Data & Programming Analytics

Group Project Final Report

<Yelp Review Analysis>

Group 18

Youngjoo (Jennie) Ryu

Jai Agrawal

Yaohui Wu

Xiaofan Zhu

Sahithi Muddana

Table of Contents

1. Introduction

a. Background

Yelp is an online review website that provides crowd-sourced reviews of local businesses. As a measurement of how important Yelps scores are to businesses, research shows that a Yelp rating can actually increase a business owner's sales by 5-9%.

In 2021 Q2 the revenue of Yelp reached more than 257 million dollars, but competitors such as Google are quickly increasing market share in this area. In this project, we would like to provide Yelp a way to gain a competitive advantage by providing a way to analyze the effect of reviews on businesses and discover the reasons why customers give positive or negative reviews to restaurants/businesses. With this analysis, we plan to create a tool that can offer accurate advice to business owners in order to increase their profit.

b. Project Plan

There are two techniques (three methods) we plan to employ in order to create this tool. First, we will apply topic modeling to uncover the reasons why users give positive or negative reviews to restaurants/businesses. Afterward, we will employ sentiment analysis to distinguish the "positive" comments from the "negative" comments (as well as classify neutral comments respectively). Finally, we will apply topic modeling on these "positive" and "negative" comments to extract the reasons why neutral comment is skewed to positive or negative.

2. Data Overview

a. Data Summary

There are two datasets, provided by Yelp, that we plan to analyze. The first dataset that we used is Yelp business dataset. This includes specific information about each of the companies as well as some review information such as name of the business, location, average stars etc. The second dataset that we used is the review only dataset, this dataset contains information like users comments, review rating, and usefulness etc.

b. Data Collection Method

(Reservoir Sampling Algorithm)

Since the review dataset contained more than 8.6 million data points, we had huge difficulty handling such a huge volume of data. Therefore, in order to load data locally (or store within Google Colab) we needed to use sampling techniques to load the datasets. The sampling method that we used was the Reservoir Sampling Algorithm. This method ensures that each data point has the same probability to be selected eliminating any chance of bias (we set our sample dataset size as 100,000).

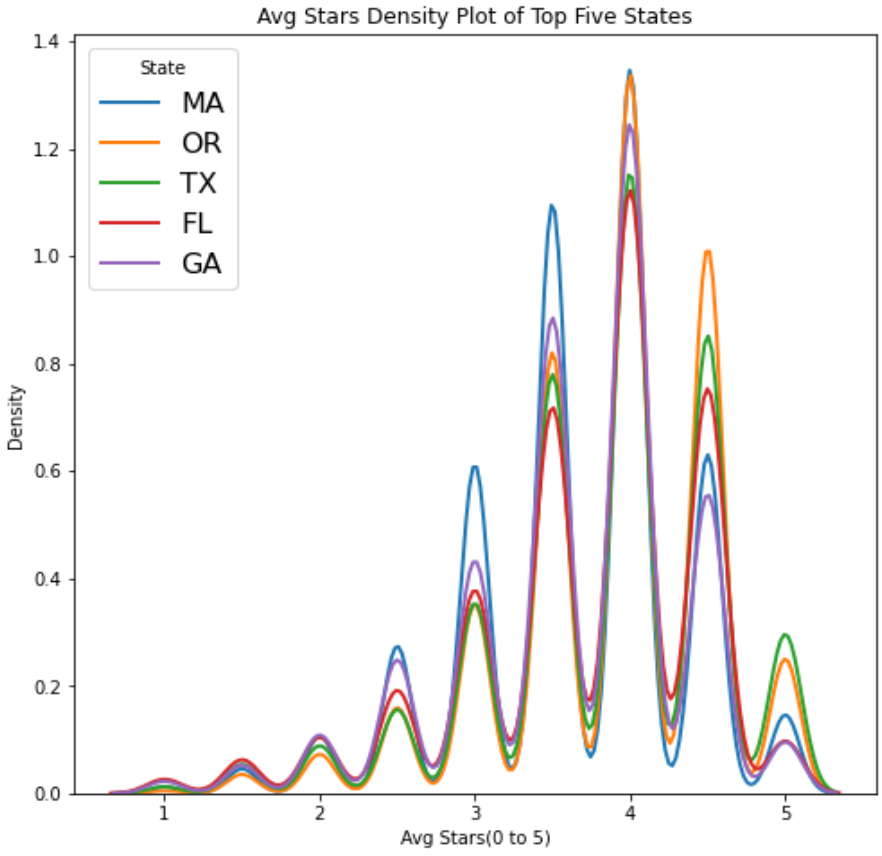
```
#reservoir sampling algorithm code
def reservoir_sampling(sampled_num, total_num):
        pool = []
        for i in range(0, total_num):#i is from 0 to 100000-1
                if i < sampled_num:
                        pool.append(i)
                else:
                        r = random.randint(0, i)
                        if r < sampled_num:
                                pool[r] = i
        return pool
```

