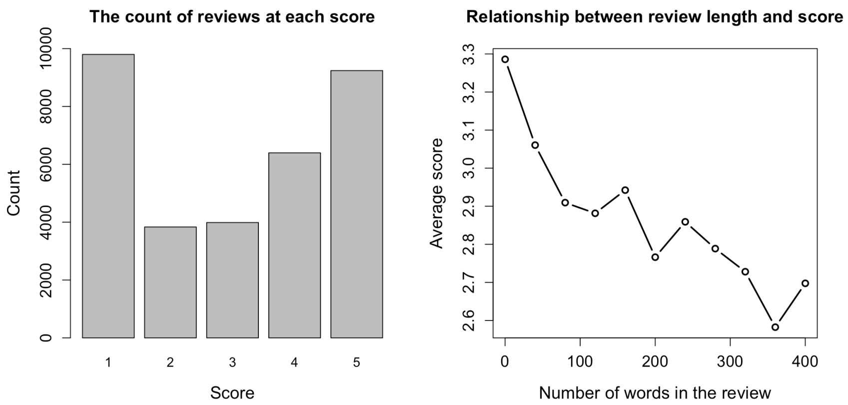

Figure 1: Summary plots for restaurant score

word "management" is actually a neutral word, but the interesting fact is it's negatively correlated with score. So we can infer that when people mentioned "management" in their review, it probably means that they are already unsatisfied with the restaurant service. The customers give scores to restaurants based on their feelings, and the reviews record their feelings in detail. The previous examples show that some words in the reviews may contain the information about why they give high/low scores to the restaurant, so there are causal relationships between these words and scores. Our purpose is to give advice to restaurant that can improve their score, so we need to find out words that can significantly effect the score and also have specific meanings.

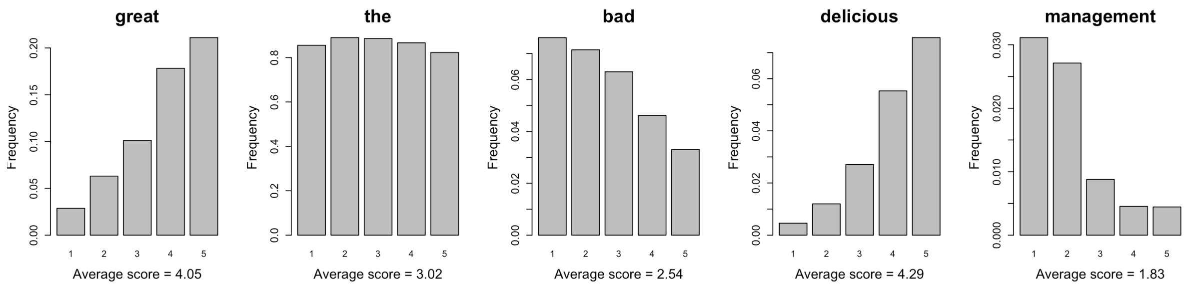

Figure 2: Frequencies of several words at each score

5 Analysis of Fast Food Restaurants (We worked on this part together)
We fit a multiple linear regression by using the embedded word matrix to find out words that affect the star ratings significantly. The explanatory variables are those words we split from reviews. The response variable is the review score. After fitting the model, we calculated the VIF values and they doesn't show serious multicollinearity problem among explanatory variable. Because the purpose of this model to help us find the significant words roughly, we don't care much about the goodness of fit and model diagnostic. Based on the model, we found out 308 significant words with $FDR < 0.10$. FDR (false discovery rate) is a common way to control the rate of type I errors when conducting multiple testing. Here we use 0.1 as the threshold, which means we expect that only 10% of significant words are actually false. Many significant words do no help to give advice, like 'excellent' and 'bad', but we still get 62 meaningful words from 308. We divide these 62 words into 6 categories, including food safety, food taste, cleanliness, service, speed and price. The word "delicious" is in food taste category. The interpretation about "delicious" from our linear model is: Controlling other words in review, having word "delicious" increases ratings by 0.36 on average. The word "management" is in service category. The interpretation about "management" from our linear model is: Controlling other words in review, having word "management" decreases ratings by 0.18 on average. Next, we design a method to combine the information of words in each category and get an evaluation score for this category. Take service as an example, there are a bunch of

