Text Mining: Sentiment Analysis and Recommendation System on Yelp Reviews

Runyu Fang
Department of Computer Science
University of Manitoba
Winnipeg, Canada
fangr1@myumanitoba.ca

Xiaoran Xie
Department of Computer Science
University of Manitoba
Winnipeg, Canada
xiex1@myumanitoba.ca

Yunji How
Department of Computer Science
University of Manitoba
Winnipeg, Canada
howyj@myumanitoba.ca

Zhengzhi Jie
Department of Computer Science
University of Manitoba
Winnipeg, Canada
zhengz1@myumanitoba.ca

***Abstract*—In real life, other people's opinions are crucial in future decisions. Besides, in the commercial field, people's reviews can help merchants know customers' preferences. Online review platforms and merchants can get information from customers' reviews to make personalized recommendations for other customers, from which customers can also benefit. This paper uses multiple text data mining methods to discover the potential information, analyze the text sentiment and construct a recommendation system. We used the Yelp review database to analyze the sentiment inside reviews and found patterns to provide personalized customer recommendations.**

Keywords—Data mining, Clustering, Text mining, Natural Language Processing, Cross Validation, Logistic Regression, Non-negative matrix factorization

I. INTRODUCTION

Data mining aims to look for implicit, previously unidentified, and possibly helpful information in data, such as groups of things that regularly occur together, as described by Leung and Brajczuk [3]. At the same time, the text often has hidden information and connections. Our goal is to discover that hidden information and put it to use. Analyzing business-related data in the era of big data will give us a clear view and direction about how to benefit both consumers and businesses. By analyzing the sentiment of reviews, we explored the trends that customers follow and the kinds of meals, goods, and services that customers cannot wait to share with others. The point is to take many reviews from customers who patronize the merchant and find out what factors make the merchant popular and what factors do not. The result can be used as data for recommendation algorithms that sort results from searching or automatically facilitating the most potential store that users would like to spend their money on. Besides, business owners can evaluate customers' reviews and compare them with the primary keywords summarized, generated, and filtered by our methods, allowing them to see what they can improve to satisfy and attract more customers.

Unlike traditional recommendation systems, the method we generated does not give recommendations based on the user's historical comments and grading records but takes advantage of other users' comments and then forms a recommendation function with algorithms to find similar businesses.

II. BACKGROUND

In this paper, we applied text mining to analyze business review data from Yelp. Yelp is a famous review site founded in 2004. It promotes consumers to discover, connect and engage with local businesses, allowing them to give reviews, make reservations, book appointments, and purchase from different merchants. Yelp's vast reviews database will enable us to try and apply text mining and classification. Generally, we look forward to identifying those businesses loved by people and with good reputations. Based on the data we could get, we found that semantic analysis of tens of thousands of reviews is a good idea for achieving that goal.

We implemented the K-mean clustering algorithm to cluster business locations, used natural language processing techniques to clean the data, vectorized the text of the review by word counts & TF-IDF, visualized the word frequency, found associate words in context by Word2Vec, searched the optimal hyper-parameters and classification model by k-fold cross validation, found the most suitable sentiment analysis classifier by comparing the results of logistic regression and support vector machine, extracted principal components of business by non-negative matrix factorization and retrieved similar business by k-mean clustering.

III. MAIN BODY

A. Data Preprocessing

1) Most frequent states and cities

To begin with, we selected the database from Yelp in the United States. We started by looking at the top 10 states with the most merchants. As Figure 1. shows below, we found that, among all US states, Pennsylvania has the highest number of businesses in the Yelp database. And then, we went deeper into Pennsylvania and tried to find the cities with the highest number of businesses. We discovered that Philadelphia city has the highest business data and is significantly higher than other cities in the Pennsylvania dataset Figure 2.

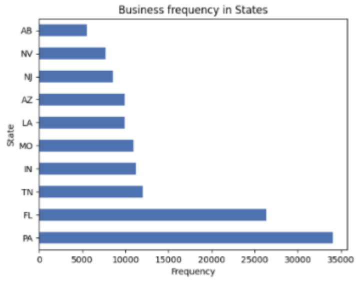


Fig. 1.

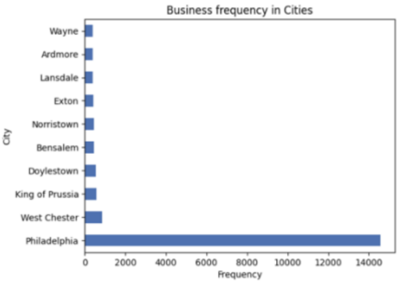


Fig. 2.

2) Most frequent businesses categories

Our next step is to find the most frequent business categories in the dataset of Philadelphia. There exist many categories to describe businesses. We calculated the frequencies of every category type of business in Philadelphia Figure 3. Then we chose the top 7 most frequent categories. However, we found that there are many intersectant categories. For example, the categories “Food” and “Restaurant” are somehow covered by each other. To solve this problem, we created seven general categories for better classification. And businesses that did not fall into these seven categories were classified as our 8th category, "Other." Therefore, our new category list includes "Restaurants," "Shopping,” "Nightlife,” "Beauty & Spas,” "Home Services,” "Local Services,” "Health & Medical,” and "Other." Some businesses were marked with multiple general categories. In this case, we only classified those businesses with the category with the highest frequencies.

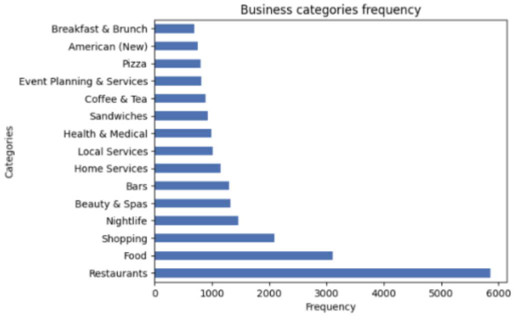


Fig. 3.

After generating general categories, we combined them with the city map to create a scatterplot Figure 4. The horizontal and vertical coordinates represent longitude and latitude, respectively. From the summary Figure 5, we found that businesses in the "Restaurant" categories have the highest frequency and dense distribution in Philadelphia. Therefore, in the future stages, we will only consider the restaurant businesses in Philadelphia.

