Yelp Restaurants Analysis

STA 141B Final Project by Bichen Kou, Jiaman Wu, Wenjie Xie, Yue Tian

Project Goal

In this project, we analyze restaurants in New York using data gathered from Yelp. Throughout the project, we are motivated by the question: what makes a good restaurant? This is an important question for many reasons. First, companies such as Yelp may want to improve rating systems to provide more accurate and concise information to customers. Customers, in turn, will make more informed decisions. Companies such as Yelp can also use restaurant ratings to return more relevant search results for customers. The fair rating system can also foster a competitive environment among businesses that will incentivize businesses to improve service.

The "goodness" of a restaurant is somewhat subjective and it consists of different perspectives. Therefore, in this project, we quantify the restaurant "goodnesses" by using restaurant ratings and construct a model that predicts the rating for a given restaurant. A rating is a single digit that indicates the overall quality of a restaurant. In this project, rantings take values ranging from one to five, with increments of 0.5. Ideally, a restaurant rating should contain as much information as possible on factors that are important to restaurant goers, such as price, foods quality, service, accommodations for special diets, and the overall customer satisfaction. Using this information, we construct a random forest model to estimate the rating for a restaurant.

This report is roughly divided into five parts. In part one, we discuss our data collection and cleaning process. In part two, we present the data summary. This part includes a map showing how restaurants are distributed in New York City, and a summary of user reviews. We also explore the relationship between Yelp user review postings and time of the day. In the third part, we use Natural Language Processing techniques to analyze restaurant reviews and calculate language sentiment scores for each review. Using language sentiment scores as "proxies" for customers' satisfaction, in the fourth part, we build a random forest model to assign/predict a rating for a restaurant. In the final conclusion part, we summarize and discuss results from our analysis, and reflect on where we can improve.

Data Collection

With a clear goal in mind, our first step is to collect data. There are two main components to our data set. The first component is restaurants' basic information, which we obtained from Yelp API. These basic information, however, is not enough to reflect a restaurant's overall quality. To solve this problem, we resort to web scraping. In this process, we scraped the first 20 reviews and restaurant attributes for each business.

Yelp API

Using Yelp Business Search endpoint, we were able to obtain data on 1,000 restaurants in New York City. Responses returned include a unique ID for each business, name, location, restaurant URL and the overall rating given by Yelp. We then converted the responses into a Pandas Dataframe with each row representing a restaurant.

Web Scraping

Business information that we obtained from Business Search endpoint is not enough to perform our analysis. We need to have more detailed information on each business in order to construct a model that estimates business overall rating/quality. Yelp does provide some information on business attributes in the "transaction" column, however, the information is limited. In addition, customer reviews are crucial in determining the quality of a restaurant. Although Yelp does have a Review endpoint that returns customer reviews for a given restaurant, it is problematic for our analysis. Yelp only returns up to three reviews for one restaurant, and each review is limited to 160 characters. The beginning of a review contains less information than a full review. The restricted access to reviews can lead to biased analysis. Therefore, we used web scraping technique to obtain the first 20 full reviews and attributes for restaurants using URLs from our previous data set.