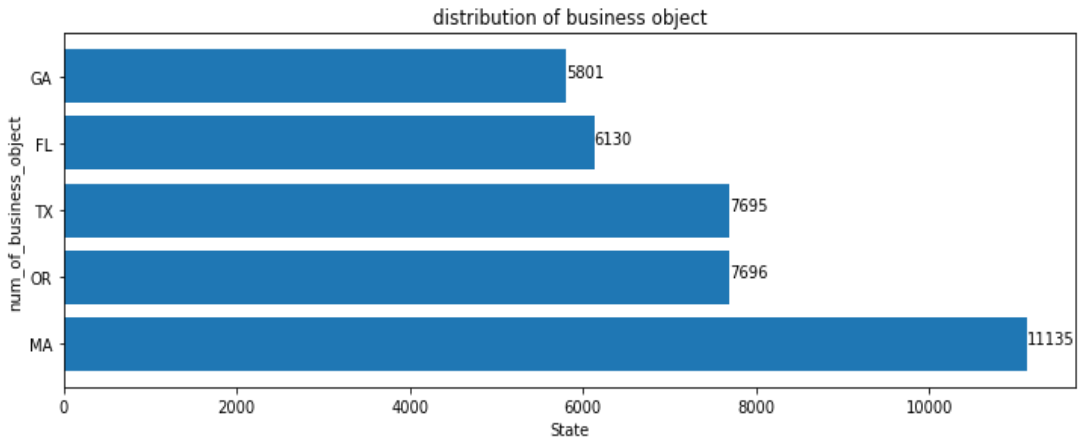
c. Visualization

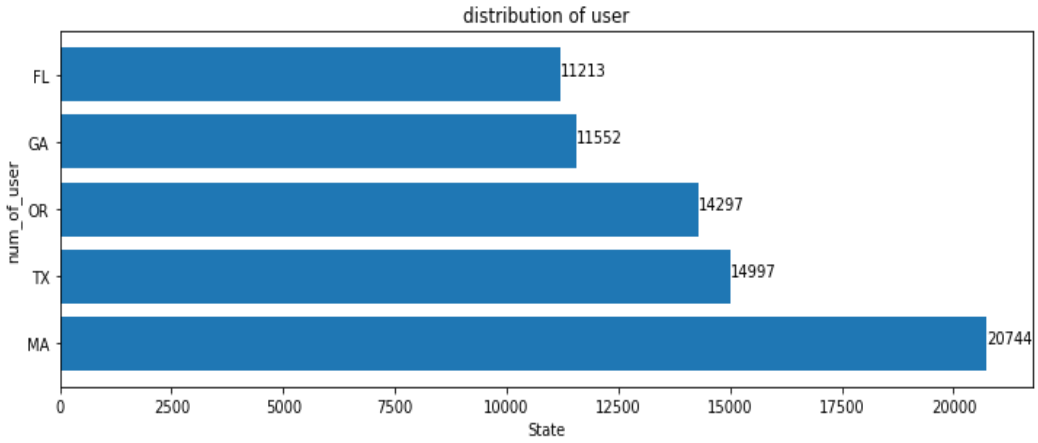
After merging the business dataset and our sample dataset, the size of our data is 100,000 rows with 16 features.