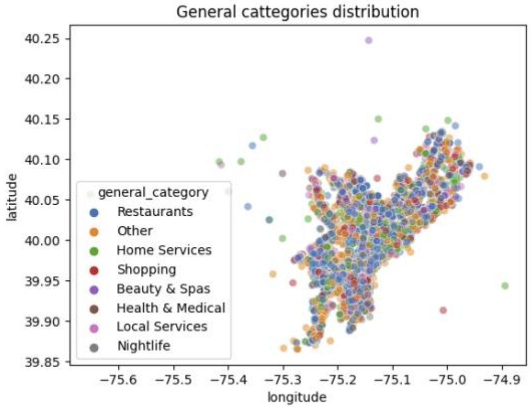
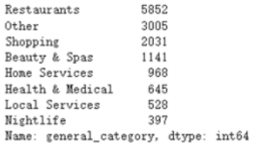


Fig. 4.



```
Restaurants          5852
Other                3005
Shopping             2031
Beauty & Spas        1141
Home Services         968
Health & Medical      645
Local Services        528
Nightlife             397
Name: general_category, dtype: int64
```

Figure 5.

3) K-Means Clustering

In this stage, we performed K-Means clustering to our data and analyzed the result. Wu[6] described that the K-Means method is simple and very efficient for analyzing various data types. We used the K-Means clustering method to find the restaurant density in specific city areas. We divided the restaurants in Philadelphia into six clusters.

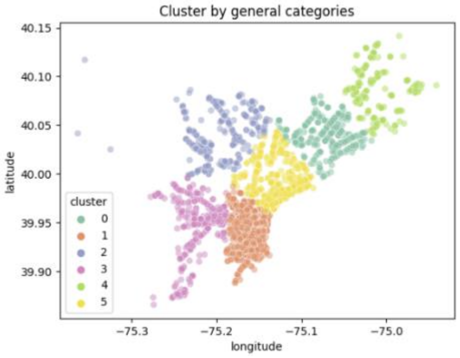


Fig 6.

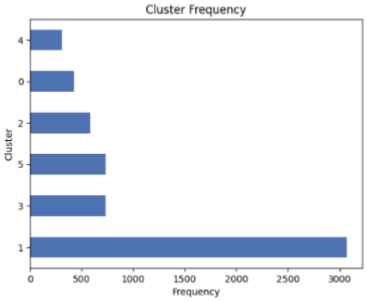


Fig. 7.

As we can see from figure 6 and figure 7, there is a significant amount of restaurant businesses located in the South Philadelphia and Center City areas (cluster 1). Still, this result might not be accurate because we needed to know if Yelp covered most merchants enough in other areas.

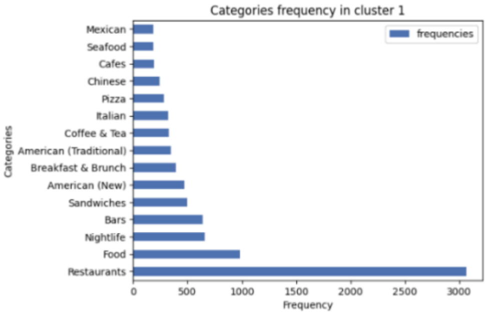


Fig. 8.

Figure 8 shows the frequency of different categories in cluster 1. After excluding the general categories like "Restaurants" and "Food," the categories are evenly distributed. The most frequent category, "nightlife," has occupied around 20% of businesses in cluster 1. Then we decided to move on with the restaurant businesses in the cluster 1 area. Randomly chosen 500 restaurant businesses in this area will be our dataset to analyze. We will randomly find 30% of reviews from customers who rated these 500 restaurants.

B.Process Reviews

In the last phase, after narrowing the range of available locations from all over the United States to the state of Pennsylvania and from Pennsylvania to Philadelphia. Among all Philadelphia merchants, the most frequent merchant type appearing on Yelp is restaurants. Accordingly, we selected 500 target restaurants and used the business ids of these restaurants to find the corresponding database of reviews. Of course, the reviews were multilingual, with English as the most common language. To facilitate subsequent natural language processing (NLP) of the text, we chose English as the target language. At this point, the initial filtering of the review's dataset is almost complete.

1) Balance Sample

First, a grid of adjacent bar charts was created to present the data from the reviews. We had to investigate whether there was any correlation between the length of the text of the user reviews and their ratings (expressed as 1-5 stars). The distribution of the horizontal coordinates (text length) in the following five graphs shows that most of the text lengths of the reviews are concentrated between 0 and 1000 words, with peaks concentrated around 200-300 words. As the text length increases, the number of comments decreases. This finding is consistent with users' usual commenting habits: users tend to use their smartphones to comment, so writing too many words for a review is impractical. The five graphs represent five different stars, and the vertical coordinates represent the number of reviews. We found that as the number of stars increases, the number of words tends to increase. Most users who leave reviews are likelier to rate a merchant's service or product with 4-5 stars, and many reviews with high stars are long. In contrast, there were fewer low-star reviews, and they tended to be short.

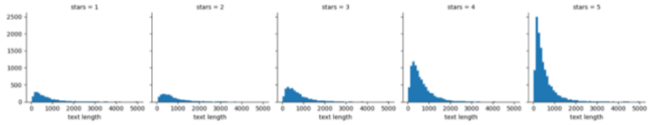


Fig. 9.

Next, the dataset contains four comment features: "text length," "useful," "funny," and "cool," derived from users' perceptions of comments posted by other users. We used a thermal map below to show their correlation to see if there is any interaction between these features. First, we exclude the diagonal data because it compares the data and itself, which is necessarily 1. Next, it is clear that "useful" is closely related to "fun" and that text length is also associated with both "fun" and "useful." Users tend to express their opinion on comments that are interesting and useful. Comments with longer text lengths are usually rated as "useful" and "interesting." This is undoubtedly true as short reviews generally lack descriptive and evidential support due to space constraints, making it difficult for them to resonate with other users.