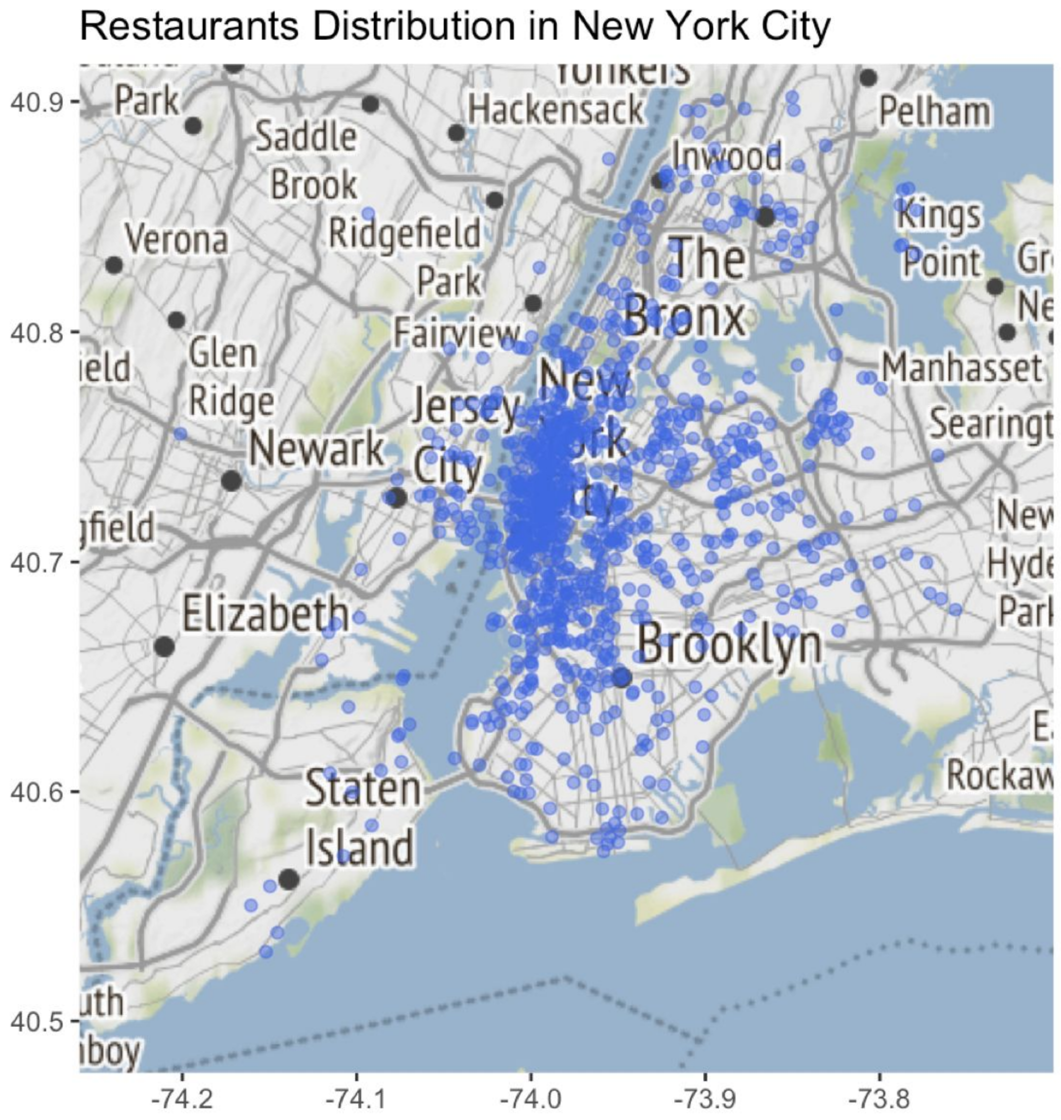
The results returned from our web scraping is in a list format, with each list element a dictionary representing one restaurant. Within each dictionary is lists that contain 20 reviews, 20 corresponding ratings associated with each review, restaurant attributes, and business ID.