words in this category, like "friendly", "polite", "rude", "management", etc. The first two words are positively correlated with review score, while the last two words are negatively correlated with review score. If we want to evaluate the service of a restaurant, we can multiple the total count of each word with the sign of its coefficient from linear model, like sign(0.18) $\times$ $n_{\text{management}}$, then add them together and divide it by total number of reviews for this restaurant. The general formula for the evaluation score is:

$$\frac{\sum_{k=1}^{m}\left(\text{sign}(\beta_k)\sum_{l=1}^{p} n_{\text{word\_}k\text{\_}l}\right)}{p}$$

where p is the number of reviews for the restaurant, $n_{\text{word\_}k\text{\_}l}$ is the total number of k-th word from the category in the l-th review, and β_k is the coefficient for the k-th word from linear model.
This formula is more general and easier to explain than directly using coefficient as the multiplier, because it doesn't depend on the exact values of coefficient from the model we fit. Continue with the previous example, We can explain the evaluation score of service as the average count (with sign) of words about service in all reviews of a restaurant.
Now we can evaluate the 6 aspects of a restaurant, but the raw value is unscaled and hard to interpret, so we calculate the percentile of the raw values in each aspect among all restaurant, and use them as scores to represent the level of the restaurant from different views.

6 Data-Driven Business Plan (We worked on this part together)

As mentioned before, we believe there are causal relationships between these words and review scores. Our purpose is to give advice to restaurant that can improve their score, so first we verify that the evaluation scores are strong predictors to the average review score.
First, we test the correlation between the evaluation score for each aspect and the average review score. Figure 3 shows the table about correlation estimates and 95% CIs for 6 aspects. We can find that 0 does not fall into any CI, so they are all significant.

	safety	taste	clean	service	speed	price
estimate	0.47	0.54	0.45	0.52	0.51	0.34
95%CI_lower	0.42	0.50	0.41	0.49	0.47	0.28
95%CI_upper	0.51	0.58	0.50	0.56	0.54	0.39

Figure 3: Correlation table

Then, we fit a new linear model based on the evaluation scores from 6 aspect to predict the average review scores of restaurants. It turns out that all 6 aspects are highly significant. The model has an R^2 of 0.72, which means these evaluation scores can explain 72% of variability for the average review score, implying a good fit. The we check the assumptions by using the Figure 4. First, we check the linearity and equal variance assumption using the first subplot. The plot doesn't show any unusual pattern, which indicates the linearity assumption is satisfied. The residuals are randomly scattered around 0 line, which means that the equal variance assumption is correct. Then, we check the normality using the second subplot. In this Q-Q plot, Although 3 points at the bottom left corner are not very close to diagonal line, most other points are approximately on the line, so there isn't a big problem. Furthermore, the third and fourth subplots show that there is no severe outlier or influential points. Finally, we also check multicollinearity by calculating VIF values. All VIF values for predictors are smaller than 2, which indicates there is no multicollinearity problem.
The previous explanation of how we get the evaluation score for each aspect shows that essence of the score is percentile. So for a specific restaurant, any aspect with evaluation score less 0.5 should be improve, because it means the restaurant is lower than average in this aspect. What's more, the aspect with lowest score should be the most urgent one, because it has the lowest percentile compared to other aspects.
We give advice for each aspect based on the words in each category.