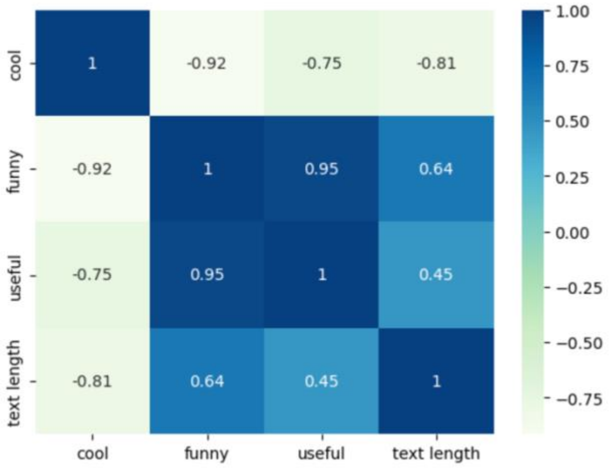


Fig. 10.

We need to explore Yelp's database of reviews further. As we said in the introduction, we aim to build a sentiment classifier for review screening. So, we will start gradually focusing on analyzing the features associated with emotions. Since the goal is to find reviews with sentiment, we filtered the data by the star ratings 1-5 that come with the Yelp dataset. We excluded 3-star reviews since they cannot give an evident emotion, cannot measure positive and negative, and are in a neutral state. Therefore, reviews with more than three stars can be categorized as positive, and those with less than three stars can be classified as unfavorable.

Then, we encountered a problem: the positive and negative reviews differed significantly in sample size. In the analysis process, if the number of positive and negative samples differs significantly or if there is a significant difference between the evaluation categories, then the categories we analyze using the classification model will be biased toward predicting the most frequently occurring samples. Even if a high accuracy rate is obtained in the end, this does not prove the goodness of the classification model algorithm.

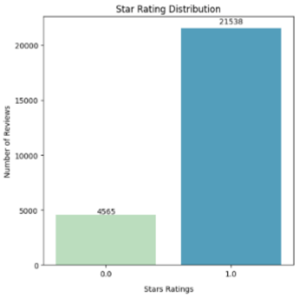


Fig. 11.

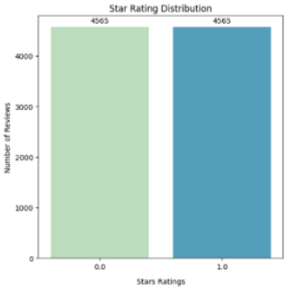


Fig. 12.

With the bar chart above, we can see the gap between the positive and negative samples now. The number of reviews in these two categories was not balanced, with the positive category accounting for nearly 20,000 reviews, while the harmful type accounts for only about 4,000 reviews. To address the imbalance and maintain the value of the accuracy of the sentiment analysis, we chose the side with the higher number of under-samplings. We down-sampled the positive reviews to the same amount as the negative ones to keep the data balance.

2) *Clean Data*

After dealing with the validity issues of the samples, we move on to cleaning up the data. Since the original text format in the comments is plain text, there are special symbols in this format: line breaks, quotation marks, etc., so we need to detect these special symbols and remove them. We use regular expressions for removing special characters such as URL, @, #hashtag, \n, etc. In addition, English words have different tenses, suffixes, and lexical forms in the context of the reviews. The presence of different morphologies of the same semantic word in a database causes the frequency of that word to be diluted by its other variants.

Moreover, while visualizing the word frequency performed later, we composed the bag-of-words model by calculating the word count. Different variants of a word are counted in the model, diluting the frequency of the word. This situation hurts the correct calculation of semantic frequencies, so we used word stemming and lemmatization to remove the affixes of words and extract only the central part of the word. Stem extraction converts words with different forms of the exact origin into stems, such as "burgers" to "burger" and "kindness" to "kind." The morphological principle also transforms words from different tenses to their original forms, such as "knew" to "know" and "known" to "know." This can not only count the words with the same semantic meaning together and restore the occurrence density of the word but also effectively solve the collision caused by different variants as high-frequency words simultaneously.

Then, we modified the default list of stop words and used it for review decomposition. As an essential tool for natural language processing, stop words are often used to improve the quality of text features or reduce their dimensionality. In other words, it is a tool for removing words that can be ignored in a sentence. After researching the default set of stop words, we found some words that we needed but discounted in this set: "no," "less," and so on. These words have solid emotional expressions. Thus, we remove the potentially emotionally biased words from the stop words list. Then we add to the list words that we think are not emotionally dominant, such as "super," "duper," "very," etc. By doing the above, we can finally decompose a review into an array of tokens with analytical meaning.

3) *N-Gram: Present words combinations*

The classification algorithm we used relies on some form of function vector to perform the classification task. “To draw insights from unstructured text data, which is the goal of text mining and natural language processing (NLP), the important and core step is transforming these unstructured text data into numerical vectors[7].” The easiest way to convert the corpus to feature vector format is through the “bag of words.” The bag of words model can convert a sentence into a vector representation. This method does not consider the word order of the words in the sentence but only the number of occurrences of the words in the word list of the sentence. In other words, it counts the frequency of words appearing in a sentence. However, it is not enough to present only the word frequency of words to explore the potential value of the reviews database, so we strengthened the Bag of words model by combining the N-gram model. In the N-gram model, N represents the number of words that make up a phrase: 1-gram refers to one word, bi-gram to a combination of two words, and so on. We also find another problem with the order of words. Like the bag of words model, the bigram does not contain order per se and can, therefore, only be considered a representation of a simple combination of words. For example, "this place is great" does not fit the phrase "place great" due to the use of both stop words and tokenization in the clean data phase. The same problem applies to the 3-gram.

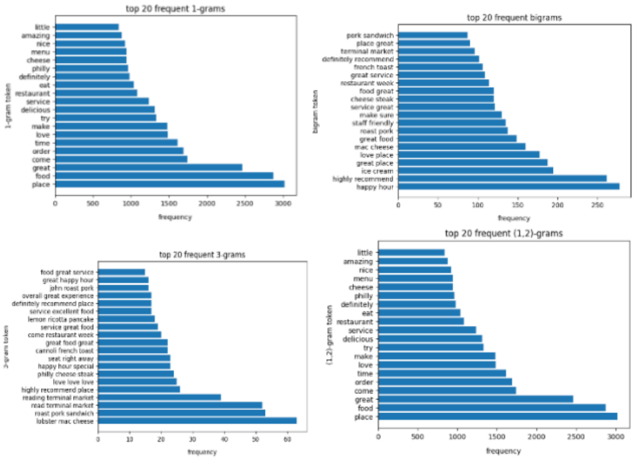


Fig. 13.