Data Cleaning

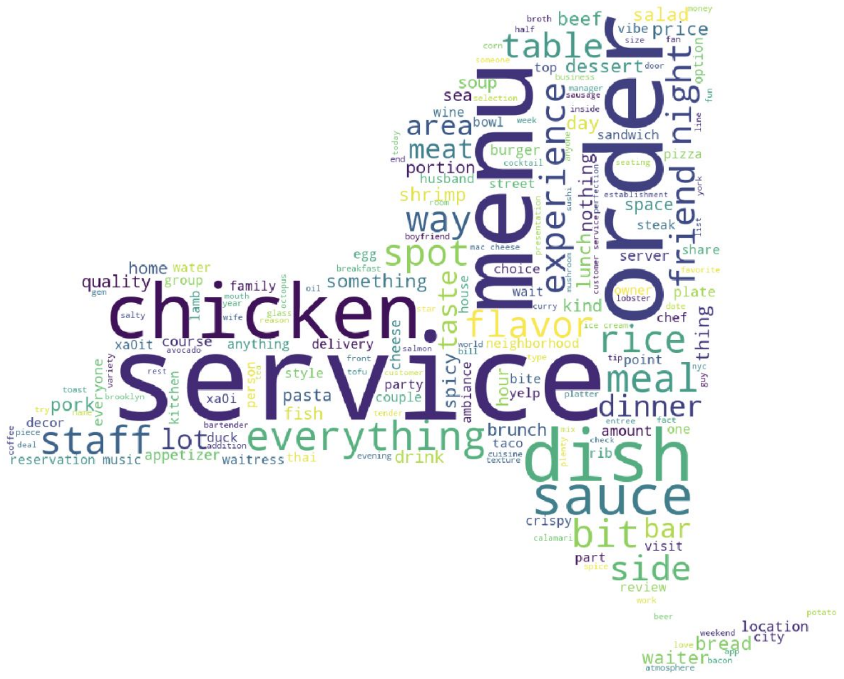
Now, we need to convert these information into a workable dataframe. We first take "attributes" from the list, then we convert "attributes" for each restaurant into a data frame, and finally, we merge all data frames together to produce a data frame with each row a restaurant and each column an restaurant attribute. We have 52 attributes in total, including “Take Reservations”, “Outdoor Seating”, “Accept credit cards”. For reviews and review ratings, we apply the join function to each column and generated a data frame with each column a variable and each row a review and review rating for a restaurant.

Data Summary

The map below shows the restaurant distribution in New York City. We see from the map that most restaurants are scatter around Manhattan and Brooklyn area.



This plot shows the most common nouns appear in customers' review in restaurants in New York City, including words such as"Service "," Manu", "Order" and "Chicken".



NLTK Language processing:

In the language processing part, we did two things. The first one is to visualize nouns that are most commonly used in Yelp reviews to figure out what factors customers care about a restaurant. The results are shown above. We selected noun because we think nouns contain more significant information about the content, like food name and specific restaurant service comparing to adjective and verb. Besides this, we are also interested in predicting a restaurant's rating score by quantify sentiment of each review. To accomplish this, we use the Sentiment Intensity Analyzer in NLTK package to give each piece of review a score, which will later be used as a variable in the random forest prediction model.

Word Frequency Detection

For this part, we apply the NLTK process on the reviews and find the top common words among the review. Specifically, we filter nouns by using POS-tagger. For the nouns, the most frequent such words appear "Service "," Manu", "Order". The most common food from the reviews in New York is "Chicken". And the most mentioned types of food among the reviews are ‘japanese“ and “American” .Then we use wordcloud package to visualize the most common nouns that appear in the New York area.

