STAT 628
Module 3: Yelp Data Analysis

1. Introduction

When someone decides to open a restaurant, the first thing that crosses their mind is the type of cuisine they will serve, the interior design of their restaurant and the services they will offer. Our analysis is focused on helping new restaurant owners in California predict their initial rating on Yelp based on features such as Food Type, Ambience, Outdoor Seating, Restaurant Reservation, Caters ,Drive-Thru, and Restaurant Takeout. We will also provide them with menu suggestions and customer complaints to help their business perform better.

2.Data Cleaning

We used the business.json and the review.json from Yelp and filtered both of them to focus on restaurants in California. For the business.json , we dropped the closed restaurants and manually categorized each restaurant into one of these categories: Italian, American, Fast Food , Deli, Bakery, Mexican, Coffee, Breakfast, Japanese, Healthy and Pizza. We categorized these restaurants into a type based on their menu or their restaurant description. Finally, we assigned attributes with missing values a value of 0.5. In other words, for each attribute we looked at three factors: 0 indicating that the restaurant does not have the attribute, 0.5 indicating the restaurant may or may not have the attribute, and 1 indicating that the restaurant has the attribute. We used the review.json to conduct a NLP sentiment analysis using Valence Aware Dictionary for Sentiment Reasoning(VADER) to be able to provide the business with recommendations.

3.Exploratory Data Analysis (EDA)

We started off by creating a barplot for each of the eleven food categories to visualize the distribution of their rating. We noticed that most of the ratings were negatively skewed and also had a median centered at around 4. To be able to make comparisons across restaurants, we decided to standardize star ratings so that a 4.5 star rating with 4 reviews isn't the same as a 4.5 stars rating with 256 reviews. We used the review.json file to calculate both the average star rating and the standard deviation for every restaurant. We then used those values along with star rating in the business.json to find the z-score for each restaurant. Based on the sign of z-score value, we either subtracted or added the margin of error $=\frac{standard\ deviation}{\sqrt{review\ count}}$ from the star rating in the business.json. The figure below shows the distribution of the stars before and after standardizing for American restaurants.

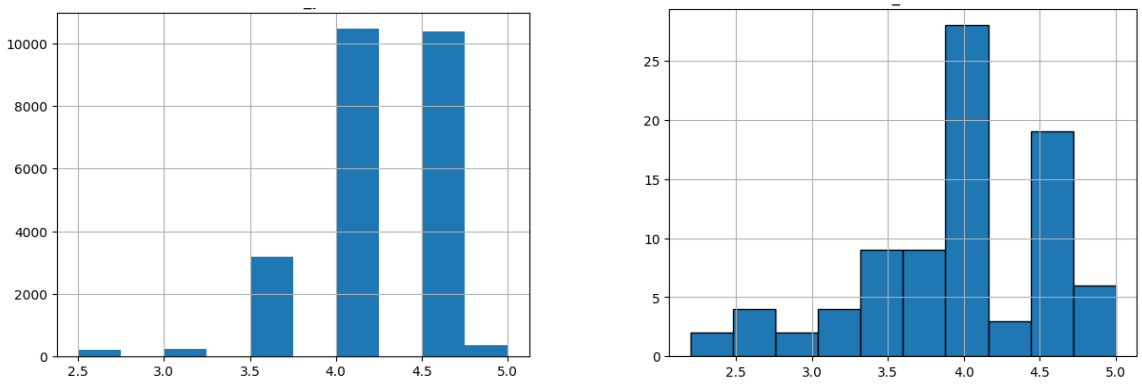

Figure 1: Distribution of Star Ratings

4.Modeling

4.1. Random Forest Model

The first task was to find out important features to help us in predicting the rating for the new business owner. For this we used a method called " Learning Vector Quantization " (LVQ). It is a type of Artificial Neural Network which is inspired by biological models of neural systems. It is based on a prototype supervised learning classification algorithm and trained its network through a competitive learning algorithm similar to Self Organizing Map. It can also deal with the multiclass classification problem. LVQ has two layers, one is the Input layer and the other one is the Output layer. The LVQ gives out the top 20 important variables out of all of the variables present in the dataset. Out of those 20 variables we chose to select only 9 variables : "OutdoorSeating", "Ambience_classy", "Ambience_casual", "Caters", "DriveThru", "RestaurantsReservations" , "Restaurants Delivery" , "Type" , and "RestaurantsTakeOut" . Furthermore, we also created a boxplot for each attribute to further study its importance on rating. Figure 2 below shows that restaurants that either had Outdoor Seating or Catering Service had a higher rating than those that did not.