The figure 13 above clearly show mining results using the N-gram model from positive reviews. In the 1-gram diagram, we find that the words: “food,” “place,” and “order” appear very frequently in reviews. We only took the top 20 entries from all the frequent 1-gram items. The figure's frequency of several smaller 1-gram tokens has reached about 1,000. Based on the size of our data set, the word frequency reflected by 1-gram is very intuitive, representative and universal. The word cloud shown later is also based on 1-gram. But when describing the results, the word cloud will lose its practical significance because it is too general. As seen in the horizontal coordinate dimension of bigram and 3-gram graphs, the frequency of two or three words has been significantly reduced. The maximum frequency of the entry in bi-gram is only about 250, and the highest frequency of 3-gram is only about 60.

However, combined words can provide more explanatory results to a certain extent. Take happy hour and roast pork sandwich as an example: with the single word "hour" alone, we can't tell whether this comment describes the time waiting for the meal or the mealtime. But if we use the combination: of "happy hour," it can represent the customer's emotions and states. For example, a single "sandwich" may indicate the heat of this food type, but a "roast pork sandwich" can be more apparent to the specific name of the popular dish. Exact words can benefit advertising delivery and give customers more precise recommendations when they consume. This is a trade-off: the frequency of an entry decreases while the descript ability increases. 1, 2 -gram is a compromise method, which includes 1-gram and bi-gram. However, because the frequency of 1-gram entries appears very high, the first 20 entries in the (1, 2)-gram graph do not contain the combinations shown in the bi-gram, but this is a compromise solution.

The Bag of words model has many disadvantages: it does not consider the order between the words in the sentence, nor can it reflect the keywords in the sentence and the context, so it will cause the accuracy of the classifier for subsequent training to be missing. To solve this problem, we later adopted the TF-IDF model.

4) *Word Cloud: Visualize top frequent words*

We have made a rough distinction between positive/negative comments based on star ratings. And we cleaned up and sorted out all text comments. Similar to the purpose of the bar graph presented by n-gram above, we now want to use the word cloud to observe more intuitively: which are more frequent words that appear in different emotions.

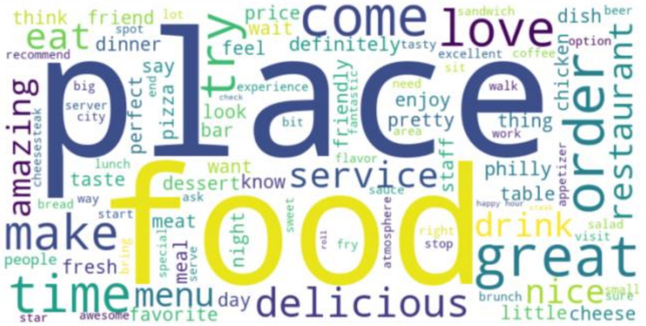


Fig. 14.

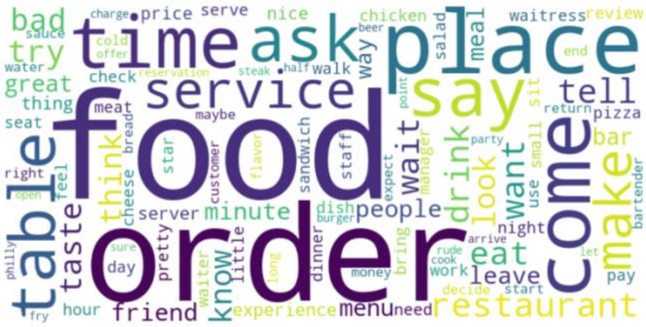


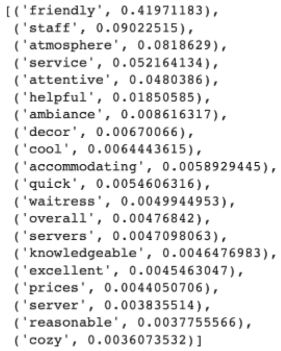
Fig. 15.

Figure 14 presents a positive attitude-related word cloud. It is easy to see that users frequently mention food and place in reviews, which is entirely in line with the characteristics of the restaurant's reviews: food and location/store are essential considerations for customers when making comments. Strong emotional adjectives, such as delicious, amazing, and great, will serve as the post-training emotional classifier. In addition, some words can provide reasoning, such as wait, service and friendly. From these words, users' preferences for restaurants are not limited to food, and a better dining environment is likely a potential trend.

Figure 15 presents a negative word cloud; many frequently mentioned words are in positive reviews. And this is interesting that many words appear in positive and negative clusters. Food is always a high-frequency word. Still, it is also reasonable that the quality of food is the first consideration for the public to comment on the quality of a restaurant. Similarly, there are other factors like time, order, etc. Although they do not have strong emotions, they can still provide valuable information. For example, time is about the length of waiting time, and order is about the speed and attitude of ordering. This speculative information will be helpful to business advice for entrepreneurs in the catering industry and a potential criterion for restaurant consumers to choose restaurants.

5) *Word2Vec: Find relevant words in context*

We can get phrases or multiple words that appear more frequently in reviews from the text processing methods mentioned above. However, due to the model's limitations, only the word frequency of the text in the entire data set can be obtained, resulting in limited results. Therefore, we hope to find words related to the context of specific terms. After that, we can understand the background and reasons for the emergence of high-frequency words and make it possible to provide more comprehensive consumption advice to Yelp users.



```
[('friendly', 0.41971183),
 ('staff', 0.09022515),
 ('atmosphere', 0.0818629),
 ('service', 0.052164134),
 ('attentive', 0.0480386),
 ('helpful', 0.01850585),
 ('ambiance', 0.008616317),
 ('decor', 0.00670066),
 ('cool', 0.006443615),
 ('accommodating', 0.0058929445),
 ('quick', 0.0054606316),
 ('waitress', 0.0049944953),
 ('overall', 0.00476842),
 ('servers', 0.0047098063),
 ('knowledgeable', 0.0046476983),
 ('excellent', 0.0045463047),
 ('prices', 0.0044050706),
 ('server', 0.003835514),
 ('reasonable', 0.0037755566),
 ('cozy', 0.0036073532)]
```

Fig. 16.