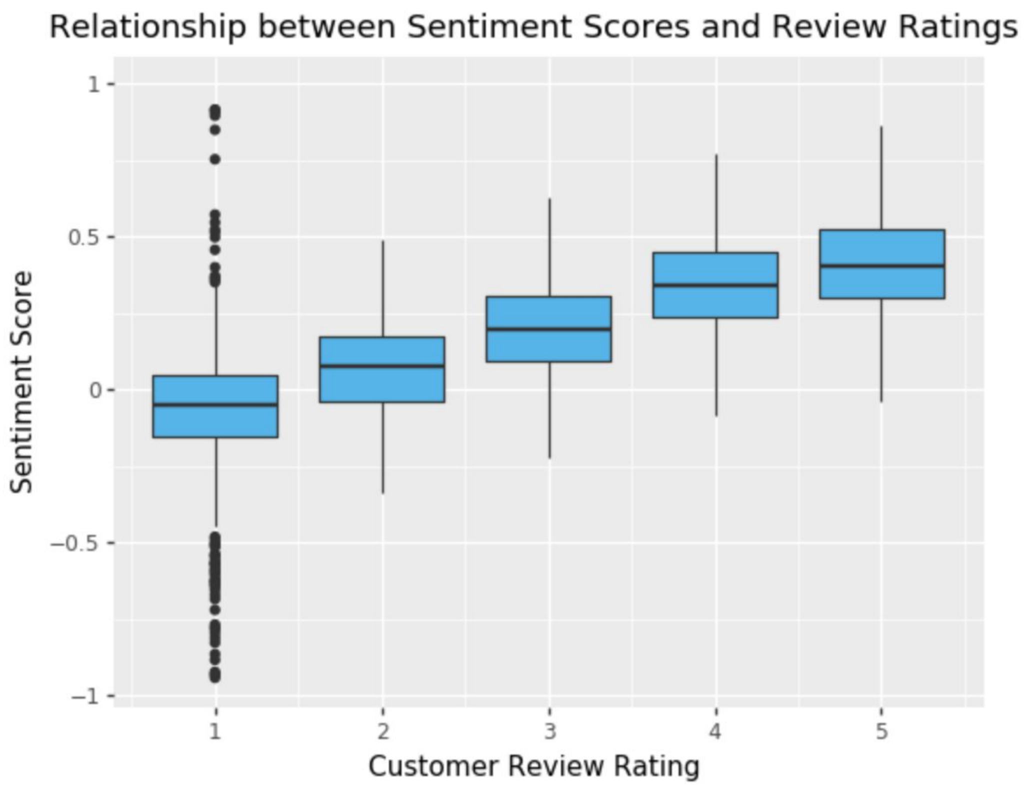
Sentiment scores

We used a simple sentiment analysis package called Vader NLTK by Python’s NLTK package. In order to get process the sentiment analysis, we first tokenize each sentences in reviews. After this, we can process the VADER (Valence Aware Dictionary and sEntiment Reasoner) on each review. The specific process is that the VADER would first run through the whole sentence and detecting the emotional and describing words such as “good”, “bad”, ”worth”. Words like “Good” is marked as a positive word, and words like “Bad” is marked as negative words and other words are defined as neutral words.

Then it would detect the “boosting words” which is usually an adverb such as “very”. Moreover, when VADER detects words like “not”, it would convert the describing words after it to the opposite category. For example “not bad” would be categorized as a positive word. After the process above, VADER would generate a number between -1 and 1 to illustrate the level of positiveness of a review; 1 as the most positive and -1 as the most negative and 0 indicates neutral. Eventually, the sentiment score indicates the positiveness of a review.

To explore the correlation between the sentiment scores and the review ratings, we generate a box plot for sentiment score and the actual ratings for each review.

Result



From the plot, there is an positive correlation between user ratings and the sentiment scores we generate from each review. Higher sentiment scores indicate more positiveness in the review content, the higher the rating the reviewer would give.