Feature	**Description**
Business_id	ID of business as per yelp
name	Name of the business
address	Exact location of the business
city	City in which it's located
state	Business established state
postal_code	Zip code of the business address

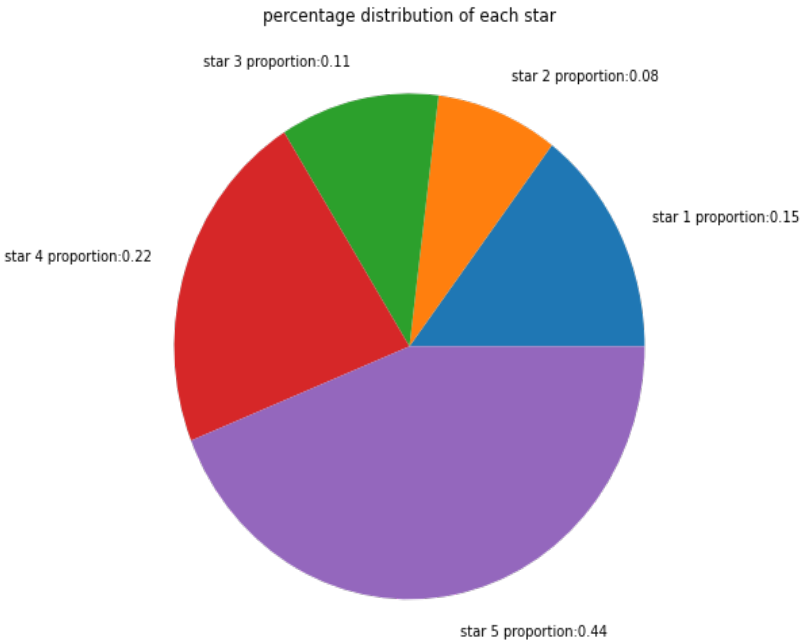
stars	Rating of each comment (1-5 stars)
review_count	Number of reviews for the business
review_id	ID for the review
user_id	ID for the reviewer
Avg_stars	average rating of each restaurant
useful	Number of other yelp users mark the review useful
funny	Number of other yelp users mark the review funn
cool	Number of other yelp users mark the review funny
text	Review description of each business
date	Date and time of the review

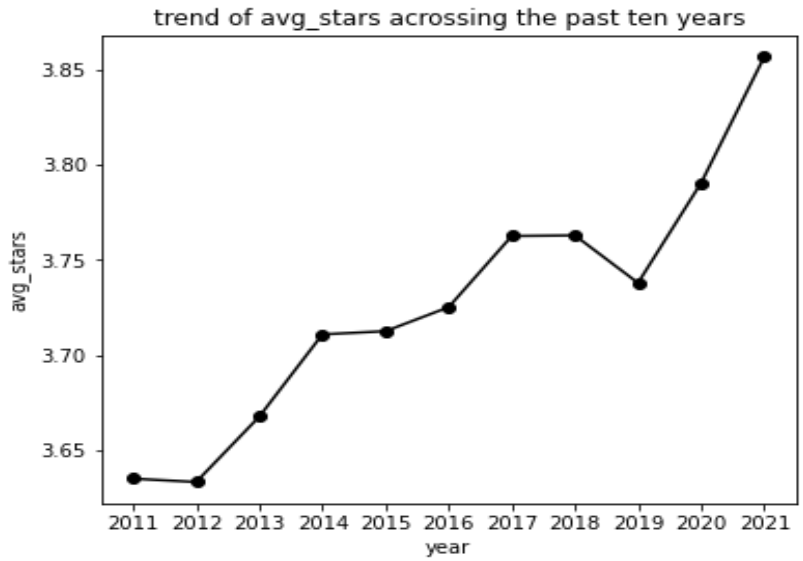






We can see from the above graph that GA(Georgia), FL(Florida), TX(Texas), OR(Oregon), MA(Massachusetts) are the top five states that have the largest number of restaurants and users. From the right side of the graph we can see that TX(Texas) and OR(Oregon) are the two states with the highest percentage of restaurants with good reputation.