Through the Word2Vec model, the vector representation of the word can retain the context information of the word in the corpus. The Skip-gram model supports predicting the words around a central word. Therefore, we can explore the most likely words in the context of the target word, which can explain the appearance of high-frequency words to a certain extent. For example, when we only consider positive reviews and use service as input, we may get words such as friendly, staff, atmosphere, helpful, etc., which can describe the service in high-frequency words—the language environment. We can speculate that the service that impresses customers is inseparable from the friendly staff, nice atmosphere, and other restaurant characteristics. These conclusions can be used as business advice to allow entrepreneurs in the catering industry to consider these factors before investing.

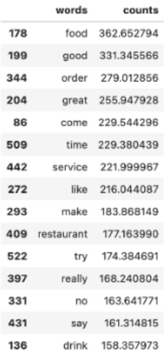
C.Sentiment Analysis

We have selected the data set to balance the degree of difference between samples. After clearing the data, the data set is roughly distinguished according to the number of stars, reviews with the above three stars were classified as positive, marked as 1. Reviews with less than three stars were classified as negative and marked as 0. These tags are stored in a new column of our data set. This emotional label is based on the Stars rating.

Although Stars is a helpful tool for understanding the overall emotions of products or services, they do not provide detailed background information about why people have specific feelings. Therefore, Stars can't always accurately reflect the mood of commentators. Some people may give low ratings just because they have had a bad experience with customer service. Others may give merchants a higher rating even if dissatisfied with any product or service aspect. For example, the dishes in a restaurant taste delicious, but the waiting time is very long, and some users are likely to get low scores. Therefore, relying on Stars as a basis alone cannot form practical speculation, and we also need to analyze the text emotionally to get a complete result. As a branch of the text mining process, the emotional analysis provides a way to quantify people's feelings about products or services, identify and extract subjective information from text data, and understand the specific causes of people's emotions. Overall, it can provide valuable insights to help us understand the motivations behind emotions and trends in customer reviews. At this stage, we will use the data set with emotion tags to train the review classifier, compare the accuracy of the data generated by different models, and find the most suitable emotion classification model.

1) Vectorization: Words Count & TF-IDF

In the previous stage, we used the word count method of the bag-of-words model to convert the sentences in reviews into a vector representation. However, this method has some disadvantages and limitations: it does not consider the context in which a word appears or the length of the text in which it is located. This means that common words like "the" and "and" will be weighted heavily, overriding rarer words with a greater reference value. Although we can remove stopwords by the step of clean data, we still cannot guarantee to sieve out all non-important words. At this point, a new approach: TF-IDF considers both the frequency of a word in the full text and the rarity of that word in all documents. It reduces the weight of common nonsignificant words. This gives an equal chance to the occurrence of rarer words that contain more emotional information. So, in the next step, we used the optimal n-gram obtained from the above cross-validation as a parameter to vectorize the words in the sentence using TF_IDF. Therefore, we could find the weighted words that occur more frequently in the document.



	words	counts
178	food	362.652794
199	good	331.345566
344	order	279.012856
204	great	255.947928
86	come	229.544296
509	time	229.380439
442	service	221.999967
272	like	216.044087
293	make	183.868149
409	restaurant	177.163990
522	try	174.384691
397	really	168.240804
331	no	163.641771
431	say	161.314815
136	drink	158.357973

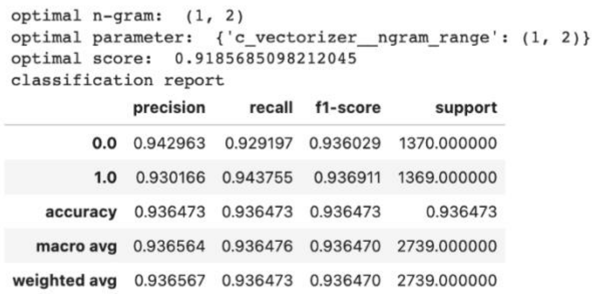
Fig. 17.

2) Cross Validation：Find hyperparameter and classifier

In the previous phase, we visualized the n-gram model and observed the results of combinations of different numbers of grams. We gradually realized that n-grams give different information densities while showing flexibility as a parameter. In this phase, reviews were attached with sentiment labels, and we also focused on sentiment analysis. The next goal was to review the training dataset to develop a generalized sentiment classifier. Since we did not know which n-gram would give the best results for which classification model would be best to handle reviews labeled with the sentiment, we wanted to find an optimal n-gram and classification model by testing different combinations of n-grams and classification models. Schaffer[4] mentioned in his paper that if we include all reasonable proportions in our experiments and choose the classification method in this way, cross-validation has no significant disadvantage compared to the performance of the best strategy, and the effectiveness and feasibility of cross-validation, in this case, can be seen from the text. After partitioning the training and test sets, cross-validation was used to examine all possible hyperparameter values and combinations. Six combinations of n-grams were enumerated as hyperparameters, namely (1,1), (1,2), (1,3), (2,2), (2,3) and (3,3). The GridSearchCV method was applied, and the model was fitted to the training dataset. Along the way, we found that the technique could determine the best values and combinations of hyperparameters that provided the best accuracy. For natural language processing, the back-of-words model is still used here for counting to vectorization. Both logistic regression and support vector models were used for training and comparison.

First, we applied the Logistic Regression model. According to Gaye and Wulamu[1], By creating a hyperplane, the binary prediction probabilistic statistical model of logistic regression analysis predicts between classes of a dataset. The class membership probability measure includes regression coefficients, intercepts, and risk variables. Figure 18 shows the results after training it.

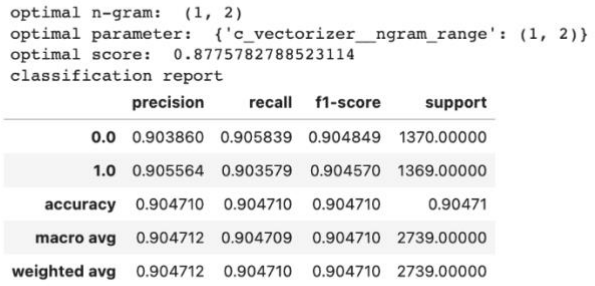
Then, we applied the Support Vector Machine model. As Netzer et al.[5] described, four fundamental ideas may be used to illustrate the SVM algorithm's concept: the soft margin, the maximum margin hyperplane, the separating hyperplane, and the kernel function. Figure 19 shows the results after training it.



```
optimal n-gram:  (1, 2)
optimal parameter:  {'c_vectorizer__ngram_range': (1, 2)}
optimal score:  0.9185685098212045
classification report
```

	precision	recall	f1-score	support
0.0	0.942963	0.929197	0.936029	1370.000000
1.0	0.930166	0.943755	0.936911	1369.000000
accuracy	0.936473	0.936473	0.936473	0.936473
macro avg	0.936564	0.936476	0.936470	2739.000000
weighted avg	0.936567	0.936473	0.936470	2739.000000