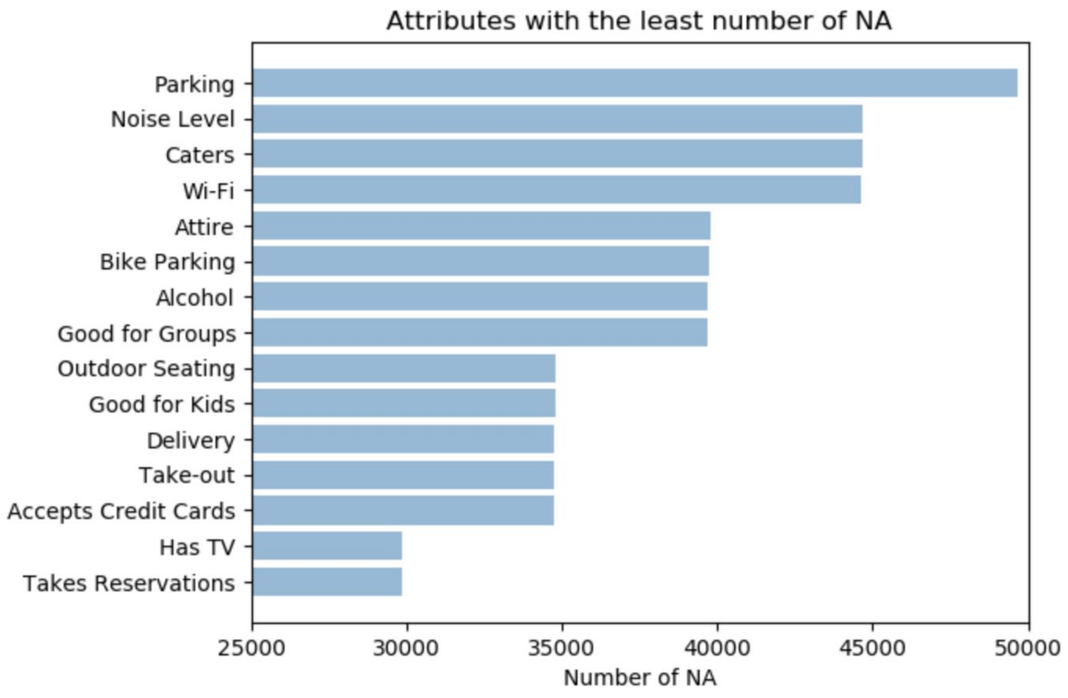
However, one thing to notice is there is a large variation in sentiment scores when a customers rates 1 (the lowest customer rating possible). In addition, the sentiment score at 0 indicates a neutral review that does not include any positive words nor negative words. However, in practice, it could be that the positive and negative scores cancel out resulting a neutral total score.

Random Forest Prediction Model

Since we have both continuous and categorical variables, we decided to use a random forest model to predict the rating of a restaurant based on the various attributes, the number of reviews, NLTK score and other attributes obtained from web scraping.

Attribute selection

Going through the attributes of restaurants, we desire the most common attributes among the restaurants to train the prediction model. There are 52 attributes in total and to find out the most common attributes, a bar plot is generated to see which attributes were mentioned most in the data we collected. From the barplot, the attributes with most NAs ("parking", "Noise Level" ..) would not be considered in our prediction model. And the attributes we used for the model is included in the bar plot below.

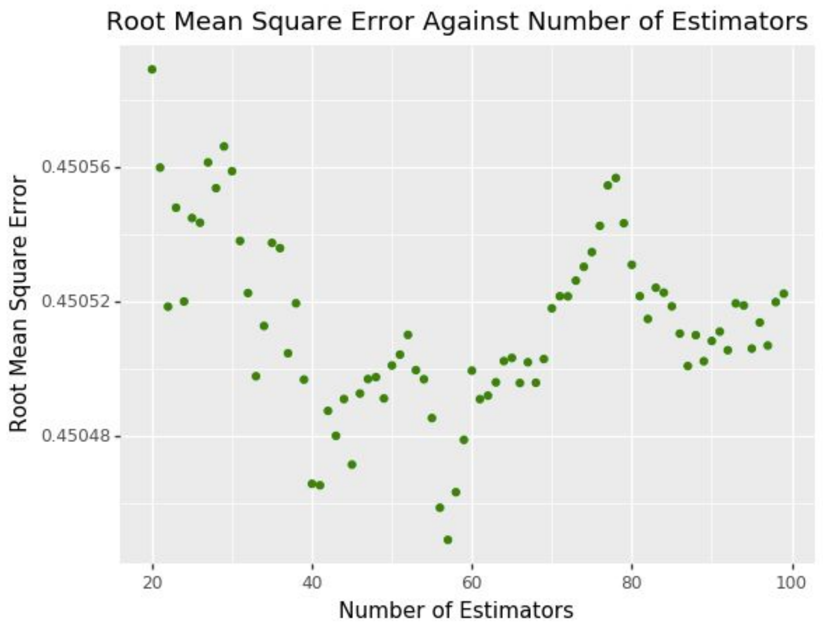


Moreover, from the 1000 restaurant we obtained, we noticed that some ratings like 2 are missing. Thus, we decided to use a random forest regression model to predict the rating of a restaurant based on the features we obtained.

