Help Phoenix Restaurants on Yelp Attract More Customers

Project Report

BYGB/ISGB 7977 – Text Analytics
Spring 2020

Section 2 – Group 11
Chaoying Bao
Ruoyuan Guo
Peisheng Li
Youping Wang

Instructor: Prof. Evangelos Katsamakas

04/19/2020

FORDHAM UNIVERSITY
THE JESUIT UNIVERSITY OF NEW YORK

Gabelli School of Business

I. Executive Summary

Nowadays, people prefer to use Yelp to find a good restaurant and leave comments for their views on restaurants they have been to.

However, the number of restaurants on Yelp in Phoenix has decreased from 2014 to 2018. It comes to be worthwhile to analyze customers' reviews since reviews indicate customers' real thoughts on restaurants which can help merchants to improve and innovate their restaurants.

This project aimed to help not only restaurants to improve their sales, but also customers to have a better experience on Yelp. After preprocessing data and doing some descriptive stats with Python and Tableau, the project separates into two parts.

In the first part, this project utilized text mining technology to find whether the sentiment in reviews truly reflects the final rating customers give to restaurants. After doing sentiment analysis, this project drew 10 topics from the POS noun tags and used these 10 topics as new variables to predict the restaurant rating with three different classification techniques - Logistic Regression, Decision Tree, and Neural Network by SPSS. The Decision Tree had the best performance, so the project selected this model to help restaurants get deep insights into factors that lead to customers' satisfaction and dissatisfaction.

In the second part, this project used two approaches to build a content-based recommender system. The first one was based on the similarities among restaurants' reviews. CounterVectorizer was used to vectorize and cosine similarity to detect the similarities between restaurants. Then, a function was built to return top 5 similar restaurants. As a result, when customers input a restaurant name, our model will recommend top 5 similar restaurants to them.

The second approach was based on the prediction of users' rating. This is a content-based recommender system consisting of one user profile and one item profile. First, the TFIDF scores of each restaurant's reviews were calculated, and only top 5 keywords for each restaurant were kept according to the TFIDF scores. Then, the item profile of restaurants with the calculated top 5 keywords' TFIDF scores was created. As for the user profile, it contained the ratings of restaurants from users, which helped the system evaluate the preference of the users.

II. Problem Statement

Text mining, also known as text analysis, is the process of transforming unstructured text data into meaningful and actionable information. Text mining utilizes different AI technologies to automatically process data and generate valuable insights, enabling companies to make data-driven decisions[2]. As a result, product reviews are an invaluable source of information for researchers to explore.

Nowadays, people tend to use Yelp to find a good restaurant and leave comments for their views on restaurants they have been to. The contents can help both restaurants and other customers. However, the number of restaurants and reviews on Yelp in Phoenix has decreased from 2014 to 2018. It comes to be worthwhile to analyze customers' reviews since reviews indicate customers' real thoughts on restaurants which can help merchants to improve and innovate their restaurants.

This project will utilize Phoenix customers' reviews to explore the factors leading to customers' satisfaction and dissatisfaction and build a recommender system to provide more related information for customers so that restaurants can attract more target customers.

III. Hypotheses

1. The customers' sentiment in a review is related to the rating.
2. Some topics in the review are related to the rating.
3. The similarity between reviews of customers and restaurants can help customers find more related restaurants.

IV. Data Description

Yelp.com is a website where users can rate local businesses on a 5-star scale where 5 is for the best while 1 is for the worst. Users can also write reviews that describe their experience. The website contains reviews about several different types of businesses including restaurants, shops, nightlife, and beauty spas.

The dataset was from Yelp Open Dataset(https://www.yelp.com/dataset) that contains actual business, user, and users' review data along with check-in information of users and the tips users suggest for different businesses. As the project intended to work with restaurant reviews, the dataset was constrained to Yelp businesses with the Restaurant category, and only used business, user, and user's review data.

The dataset consists of 8,021,122 reviews, 209,393 businesses, and 1,968,703 users. All data was in JSON format and was converted to DataFrame in python.

Table 1. Original Dataset

Name	Size	Type	Description
Business	131MB	JSON	Contains business data including location data, attributes, and categories.
Review	4.97GB	JSON	Contains full review text data including the user_id that wrote the review and the business_id the review is written for.
User	2.31GB	JSON	User data including the user's friend mapping and all the metadata associated with the user.

V. Methodology

i. Outline Steps

Figure 1 shows the diagram of the overall methodology.