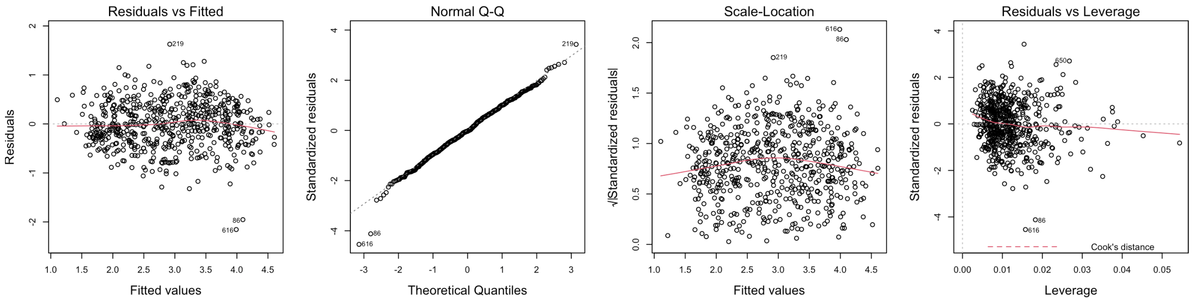

Figure 4: Correlation table

For food safety aspect, there are words about foreign matter in food, food undercooked and health. So, we advice the restaurants below average in this aspect: 1. Be careful with grub, fly, hair during cooking. 2. Don't provide undercooked food to customer. 3. Provide healthy food.
For food taste aspect, there are words about food taste, texture, and authentication and words. So, we advice the restaurants below average in this aspect: 1. Try to make the food delicious and juicy. 2. Try to get authentication for signature food.
For restaurant cleanliness aspect, there are words about cleanliness and mess. So, we advice the restaurants below average in this aspect: 1.Clean tables, walls and floor as soon as possible. 2. Organize the items in the restaurant, don't make them in mess.
For service aspect, there are words about the attitude of the staff and the management in the restaurant. So, we advice the restaurants below average in this aspect: 1. The staffs should be more friendly, patient, helpful and polite. 2. The manager should train and supervise staffs about how to serve customers.
For speed aspect, there are words about speed, efficiency and time. So, we advice the restaurants below average in this aspect: 1. Improve the efficiency and proficiency of employees. 2. Hire more people if the restaurant is too busy.
For price aspect, there are words about price and consistency. So, we advice the restaurants below average in this aspect: 1. Food may be overcharged. Reduce the price if possible 2. The product may not be consistent with your description.
We also extract 19 attributes from the bussiness_city.json file. Some attributes contains too many NAs or have unbalanced sample size, so we remove them. We transfer all remaining attributes into 2-level True/False judgements and did t-test on them. We got 6 significant attributes, which are bike parking, caters, driving through, noise, outdoor seating and good for groups. We also give advice on providing relevant service about these attributes if they currently not.

7 Strengths & Weaknesses (DS worked on this part)

The first strength of our model is that the model has a relatively stable performance and is highly explainable as the R^2 of the model is 0.72. Second, the model is very applicable and practical. Just count the number of related words in the review, we can easily get the review scores and their percentile.
For the weaknesses, first, our advice is too general. We only give advice on certain big aspects and don't consider much about the specific characteristics and conditions of each restaurant. Second, we doesn't take some significant word into account, especially some food names, such as "pizza", "vanilla", "salmon", etc. Maybe we can work out a good way to deal with those food names in the future.

8 Conclusion (We worked on this part together)

In conclusion, through the business data about fast food, we find out a bunch of meaningful words which significantly related with review stars, then divide them into 6 categories, each corresponding to an business aspect. Then, we set up an evaluation method for these aspects based on the count of words. Business owners can check the suggestion we provide in our Shiny app by selecting the brand name and address. If the restaurant performs worse than average in any aspect, then the owner should improve it to get better review star ratings from customers. We also give advice based on restaurant attributes, which are also useful.

References

[1] NLTK 3.5 documentation: WordNet Interface,
https://www.nltk.org/howto/wordnet.html

[2] Stanford Natural Language Processing Group: Software: Stanford Log-linear Part-Of-Speech Tagger:
https://nlp.stanford.edu/software/tagger.html

[3] Wikipedia: False discovery rate,
https://en.wikipedia.org/wiki/False_discovery_rate