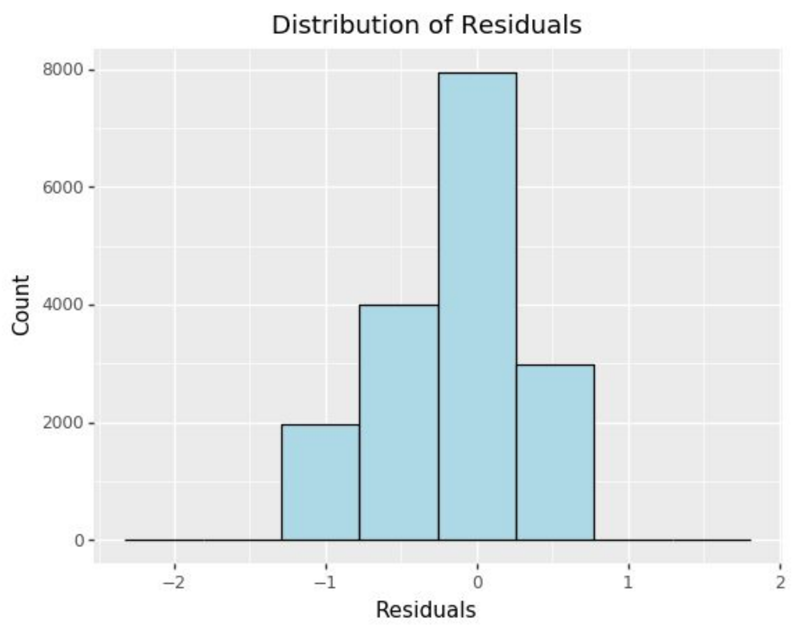
Model Selection in Random Forest

At first, we divided 1000 observations of restaurants into testing and training dataset, we used 0.2 as our test size which represents the proportion of the dataset to include in the test split. Then, we scaled our y variable: restaurant rating, and x variables: different features of the restaurant dataset.

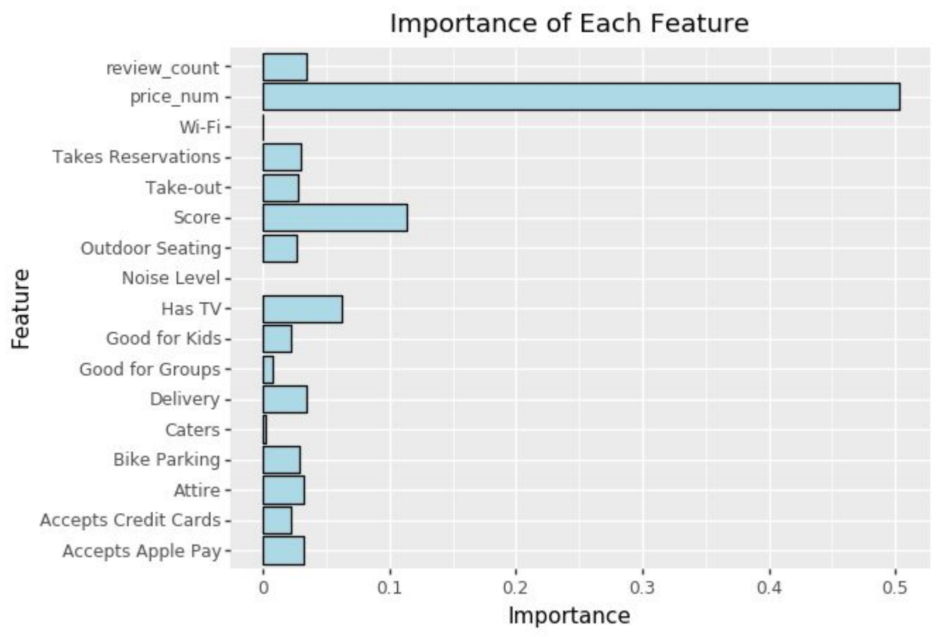
After scaling, we trained our random forest algorithm based on different number estimators ranging from 20 to 100. Estimators are defined as the number of trees in the random forest. To evaluate the performance of our algorithm, we calculated the root mean squared error(RMSE). By plotting out the RMSE values against the number of estimates, we noticed that the algorithm has the lowest RMSE values when the number of estimators is around 57.