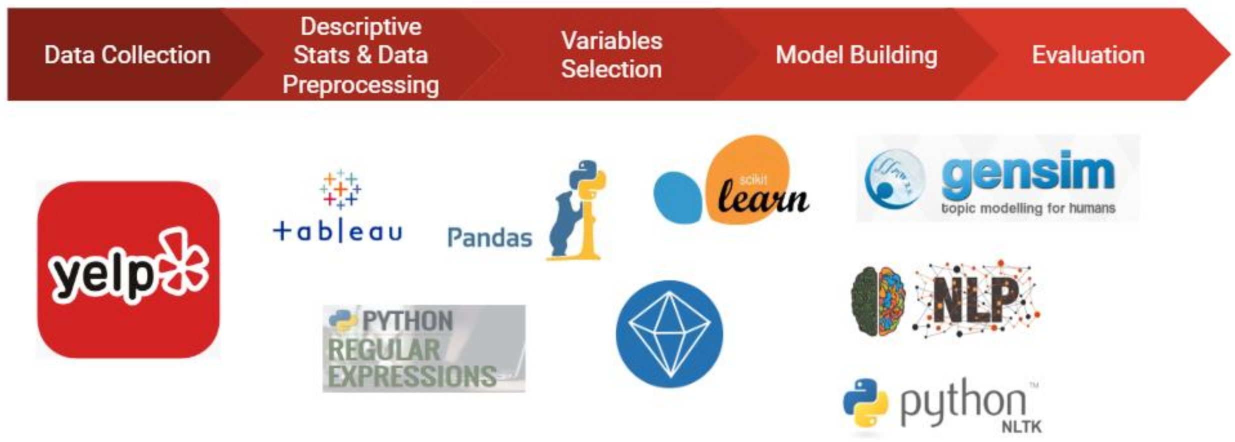


Figure 1. Diagram of Overall Methodology

ii. Descriptive Stats

After converting the JSON file to DataFrame in Python and selecting all businesses with the Restaurant category, the data was used to do descriptive stats with Tableau. Figure 2 shows the results of the descriptive stats. Among all states in the dataset, Arizona has a large number of not only restaurants but also reviews. When the data was drilled down to Arizona, Phoenix has the most review records in the dataset. Besides, from the time series, we can see that both the number of restaurants opening on Yelp and the number of review records in the dataset have decreased from 2014 to 2018. Therefore, the project selected Phoenix as the target city, and 2014 to 2018 as the target time series. Dealing with text data requires much time, to make the process more efficient when the length of reviews was ensured, the reviews were tokenized, and those reviews with 70-300 words were selected in the final dataset. The final dataset used to predict rating has 3504 users, 1694 restaurants, and 5961 reviews.

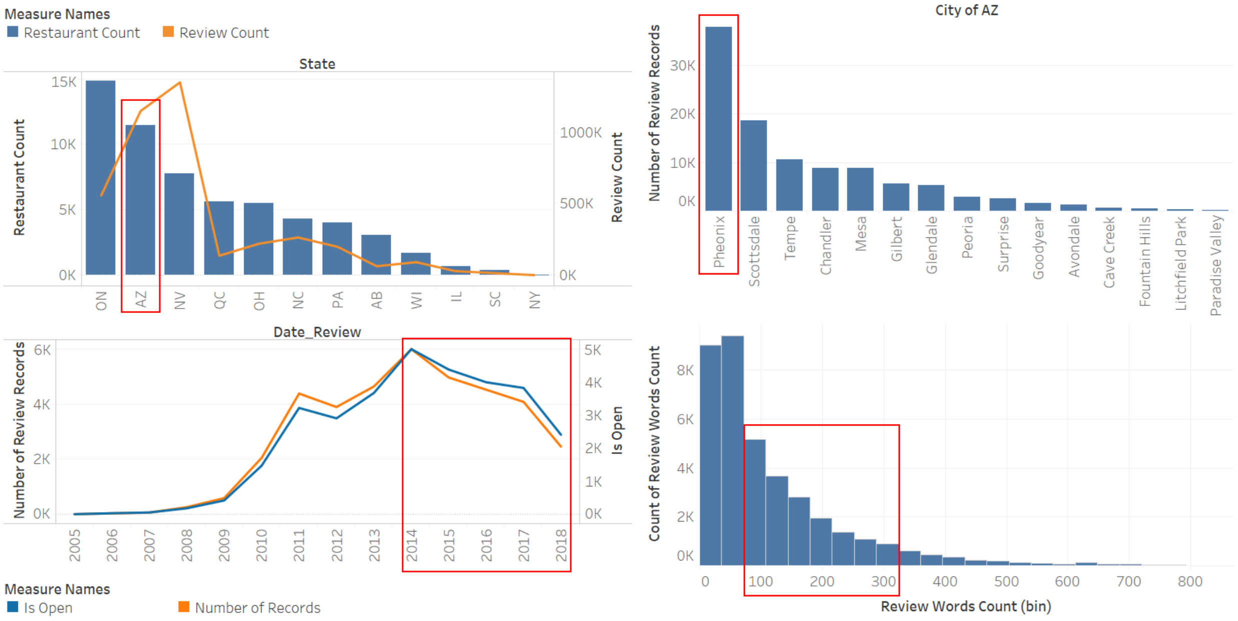


Figure 2. Descriptive Stats

iii. Data Preprocessing

The data preprocessing includes 6 steps.

1. Convert large JSON file to DataFrame in Python
2. Select businesses with the restaurant category
3. Select Arizona as target state
4. Select Phoenix as target city
5. Tokenize the reviews
6. Select reviews with 70-300 words from 2014 to 2018

iv. Variables Selection

After the target records were selected, the reviews were used to do sentiment analysis, POS tagging, stopwords removal, Porter stemming, term-document matrix generation, topic modeling with LDA, and TFIDF calculation. The new variables were created after the data preprocessing was done.