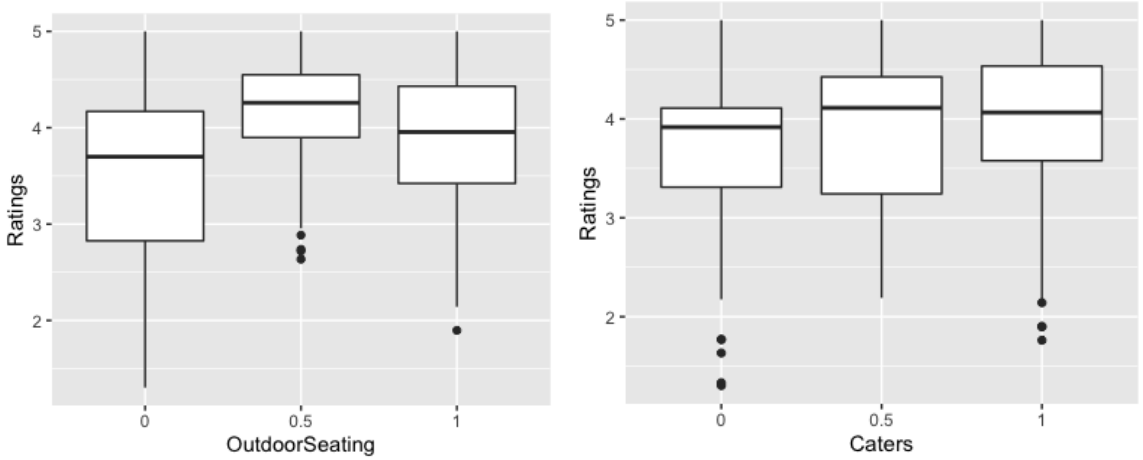

Figure 2: Boxplot for OutdoorSeating and Caters

We then began to make a model for the prediction and started with building the Random Forest Model(RF). To be able to do RF, we grouped the standardized stars into 6 factor levels where ratings below 1.75 were classified as a 1.5, ratings between 1.75 and 2.25 were classified as a 2 ,ratings between 2.25 and 2.75 were classified as a 2.5, ratings between 2.75 and 3.25 were classified as a 3,ratings between 3.25 and 3.75 were classified as a 3.5, ratings between 3.75 and 4.25 were classified as a 4,ratings between 4.25 and 4.75 were classified as a 4.5 and ratings greater than a 4.75 were classified as a 5. The Random Forest model takes in the 9 variables we selected, spits out a probability for each category and selects the one with the highest probability as the output prediction. The model when trained on the entire dataset gives an accuracy of 83%.

The figures below show the prediction in our Shiny App for 2 different scenarios. In Figure 3, when the type of restaurant being selected is Healthy and "No" is selected for all the other attributes, our model gives us a prediction of 1.5.

Prediction	X1.5	X2	X2.5	X3	X3.5	X4	X4.5	X5
1.5	0.43	0.03	0.11	0.10	0.05	0.21	0.06	0.01

Figure 3: Rating Prediction for a Healthy Restaurant

However, when we change two of our attributes, Ambience_Casual and Restaurant TakeOut, to "Yes" the prediction changes to a 4 which shows that by adding a set of attributes business owners would be able to increase their ratings.

Prediction	X1.5	X2	X2.5	X3	X3.5	X4	X4.5	X5
4	0.00	0.01	0.24	0.08	0.06	0.49	0.12	0.00

Figure 4: Rating Prediction for a Healthy Restaurant

4.2. Strength and Weakness of Model

When we tried a train-test split our model's accuracy dropped from 83% to an accuracy of about 50%. There is a major difference between the accuracy of the train-test split and the accuracy of the entire dataset because there is an imbalance in data which results in an imbalance in the test dataset everytime the RF runs. To further elaborate, our dataset only had a few rows with low ratings and all the other rows had high ratings.If all the low ratings go into the test when we do a train-test split, our model has no observations to train on resulting in a misclassification. Random Forest is hard to interpret compared to linear models but when we tried to fit a linear model with the 9 attributes the R-squared value was at 26.7%.