Fig. 18.



```
optimal n-gram:  (1, 2)
optimal parameter:  {'c_vectorizer__ngram_range': (1, 2)}
optimal score:  0.8775782788523114
classification report
```

	precision	recall	f1-score	support
0.0	0.903860	0.905839	0.904849	1370.00000
1.0	0.905564	0.903579	0.904570	1369.00000
accuracy	0.904710	0.904710	0.904710	0.90471
macro avg	0.904712	0.904709	0.904710	2739.00000
weighted avg	0.904712	0.904710	0.904710	2739.00000

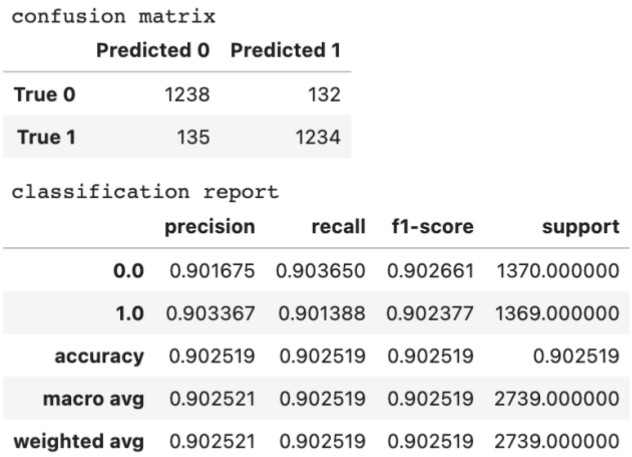
Fig. 19.

We found from comparing the two and cross-validating the results that the optimal parameter is presented as (1, 2), so we chose (1, 2)-gram as the parameter for vectorization and training the classifier. In addition, logistic regression has a higher optimal score, and the accuracy presented in the report is higher than that of SVM, so Logistic Regression was chosen as our target classifier.

3) Supervised Learning: Logistic Regression

The main goal of sentiment analysis is still to distinguish between positive and negative review texts, which is essentially a binary classification task. Hence, the combination of logistic regression is well suited for such tasks. Since there may be more negative examples than positive ones in a collection for sentiment research, logistic regression is resilient to noise. It can manage unbalanced datasets. We also balanced the samples in the data processing stage, so we would theoretically have higher accuracy. And logistic regression can provide interpretable results because it generates a probability score indicating the likelihood that a given sample belongs to a particular category (e.g., positive or negative). This is useful for understanding the reasons behind classification decisions. And with cross-validation, we can also determine if the classifier is a logistic regression. After vectorizing the text, we start training and validating the model. To observe the goodness of the classifier selection, we need to take the help of the Confusion matrix, which is a tool to prove the classifier's performance. Gaye and Wulamu[1] stated that the strategies for categorizing the review dataset might be found by computing statistical measures such as True Positives, True Negatives, False Positives, and False Negatives.

All correct predictions are on the diagonal, so it is easy to visualize where there are errors from the confusion matrix, as they are presented outside the diagonal. The report shows that the classifier has an accuracy of about 90%, which can be applied to most of our reviews for sentiment classification work.



```
confusion matrix
```

	Predicted 0	Predicted 1
True 0	1238	132
True 1	135	1234

```
classification report
```

	precision	recall	f1-score	support
0.0	0.901675	0.903650	0.902661	1370.000000
1.0	0.903367	0.901388	0.902377	1369.000000
accuracy	0.902519	0.902519	0.902519	0.902519
macro avg	0.902521	0.902519	0.902519	2739.000000
weighted avg	0.902521	0.902519	0.902519	2739.000000

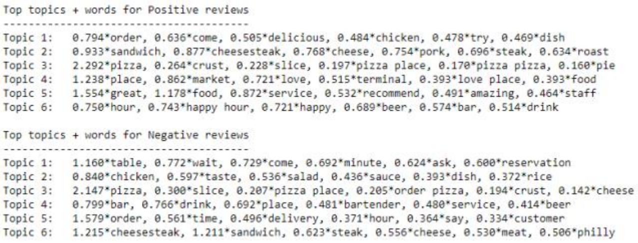
Fig. 20.

D. Recommendation System

1) NMF: Search for principal components

At this point, we have been able to identify the emotions embedded in the user's reviews by using a classifier that has been trained. Here we need to reach a consensus that the reviews identified as positive reflect the user's expressed favourable feelings and preferences towards the restaurant. With this consensus as a premise, we can find restaurants with similar tastes by mining the restaurants where users leave positive reviews to form a restaurant recommendation system. Like the steps of training a sentiment classifier, we will continue to use data vectorized by TF-IDF as our input.