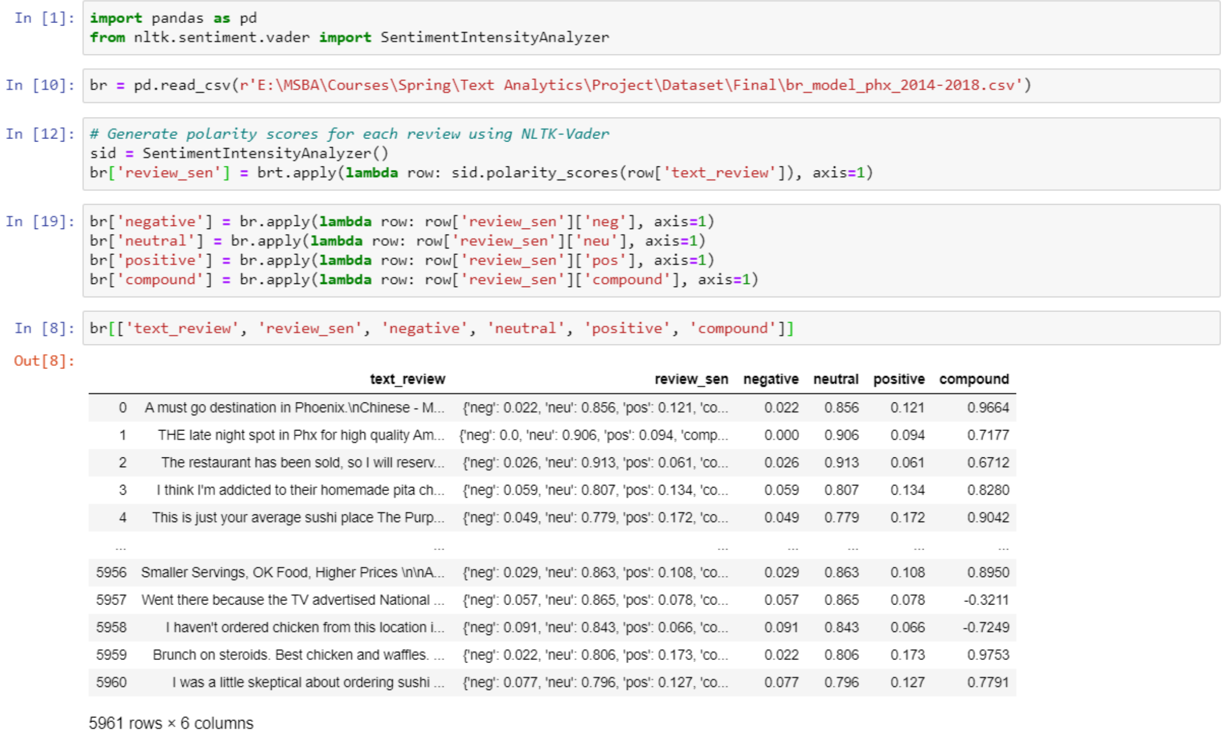
Table 2. Variables Selection for Rating Prediction

Variable Type	**Variable Name**	**Definition**	**Data Type**	**Example**
Independent Variable	useful	Number of useful votes sent by the user	Integer	0
	funny	Number of funny votes sent by the user	Integer	0
	cool	Number of cool votes sent by the user	Integer	0
	postal_code	Postal code of a restaurant	String	94107
	review_count	Number of reviews of a restaurant	Integer	1198
	review_words_counts	Number of words in a review	Integer	68
	negative	Negative score of a review calculated by NLTK Vader	Float	0.022
	neutral	Neutral score of a review calculated by NLTK Vader	Float	0.856
	positive	Positive score of a review calculated by NLTK Vader	Float	0.121
	compound	Compound score of a review calculated by NLTK Vader	Float	0.9664
	“0” - “9”	Weight of 10 topics that were modeled based on POS noun tags in a review	Float	0
Dependent Variable	rating_group	Two groups of business ratings	Integer	0

VI. Model Building

i. Rating Prediction

Before the prediction, the original reviews were used to calculate sentiment polarity scores by NLTK Vader in Python, each review will get four scores - negative score, neutral score, positive score, and compound score.



```
In [1]:  import pandas as pd
         from nltk.sentiment.vader import SentimentIntensityAnalyzer

In [10]: br = pd.read_csv(r'E:\MSBA\Courses\Spring\Text Analytics\Project\Dataset\Final\br_model_phx_2014-2018.csv')

In [12]: # Generate polarity scores for each review using NLTK-Vader
         sid = SentimentIntensityAnalyzer()
         br['review_sen'] = brt.apply(lambda row: sid.polarity_scores(row['text_review']), axis=1)

In [19]: br['negative'] = br.apply(lambda row: row['review_sen']['neg'], axis=1)
         br['neutral'] = br.apply(lambda row: row['review_sen']['neu'], axis=1)
         br['positive'] = br.apply(lambda row: row['review_sen']['pos'], axis=1)
         br['compound'] = br.apply(lambda row: row['review_sen']['compound'], axis=1)

In [8]:  br[['text_review', 'review_sen', 'negative', 'neutral', 'positive', 'compound']]
```