5. NLP Sentiment Analysis

We use NLTk, a prominent program in Python, to tokenize text, remove stop words and find high-frequency words. We filtered out common stop words with no contextual significance such as "a","the", "and", "but", etc. We then used the ngram_range parameter in the NLTk packages to

create bigrams, phrases consisting of two words, and trigrams, phrases consisting of three words. We also used the lowercase parameter which has a default value of True to convert all characters to lowercase.

6.Recommendations

To create recommendations, we used restaurants that had a 4.5 star rating or above to generate bigrams and trigrams for each restaurant type. Once we had those bigrams and trigrams, we created barplots, like the one in Figure 5,to see the most frequent words. To form a suggestion, we only looked at words that were meaningful. For instance, a word like 'italian restaurant' or 'santa barbara' would be considered irrelevant. Using the words from the barplot,a meaningful suggestion for Italian restaurants would be to have 'garlic bread' , 'olive oil' and a 'wine list'. To further check that people actually enjoyed the 'garlic bread' , 'olive oil' and 'wine list', we also looked at the trigram that contained those phrases. We saw that it was 'good garlic bread' , 'great wine list' and 'best olive oil' which means that these are menu items that Italian restaurants should perfect or try to serve to achieve a higher rating. Similarly, we also looked at restaurants that had a 3.5 star rating or below to see their most frequent phrases. Those phrases were usually negative so we used those phrases as things restaurants should try to avoid. For instance, something that Italian restaurants should be cautious about is having 'service absolutely terrible'.

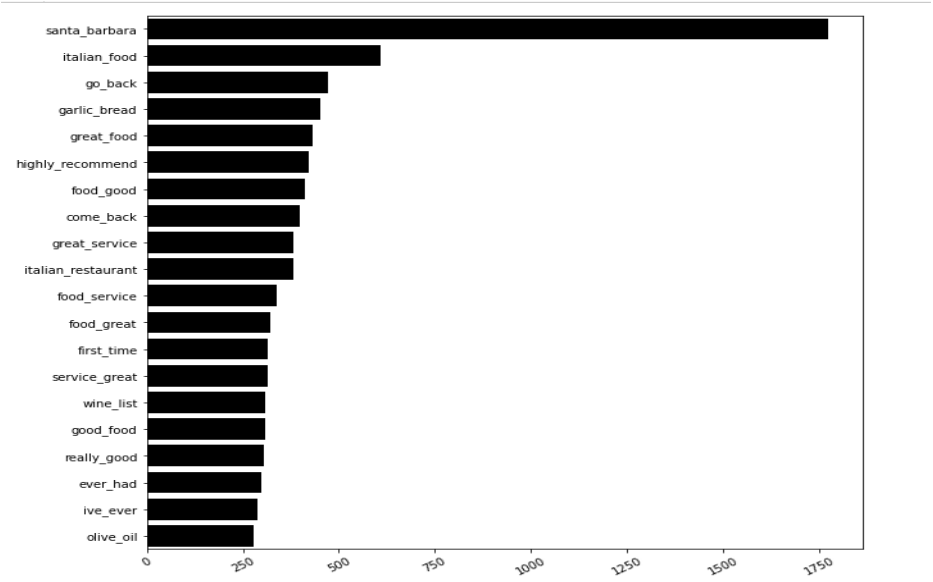

Figure 5: Bigram barplot for Italian restaurants

7.Conclusion

In our analysis, we used random forest to generate an initial Yelp rating for new restaurant owners. We also used NLP sentiment analysis to provide these new restaurant owners with menu suggestions and advice on things to avoid.

Contributions:
AA: Cleaned the hours and days of the week columns, attempted different models and created our final model, worked on modeling code for the Shiny App.
ME:Categorized the restaurants into types, calculated the total hours of operation per week,attempted a regression model, standardized the stars, and worked on the Shiny App.
SH: Read in the json files and converted them to CSV, converted the attributes into columns, performed the NLP sentiment analysis, and worked on the Shiny App.

Together: Worked on our respective parts in both the presentation and paper.