From the pie chart, we can see that positive (4 or 5 stars) reviews take up 66% of the dataset, the negative (1 or 2 stars) reviews take up 23% of the data and neutral (3 stars) reviews take up 11% of the data in our sample dataset.

What's more, when we observe the trend of average stars across the past 10 years, the average rating of restaurants on Yelp is increasing fairly consistently.

3. Data Cleaning & Pre-processing

```
 1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100000 entries, 0 to 99999
Data columns (total 16 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   review_id     100000 non-null  object
 1   user_id       100000 non-null  object
 2   business_id   100000 non-null  object
 3   stars         100000 non-null  float64
 4   useful        100000 non-null  int64
 5   funny         100000 non-null  int64
 6   cool          100000 non-null  int64
 7   text          100000 non-null  object
 8   date          100000 non-null  object
 9   name          100000 non-null  object
 10  address       100000 non-null  object
 11  city          100000 non-null  object
 12  state         100000 non-null  object
 13  postal_code   100000 non-null  object
 14  avg_stars     100000 non-null  float64
 15  review_count  100000 non-null  int64
dtypes: float64(2), int64(4), object(10)
memory usage: 13.0+ MB
```

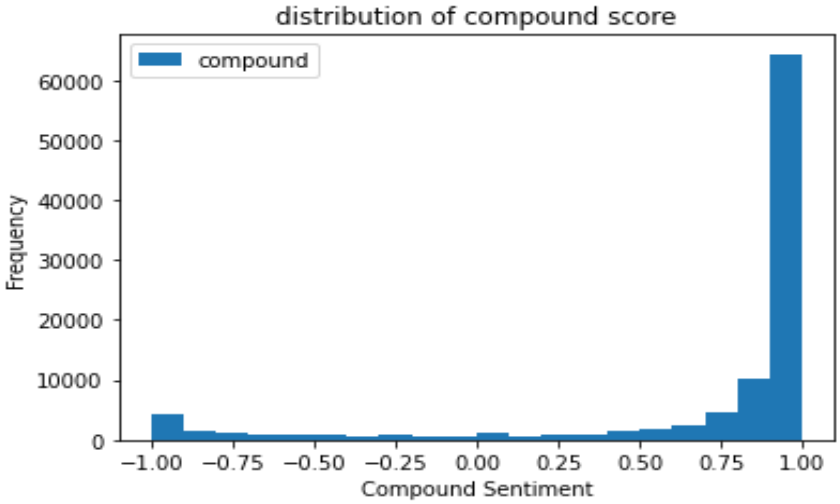
We can see from the above graph that our dataset doesn't contain any NA/null values, so no null value data cleaning was required.

```
807      ディビジョンのローカルなフローズンヨーグルト屋さん。自前のキャラも作って、お店も広くて清潔。...
1075     Solid modern Shanghainese food. Quality ingred...
2218     店員のおねーちゃんがタイ人かな？とっても笑顔が素敵でした。お水もなくなったらすぐ入れに来てく...
2627     一月十四日我們四人叫了菜，其中一道油豆腐已經壞了，有告知服務人員，他說另外給我們一份油豆腐，...
3687     20180504 after my interview, I went downtown V...
                                ...
92533    没想到吃火锅会食物中毒！ 我和先生前天晚上10月11号晚5点后到达火锅店，以前也来过这家两次...
93090    Restaurant was rather packed and required some...
94282    小さな屋台である。希望を聞いて巻いてくれるが、スパムと野菜を巻いてもらって美味しかった。ソウ...
97901    We really like their fish bbq 烤鱼 and also real...
99499    Among all, Kung Fu (功夫茶 is probably the sounde...
Name: text, Length: 64, dtype: object
```