Based on the model with number of estimators 57, the algorithm generates 57 times regression trees and then calculate the average of the outcomes. And the average of the 57 outcomes would be the most accurate predictive value for the rating of a restaurant based on the attributes and the sentiment scores we generated. Moreover, the histograms below show the distribution of the residuals, which is roughly normally distributed around 0.



As shown by the plot below, the "price_num" (number of the dollar size) is the most important feature, followed by the "score" obtained by NLTK processing, and then "has TV or not". The feature contributes the least to the model is WiFi.



Conclusion

In this project, we request 1000 restaurants' business data and top 20 reviews information for each of them in New York. For the Yelp data, we wanted to find out which attributes affect the rating of a restaurant and whether the reviews of a restaurant is correlated with its rating. Using the Natural Language process tools and the VADER model, we generate a value indicating the "positiveness" of a review and the result of comparing the value from VADER model with the actual rating for each review states a positive correlation between the review content and the review rating.

Furthermore, more attributes were tested to see if they are influential enough to affect the general rating for a restaurant. The attributes aside from the content of the reviews include attributes such as "Reservations" "accepting credit card", "take out service". After training the random forest model, we used these attributes along sentiment score to generate a predictive value for a restaurant rating. And from the predictive model, we found the price, the content of the review and having TV are three main attributes affecting the restaurant rating.

Reflection

In this section, we discuss potential problems with our analysis. First, our sample size is somewhat limited. We tried to maximize our sample size, but the company tends to protect its data. The upper limit of the restaurant data request is 1,000. However, it is a manageable size and we were able to extract useful information from it. Second, the lack of variations in our response variable, restaurant ratings, can lead to biased estimates. The lack of variations has two reasons: first, restaurant ratings only take set values, such as 2, 2.5, and 3; second, most restaurant ratings in our sample are clustered at around 3 to 4.5 and we could not find any restaurant with the 2-stars rating. The limited observations on high ratings (4, 4.5, 5) and low rating (1, 1.5) and the lack of variations inevitably cause bias in random forest model.

Another source of bias comes from the fact that we only used the top 20 reviews from our web scraping. The Yelp review sorting algorithm is unknown to us. By taking the top 20 reviews, we face the risk of bias because it could be that the reviews we have all share certain characteristics and therefore, the review data is not a random sample.

As for the sentiment scores, the score we generate for each restaurant is the mean value for all the numbers we get from the VADER process. This could be a potential problem since the value would be affected by extreme reviews. One review with strong emotions could affect the score heavily. Another interesting observation is that the number of negative sentiment scores is significantly smaller less than the positive sentiment scores. This could be the result of using only the top 20 reviews. Moreover, for the sentiment scores, we noticed that there is a certain amount of neutral reviews. One thing to improve is to specify the model so that it could detect more words as positive or negative words.

Reference:

Yelp API Tutorial: https://python.gotrained.com/yelp-fusion-api-tutorial

Simple Sentiment Analysis with Python:https://www.pingshiuanchua.com/blog/post/simple-sentiment-analysis-python?utm_campaign=News&utm_medium=Community&utm_source=DataCamp.com

Random forest model : https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/