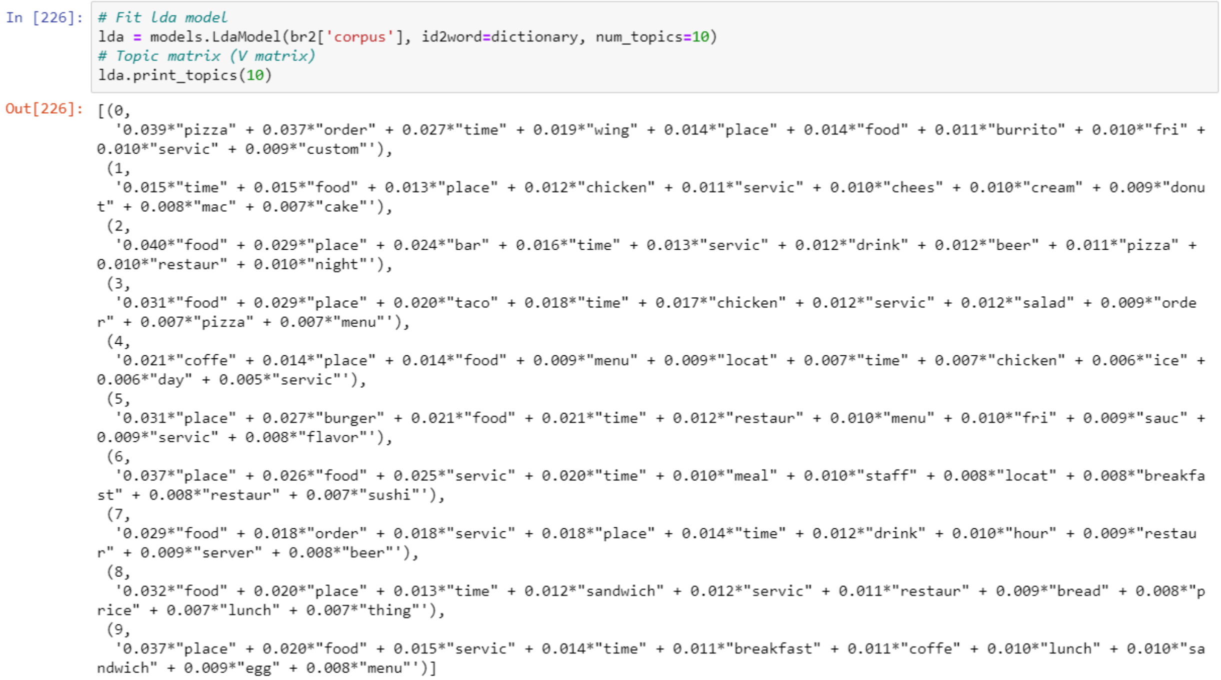
Out[8]:

	text_review	review_sen	negative	neutral	positive	compound
0	A must go destination in Phoenix.\nChinese - M...	{'neg': 0.022, 'neu': 0.856, 'pos': 0.121, 'co...	0.022	0.856	0.121	0.9664
1	THE late night spot in Phx for high quality Am...	{'neg': 0.0, 'neu': 0.906, 'pos': 0.094, 'comp...	0.000	0.906	0.094	0.7177
2	The restaurant has been sold, so I will reserv...	{'neg': 0.026, 'neu': 0.913, 'pos': 0.061, 'co...	0.026	0.913	0.061	0.6712
3	I think I'm addicted to their homemade pita ch...	{'neg': 0.059, 'neu': 0.807, 'pos': 0.134, 'co...	0.059	0.807	0.134	0.8280
4	This is just your average sushi place The Purp...	{'neg': 0.049, 'neu': 0.779, 'pos': 0.172, 'co...	0.049	0.779	0.172	0.9042
...	...	...	...	...	...	...
5956	Smaller Servings, OK Food, Higher Prices \n\nA...	{'neg': 0.029, 'neu': 0.863, 'pos': 0.108, 'co...	0.029	0.863	0.108	0.8950
5957	Went there because the TV advertised National ...	{'neg': 0.057, 'neu': 0.865, 'pos': 0.078, 'co...	0.057	0.865	0.078	-0.3211
5958	I haven't ordered chicken from this location i...	{'neg': 0.091, 'neu': 0.843, 'pos': 0.066, 'co...	0.091	0.843	0.066	-0.7249
5959	Brunch on steroids. Best chicken and waffles. ...	{'neg': 0.022, 'neu': 0.806, 'pos': 0.173, 'co...	0.022	0.806	0.173	0.9753
5960	I was a little skeptical about ordering sushi ...	{'neg': 0.077, 'neu': 0.796, 'pos': 0.127, 'co...	0.077	0.796	0.127	0.7791