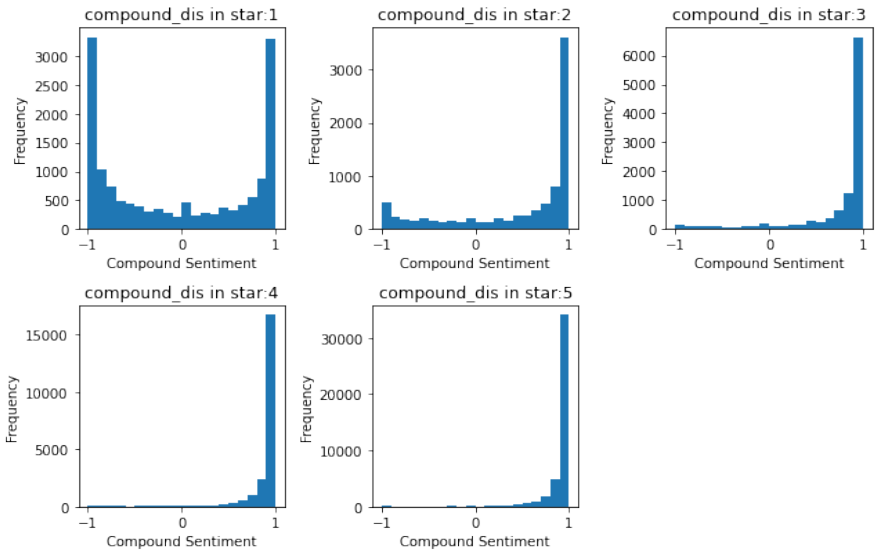
Since we wanted to mainly focus on English, we decided to remove any comment that was written in any foreign languages. For text data, first we used regex to remove special symbols like "!,?" etc. Then we turned English letters into lowercase. After that, we applied the NLTK package to remove all of the stopwords in the reviews. Last but not least, we also "amplified" each user's comment by making use of "useful" features.

4. Sentiment Analysis

Before doing any sentiment analysis, we manually labelled any comment with 1 or 2 stars as negative comments, any comment with 3 stars as neutral comments and any comment with 4 or 5 stars as positive comments.



Since the average stars of our dataset is around 3.5, we can see that the distribution of compound score is skewed to positive 1 from the above graph.



From the above graphs, we can see that with the increase of stars, the distribution of compound scores is moving toward to positive 1.

For neutral comments we labelled any 3 stars neutral comments with compound score less than the 25% quantile as a negative comment, and any 3 stars neutral comment with compound score larger than the 75% quantile as positive comment.

5. Topic Modeling

We are interested in the reasons why customers give negative or positive comments. We can achieve this purpose by using machine learning tools to extract keywords. Using this method, we can offer general advice to restaurants that received low stars to improve their performance in specific aspects and inform those with a high reputation in which aspects they should keep their advantages.

The machine learning tools that we used to build our model are TfidfVectorizer and TruncateSVD. The TfidfVectorizer function uses text data as an input and outputs a document-term matrix. The form of the matrix is (A, B) C. The 'A' represents the index for each comment, the 'B' represents the index for each keyword, while 'C' is TF-IDF score which indicates the level of importance of the keyword 'B' in comment 'A'. In the TF-IDF score,TF stands for Term Frequency and IDF represents Inverse Document Frequency. The higher the TF-IDF score, the stronger the importance of a word.

$$TF = FM$$

$$IDF = \log(1+nb1+df(d,t))$$

$$TF\text{-}IDF = TF \times IDF$$

'F' stands for the number of times a specific word appears in a sentence and 'M' stands for the total number of words in a sentence. 'nb' is the total number of documents in the corpus package and 'df(d,t)' is the number of documents that contain this word.