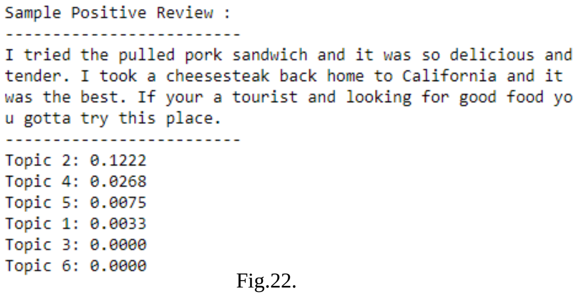
The Non-negative matrix factorization (NMF) method is chosen to find the main components of the reviews, but in this scenario, we can refer to the main components as TOPIC. As Lee and Seung[2] proposed, A matrix may be broken down using NMF. Principal components analysis and vector quantization are examples of unsupervised learning methods that may be thought of as factoring a data matrix under various restrictions. This method obtained the top topics of positive and negative reviews.



```
Top topics + words for Positive reviews
---------------------------------------
Topic 1:   0.794*order, 0.636*come, 0.505*delicious, 0.484*chicken, 0.478*try, 0.469*dish
Topic 2:   0.933*sandwich, 0.877*cheesesteak, 0.768*cheese, 0.754*pork, 0.696*steak, 0.634*roast
Topic 3:   2.292*pizza, 0.264*crust, 0.228*slice, 0.197*pizza place, 0.170*pizza pizza, 0.160*pie
Topic 4:   1.238*place, 0.862*market, 0.721*love, 0.515*terminal, 0.393*love place, 0.393*food
Topic 5:   1.554*great, 1.178*food, 0.872*service, 0.532*recommend, 0.491*amazing, 0.464*staff
Topic 6:   0.750*hour, 0.743*happy hour, 0.721*happy, 0.689*beer, 0.574*bar, 0.514*drink

Top topics + words for Negative reviews
---------------------------------------
Topic 1:   1.160*table, 0.772*wait, 0.729*come, 0.692*minute, 0.624*ask, 0.600*reservation
Topic 2:   0.840*chicken, 0.597*taste, 0.536*salad, 0.436*sauce, 0.393*dish, 0.372*rice
Topic 3:   2.147*pizza, 0.300*slice, 0.207*pizza place, 0.205*order pizza, 0.194*crust, 0.142*cheese
Topic 4:   0.799*bar, 0.766*drink, 0.692*place, 0.481*bartender, 0.480*service, 0.414*beer
Topic 5:   1.579*order, 0.561*time, 0.496*delivery, 0.371*hour, 0.364*say, 0.334*customer
Topic 6:   1.215*cheesesteak, 1.211*sandwich, 0.623*steak, 0.556*cheese, 0.530*meat, 0.506*philly
```

Fig.21.

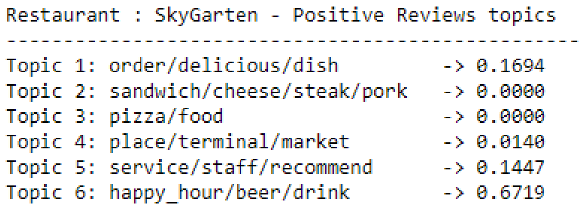
The above figure shows the six topics NMF helped us to decompose the reviews. Our purpose is to help the user find his favourite restaurant based on the positive reviews, so we have been ignoring the negative reviews. Let's take one of the positive reviews and show the topics that it was decomposed from. As the figure below shows:



```
Sample Positive Review :
------------------------
I tried the pulled pork sandwich and it was so delicious and
tender. I took a cheesesteak back home to California and it
was the best. If your a tourist and looking for good food yo
u gotta try this place.
------------------------
Topic 2: 0.1222
Topic 4: 0.0268
Topic 5: 0.0075
Topic 1: 0.0033
Topic 3: 0.0000
Topic 6: 0.0000
```

Fig.22.

From the above figure 22, we can see that the review similarity is biased toward topic 2. We use this method for all positive reviews of the restaurant to find which topics the restaurant is little towards.



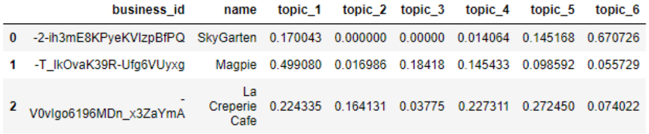
```
Restaurant : SkyGarten - Positive Reviews topics
-------------------------------------------------
Topic 1: order/delicious/dish          -> 0.1694
Topic 2: sandwich/cheese/steak/pork    -> 0.0000
Topic 3: pizza/food                    -> 0.0000
Topic 4: place/terminal/market         -> 0.0140
Topic 5: service/staff/recommend       -> 0.1447
Topic 6: happy_hour/beer/drink         -> 0.6719
```

Fig.23.

We randomly selected a restaurant and calculated the restaurant's weight on these six topics. From the figure 23, we can see that the restaurant is biased towards topic1, topic5 and topic 6, and then we repeat this process for each other restaurant to get the weight of each other on these six topics. So, when two restaurants have similar topics and their weights of the topics are very close. Then we can consider these two restaurants as similar restaurants.

2) K-mean Clustering：Retrieve Similar Restaurant

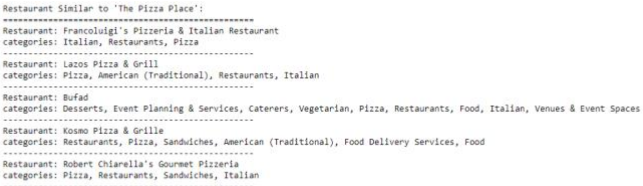
In the previous step, we found a way to find similar restaurants for customers. And in this step, we want to implement this function. Based on the NMF model, we generated six topics, and again after the process mentioned in (D.1), we can finally get the weights of all restaurants on these six topics.



	business_id	name	topic_1	topic_2	topic_3	topic_4	topic_5	topic_6
0	-2-ih3mE8KPyeKVIzpBfPQ	SkyGarten	0.170043	0.000000	0.00000	0.014064	0.145168	0.670726
1	-T_IkOvaK39R-Ufg6VUyxg	Magpie	0.499080	0.016986	0.18418	0.145433	0.098592	0.055729
2	-V0vIgo6196MDn_x3ZaYmA	La Creperie Cafe	0.224335	0.164131	0.03775	0.227311	0.272450	0.074022

Fig.24.

Next, we consider these topics as the features needed for k-mean clustering. After clustering the restaurants into 6 clusters, each cluster is crowded with restaurants with similar topics. So, when we arbitrarily select a restaurant in the dataset, we can find the cluster it belongs to and thus access other similar restaurants. In other words, we see the restaurants that the user might be interested in. To present this method concretely, we randomly selected a restaurant: "The Pizza Place."



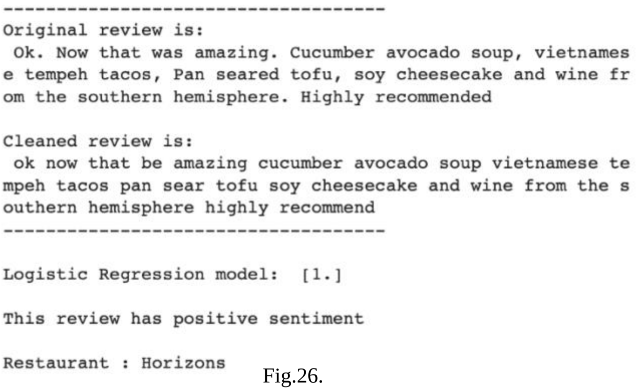
```
Restaurant Similar to 'The Pizza Place':
================================================
Restaurant: Francoluigi's Pizzeria & Italian Restaurant
categories: Italian, Restaurants, Pizza
------------------------------------------------
Restaurant: Lazos Pizza & Grill
categories: Pizza, American (Traditional), Restaurants, Italian
------------------------------------------------
Restaurant: Bufad
categories: Desserts, Event Planning & Services, Caterers, Vegetarian, Pizza, Restaurants, Food, Italian, Venues & Event Spaces
------------------------------------------------
Restaurant: Kosmo Pizza & Grille
categories: Restaurants, Pizza, Sandwiches, American (Traditional), Food Delivery Services, Food
------------------------------------------------
Restaurant: Robert Chiarella's Gourmet Pizzeria
categories: Pizza, Restaurants, Sandwiches, Italian
------------------------------------------------
```

Fig.25.