5961 rows × 6 columns

Figure 3. Sentiment Analysis with Vader

Meanwhile, the original reviews were used to do POS tagging, and only nouns were kept, because we think nouns can carry more meanings than other types of tags. New word lists with only nouns were used to do stopwords removal and Porter stemming. After the final corpus was built, the term-document matrix was generated, which was used to do topic modelling with LDA.



```
In [226]: # Fit lda model
          lda = models.LdaModel(br2['corpus'], id2word=dictionary, num_topics=10)
          # Topic matrix (V matrix)
          lda.print_topics(10)

Out[226]: [(0,
            '0.039*"pizza" + 0.037*"order" + 0.027*"time" + 0.019*"wing" + 0.014*"place" + 0.014*"food" + 0.011*"burrito" + 0.010*"fri" +
          0.010*"servic" + 0.009*"custom"'),
           (1,
            '0.015*"time" + 0.015*"food" + 0.013*"place" + 0.012*"chicken" + 0.011*"servic" + 0.010*"chees" + 0.010*"cream" + 0.009*"donu
          t" + 0.008*"mac" + 0.007*"cake"'),
           (2,
            '0.040*"food" + 0.029*"place" + 0.024*"bar" + 0.016*"time" + 0.013*"servic" + 0.012*"drink" + 0.012*"beer" + 0.011*"pizza" +
          0.010*"restaur" + 0.010*"night"'),
           (3,
            '0.031*"food" + 0.029*"place" + 0.020*"taco" + 0.018*"time" + 0.017*"chicken" + 0.012*"servic" + 0.012*"salad" + 0.009*"orde
          r" + 0.007*"pizza" + 0.007*"menu"'),
           (4,
            '0.021*"coffe" + 0.014*"place" + 0.014*"food" + 0.009*"menu" + 0.009*"locat" + 0.007*"time" + 0.007*"chicken" + 0.006*"ice" +
          0.006*"day" + 0.005*"servic"'),
           (5,
            '0.031*"place" + 0.027*"burger" + 0.021*"food" + 0.021*"time" + 0.012*"restaur" + 0.010*"menu" + 0.010*"fri" + 0.009*"sauc" +
          0.009*"servic" + 0.008*"flavor"'),
           (6,
            '0.037*"place" + 0.026*"food" + 0.025*"servic" + 0.020*"time" + 0.010*"meal" + 0.010*"staff" + 0.008*"locat" + 0.008*"breakfa
          st" + 0.008*"restaur" + 0.007*"sushi"'),
           (7,
            '0.029*"food" + 0.018*"order" + 0.018*"servic" + 0.018*"place" + 0.014*"time" + 0.012*"drink" + 0.010*"hour" + 0.009*"restau
          r" + 0.009*"server" + 0.008*"beer"'),
           (8,
            '0.032*"food" + 0.020*"place" + 0.013*"time" + 0.012*"sandwich" + 0.012*"servic" + 0.011*"restaur" + 0.009*"bread" + 0.008*"p
          rice" + 0.007*"lunch" + 0.007*"thing"'),
           (9,
            '0.037*"place" + 0.020*"food" + 0.015*"servic" + 0.014*"time" + 0.011*"breakfast" + 0.011*"coffe" + 0.010*"lunch" + 0.010*"sa
          ndwich" + 0.009*"egg" + 0.008*"menu"')]
```

Figure 4. Ten topics with Topic Modeling

The four scores and ten topics along with other original variables were finally used to build classification models to predict the restaurant rating.

All restaurants were divided into two groups. One was the low rating group whose business rating was ≤ 3, and the other was the high rating group whose business rating was >3. Then, 70% of the dataset was partitioned as training data, and the rest 30% was partitioned as testing data. However, the dataset was oversampled with the high rating group, so the SMOTE node in SPSS was used to deal with the imbalance data in the training set.

Then, the Logistic Regression, Neural Network, and Decision Tree were used by SPSS, but Decision Tree showed the highest accuracy, which is 80.39%. Thus, the Decision Tree was selected as our final model.