For TruncateSVD, we will use the document-term matrix as the input and the output is a topic-term matrix. Each row in the matrix represents a topic and each value in each row indicates the level of importance of the corresponding keyword in the corresponding topic.

5.1 Model Building Process

Step1: Customize our stopwords list

The first step to build our model is to create our own stopword list. We built our own stopword list by combining the python stopword packages spacy and nltk. The first stopword package contains 326 English stopwords while the second package contains 179 words. We also manually created 20 stopwords. Therefore, our own stopword list has 525 words in total.

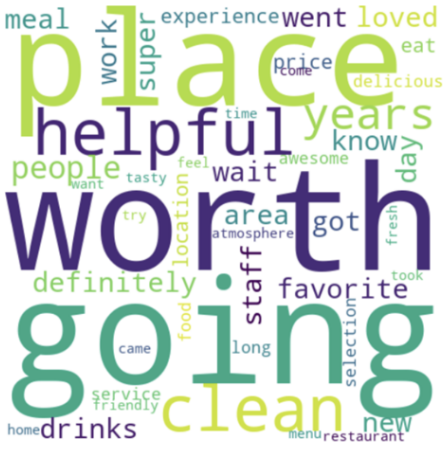
Step2: Parameter Adjusting

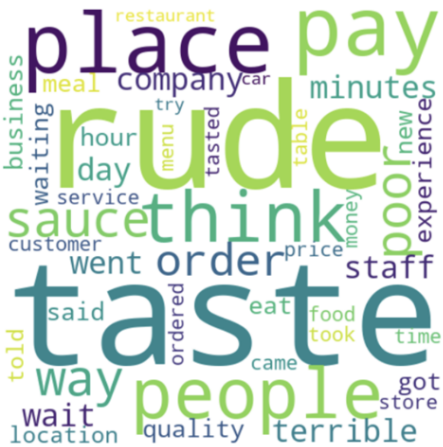
For TfidfVectorizer, we set the parameters: max_features as 500, max_df as 0.35, and min_df as 0.05. In our project, max_features represents the max number of keywords that we want to keep. And [max_df,min_df] = [0.05,0.35] means when building our model, we ignore any keywords whose document frequency was lower than 0.05 or higher than 0.35. What's more, we set the parameter n_components as 10 in TruncatedSVD. This parameter indicates how many rows we want to have for the topic-term matrix.

After finishing these steps, we used text data as the input for TfidfVectorier and generated a document-term matrix from this function. Then we used the document-term matrix as the input for TruncatedSVD and recieved a topic-term matrix output. Finally, we sorted the keywords based on the score of each topic and extracted the top 10 keywords in each topic.

6. Result demonstration

a. Positive and Negative comment





The left side of wordcloud shows the reasons why users give positive comments (4 or 5 stars) to the restaurant/business, and the right side of wordcloud shows the reasons why users give negative comments.

b. Neutral comment

The left side of wordcloud shows the reasons why neutral comments are skewed to being more positive and the right side of wordcloud shows the reasons why neutral comments are skewed to being more negative.

7. Conclusion

Our model successfully extracted why users give positive or negative comments. We also extracted the negative and positive comments from neutral reviews and uncovered the reason why the neutral comment is skewed toward positive or negative. With this information, we can offer accurate advice to restaurants/businesses based on the results that our model generated.

8. Reference

https://www.yelp.com/dataset/documentation/main
https://www.yelp.com/dataset/documentation/main
https://www.adweek.com/social-marketing/yelp-looks-to-differentiate-how-service-businesses-appear-on-its-platform/
https://www.eater.com/2018/11/16/18094979/yelp-stock-plunge-future-viability-competition-google-instagram-twitter
https://www.convinceandconvert.com/social-media-strategy/yelp-success-vs-ratings-and-reviews/
https://downdetector.com/status/yelp/
https://www.statista.com/statistics/278651/yelps-quarterly-net-revenue/