The figure 25 shows the top 5 restaurants like "The Pizza Place."

3) Application: Sentiment Classifier and Recommendation System

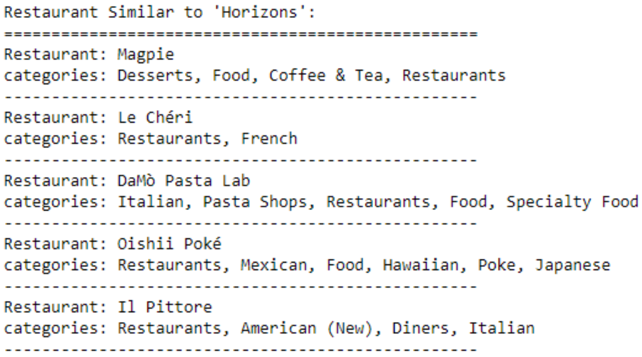
Up to this point, we have completed our text-mining work. Based on the text mining results of the Yelp reviews database, we trained a sentiment classifier and built a system to recommend similar restaurants to customers. Fig.26 shows how to apply these two features to get the information we want. First, we need to build a user scenario: a customer who finishes his meal is so satisfied with the restaurant, then he leaves his reviews on Yelp. He thinks about which restaurant he can have a similar dining experience tomorrow as he did tonight. Our sentiment classifier can identify the polarity of this review. If it is positive, we will find him a similar restaurant as his choice based on the id of this restaurant.



```
-----------------------------------
Original review is:
 Ok. Now that was amazing. Cucumber avocado soup, vietnames
e tempeh tacos, Pan seared tofu, soy cheesecake and wine fr
om the southern hemisphere. Highly recommended

Cleaned review is:
 ok now that be amazing cucumber avocado soup vietnamese te
mpeh tacos pan sear tofu soy cheesecake and wine from the s
outhern hemisphere highly recommend
-----------------------------------

Logistic Regression model:  [1.]

This review has positive sentiment

Restaurant : Horizons
```

Fig.26.

The example shown in Fig.26 analyzes the review's sentiment, while Fig.27 recommends similar restaurants based on the user's preference.



```
Restaurant Similar to 'Horizons':
================================================
Restaurant: Magpie
categories: Desserts, Food, Coffee & Tea, Restaurants
------------------------------------------------
Restaurant: Le Chéri
categories: Restaurants, French
------------------------------------------------
Restaurant: DaMò Pasta Lab
categories: Italian, Pasta Shops, Restaurants, Food, Specialty Food
------------------------------------------------
Restaurant: Oishii Poké
categories: Restaurants, Mexican, Food, Hawaiian, Poke, Japanese
------------------------------------------------
Restaurant: Il Pittore
categories: Restaurants, American (New), Diners, Italian
------------------------------------------------
```

Fig.27.

Note: The category shown in the results is to facilitate observing the results. Restaurant recommendation is not based on category. If you need to review the relevant details, please visit (D.1)

IV. CONCLUSION

Text mining allows us to dig into a massive amount of potential information. With the help of Yelp's dataset, we summarized the information we obtained.

Business：

1. The distribution of the number of different categories of businesses

2. Geographical distribution of specific types of businesses

Reviews：

1. Find the correlation according to the distribution of reviews and stares
2. Words and word phrases that occur frequently
3. The polarity of reviews, reviews sentiment classification and prediction
4. Principal component analysis of the text, principal component analysis of the restaurant
5. Find similar restaurants through positive reviews

In addition, we build an efficient sentiment analysis model and a recommendation system to find similar businesses. As you can see, Text mining benefits both users and companies—by using different types of technology combined with massive amounts of data.

From a business perspective, there is a tremendous demand for sentiment analysis in the industry. Both entrepreneurs, investors, and executive departments want to know how consumers feel about their services and products to better plan their businesses, find the proper positioning for their products, and adjust their business strategies when appropriate. There is a significant demand for recommendation systems from the point of view of customers and consumers of products or services who need to enjoy the best products and services without wasting time looking for restaurants with good reputations.

In addition, many areas could still be optimized during the project. In the data cleaning part, the consideration of text lexicality can be added. When vectorizing the data, the TF-IDF calculation method can be optimized to improve the weight distribution of keywords. In the cross-validation section, more classification models can be compared. When obtaining principal restaurant components, different dimensionality reduction methods can be compared.

V. References

[1] B. Gaye and A. Wulamu, “Sentiment analysis of text classification algorithms using confusion matrix,” Communications in Computer and Information Science, pp. 231–241, 2019.

[2] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[3] C. K.-S. Leung and D. A. Brajczuk, “Mining uncertain data for constrained frequent sets,” *Proceedings of the 2009 International Database Engineering & Applications Symposium on - IDEAS '09*, 2009.

[4] M. Netzer, G. Millonig, M. Osl, B. Pfeifer, S. Praun, J. Villinger, W. Vogel, and C. Baumgartner, “A new ensemble-based algorithm for identifying breath gas marker candidates in liver disease using ion molecule reaction mass spectrometry,” *Bioinformatics*, vol. 25, no. 7, pp. 941–947, 2009.

[5] C. Schaffer, “Selecting a classification method by cross-validation,” *Machine Learning*, vol. 13, no. 1, pp. 135–143, 1993.

[6] J. Wu, “Cluster analysis and K-means clustering: An introduction,” Advances in K-means Clustering, pp. 1–16, 2012.

[7] R. Zhao and K. Mao, “Fuzzy Bag-of-words model for document representation,” *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 794–804, 2018.