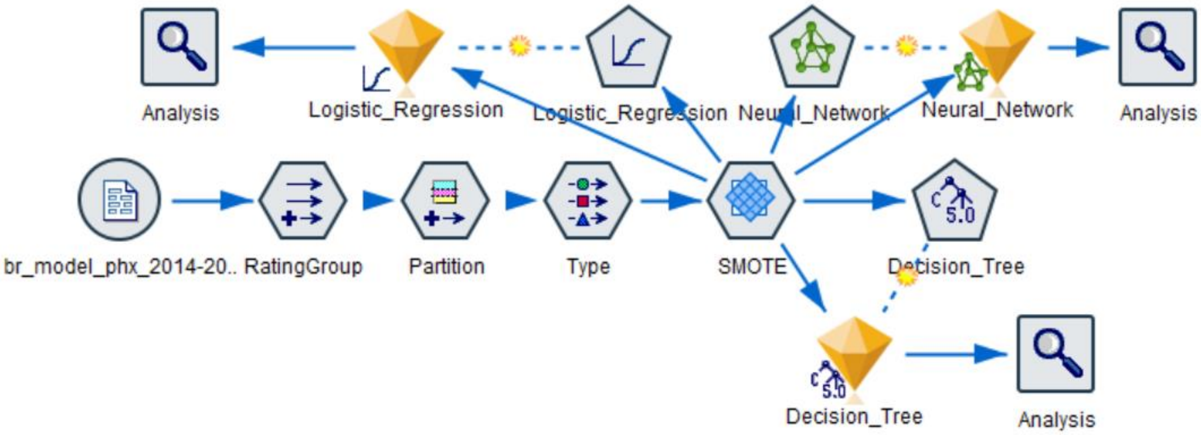


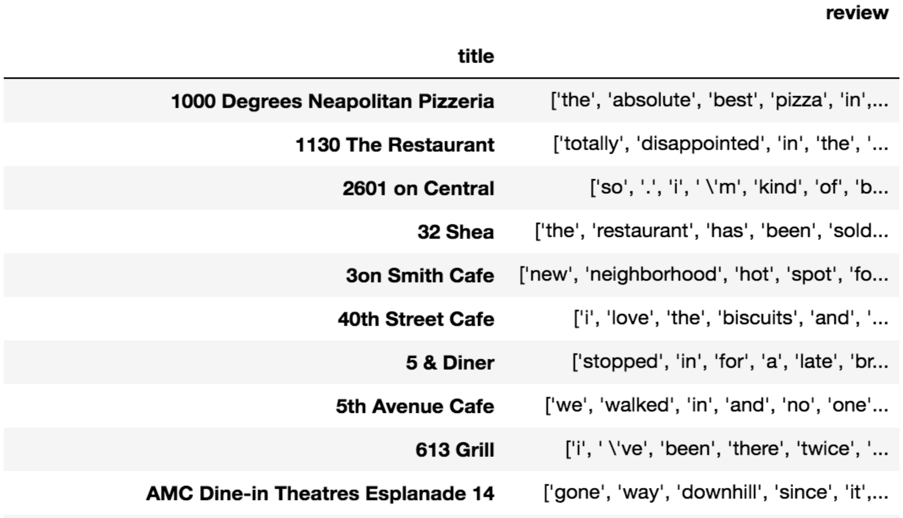
Figure 5. Model Building in SPSS

ii. Content-Based Recommender System

1. Approach 1 - Recommending based on the similarity of item contents

This approach works when customers search a specific restaurant name, it will recommend five similar restaurants to them automatically.

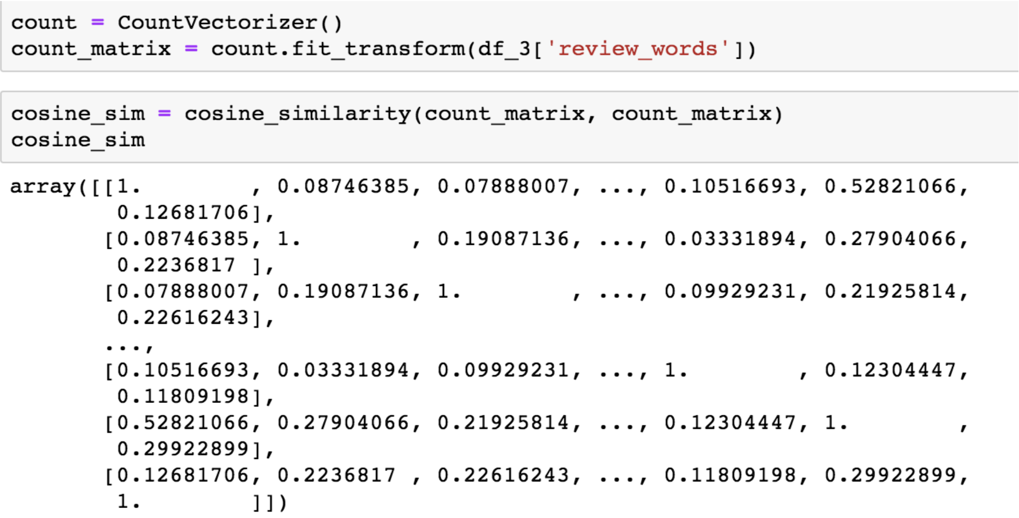
Firstly, all preprocessed review words (after tokenization, stopwords removal, and Porter stemming without selecting reviews with 70-300 words and special years) were grouped by the restaurants' names and put into a new DataFrame.



	review
title	
1000 Degrees Neapolitan Pizzeria	['the', 'absolute', 'best', 'pizza', 'in',...
1130 The Restaurant	['totally', 'disappointed', 'in', 'the', '...
2601 on Central	['so', '.', 'i', ' \'m', 'kind', 'of', 'b...
32 Shea	['the', 'restaurant', 'has', 'been', 'sold...
3on Smith Cafe	['new', 'neighborhood', 'hot', 'spot', 'fo...
40th Street Cafe	['i', 'love', 'the', 'biscuits', 'and', '...
5 & Diner	['stopped', 'in', 'for', 'a', 'late', 'br...
5th Avenue Cafe	['we', 'walked', 'in', 'and', 'no', 'one'...
613 Grill	['i', ' \'ve', 'been', 'there', 'twice', '...
AMC Dine-in Theatres Esplanade 14	['gone', 'way', 'downhill', 'since', 'it',...

Figure 6. Restaurants with All Reviews' Words

Secondly, CountVectorizer was used to vectorize all words' frequency. Since every word is important, each word's frequency was counted. Then the cosine_similarity function was applied to analyze the similarities between restaurants.



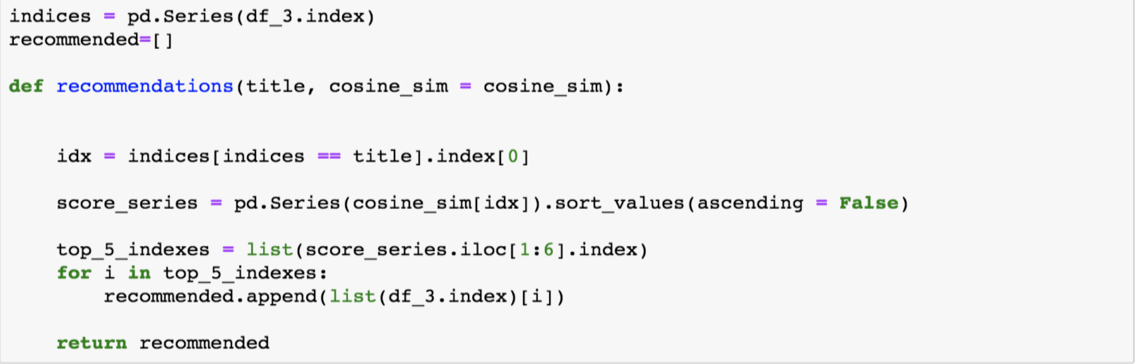
```
count = CountVectorizer()
count_matrix = count.fit_transform(df_3['review_words'])
```

```
cosine_sim = cosine_similarity(count_matrix, count_matrix)
cosine_sim
```

```
array([[1.        , 0.08746385, 0.07888007, ..., 0.10516693, 0.52821066,
        0.12681706],
       [0.08746385, 1.        , 0.19087136, ..., 0.03331894, 0.27904066,
        0.2236817 ],
       [0.07888007, 0.19087136, 1.        , ..., 0.09929231, 0.21925814,
        0.22616243],
       ...,
       [0.10516693, 0.03331894, 0.09929231, ..., 1.        , 0.12304447,
        0.11809198],
       [0.52821066, 0.27904066, 0.21925814, ..., 0.12304447, 1.        ,
        0.29922899],
       [0.12681706, 0.2236817 , 0.22616243, ..., 0.11809198, 0.29922899,
        1.        ]])
```

Figure 7. Similarity Matrix

Finally, the recommender system was built with a function which takes names as input and returns the top 5 similar restaurants. To match the similarity matrix indexes to the actual restaurants' names, a series of names with numerical indexes was also built in the function. And the unit value (1) was deleted, so the input restaurant itself would not be returned.



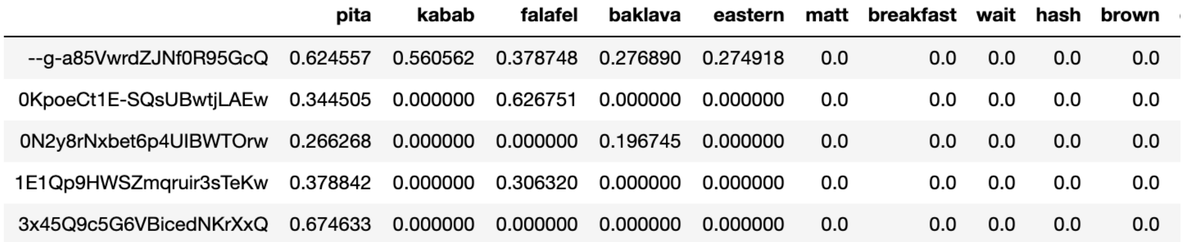
```
indices = pd.Series(df_3.index)
recommended=[]

def recommendations(title, cosine_sim = cosine_sim):


    idx = indices[indices == title].index[0]

    score_series = pd.Series(cosine_sim[idx]).sort_values(ascending = False)

    top_5_indexes = list(score_series.iloc[1:6].index)
    for i in top_5_indexes:
        recommended.append(list(df_3.index)[i])

    return recommended
```

Figure 8. Function of Recommendation

2. Approach 2 - Recommending based on the prediction of the user's rating

This approach recommends restaurants' customers should visit based on which restaurants they have rated and how much they have rated them. As a result, customers' preferences are taken by the recommender system.

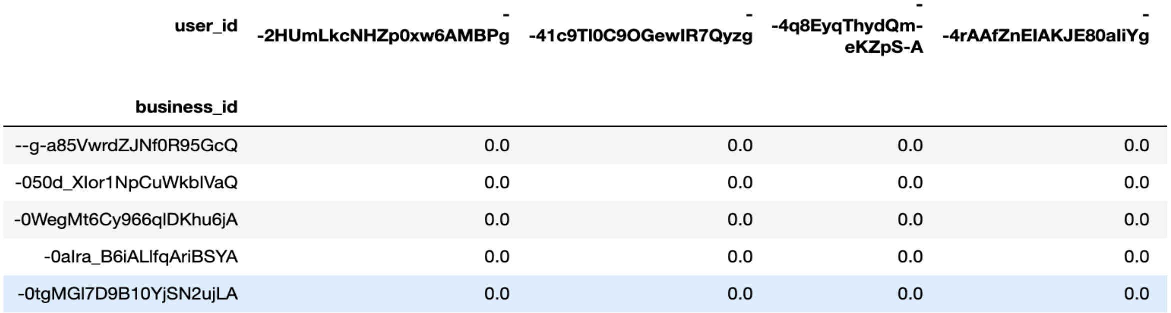
Firstly, all preprocessed review words (after tokenization, stopwords removal, and Porter stemming without selecting reviews with 70-300 words and special years) were used to calculate TFIDF. TFIDF vectors for each restaurant were used to build an item profile. The score of each TFIDF was normalized so that the sum of squared TFIDF scores equals 1 in each restaurant. This item profile has 2993 restaurants and 6744 keywords.



	pita	kabab	falafel	baklava	eastern	matt	breakfast	wait	hash	brown
--g-a85VwrdZJNf0R95GcQ	0.624557	0.560562	0.378748	0.276890	0.274918	0.0	0.0	0.0	0.0	0.0
0KpoeCt1E-SQsUBwtjLAEw	0.344505	0.000000	0.626751	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0
0N2y8rNxbet6p4UIBWTOrw	0.266268	0.000000	0.000000	0.196745	0.000000	0.0	0.0	0.0	0.0	0.0
1E1Qp9HWSZmqruir3sTeKw	0.378842	0.000000	0.306320	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0
3x45Q9c5G6VBicedNKrXxQ	0.674633	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0

Figure 9. Item Profile with TFIDF Vectors for Each Restaurant

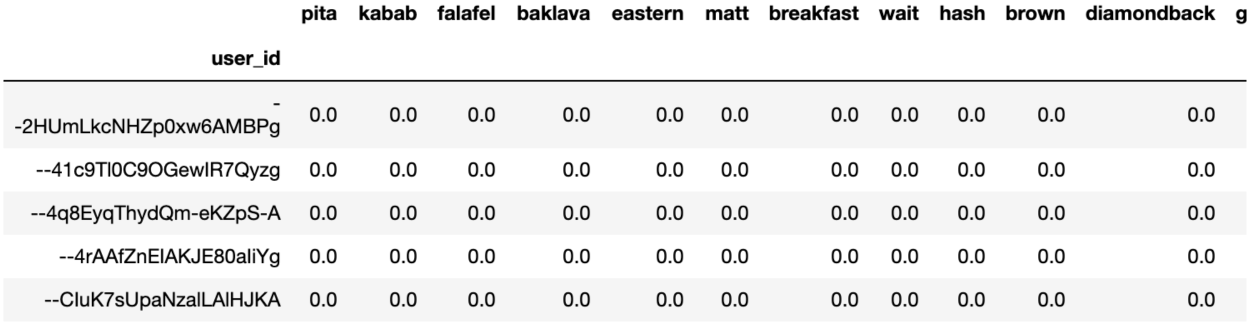
Secondly, a user profile of the ratings was created, which represents how much a user likes or dislikes a restaurant. 5 is for the best while 1 is for the worst. For restaurants that the users have not rated, the rating value is 0. This user profile has 2993 restaurants and 14510 users. Most of the ratings are 0, which are the ratings needed to be predicted.



user_id	--2HUmLkcNHZp0xw6AMBPg	--41c9TI0C9OGewIR7Qyzg	--4q8EyqThydQm-eKZpS-A	--4rAAfZnEIAKJE80aIiYg
business_id				
--g-a85VwrdZJNf0R95GcQ	0.0	0.0	0.0	0.0
-050d_XIor1NpCuWkbIVaQ	0.0	0.0	0.0	0.0
-0WegMt6Cy966qIDKhu6jA	0.0	0.0	0.0	0.0
-0aIra_B6iALIfqAriBSYA	0.0	0.0	0.0	0.0
-0tgMGI7D9B10YjSN2ujLA	0.0	0.0	0.0	0.0

Figure 10. User Profile with Ratings for Each Restaurant

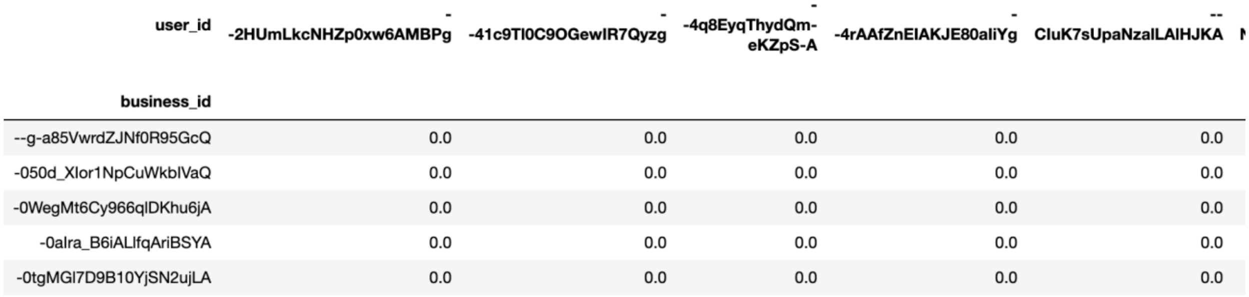
Thirdly, np.matmul in NumPy was used to compute user preference vectors based on the item profile and user profile. The values in this table represent how much a user likes or dislikes a keyword. There are 14510 users and 6744 keywords in this table.



	pita	kabab	falafel	baklava	eastern	matt	breakfast	wait	hash	brown	diamondback	g
user_id												
--2HUmLkcNHZp0xw6AMBPg	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
--41c9Tl0C9OGewIR7Qyzg	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
--4q8EyqThydQm-eKZpS-A	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
--4rAAfZnElAKJE80aliYg	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
--CluK7sUpaNzalLAlHJKA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Figure 11. User Preference Vectors

Finally, np.matmul was used to compute the predicted rating based on the user preference vectors and item profile. For each user, we will recommend the restaurants that have high predicted ratings.



user_id	--2HUmLkcNHZp0xw6AMBPg	--41c9Tl0C9OGewIR7Qyzg	--4q8EyqThydQm-eKZpS-A	--4rAAfZnElAKJE80aliYg	--CluK7sUpaNzalLAlHJKA	N
business_id						
--g-a85VwrdZJNf0R95GcQ	0.0	0.0	0.0	0.0	0.0	
-050d_Xlor1NpCuWkblVaQ	0.0	0.0	0.0	0.0	0.0	
-0WegMt6Cy966qlDKhu6jA	0.0	0.0	0.0	0.0	0.0	
-0alra_B6iALlfqAriBSYA	0.0	0.0	0.0	0.0	0.0	
-0tgMGl7D9B10YjSN2ujLA	0.0	0.0	0.0	0.0	0.0	

Figure 12. Predicted Rating

VII. Results & Evaluation

i. Rating Prediction

Figure 13 shows that sentiment scores of each review had significant relationships with the review rating. The average review rating increases with positive score, and decreases with neutral and negative scores, which means the sentiment of reviews can truly reflect the rating.

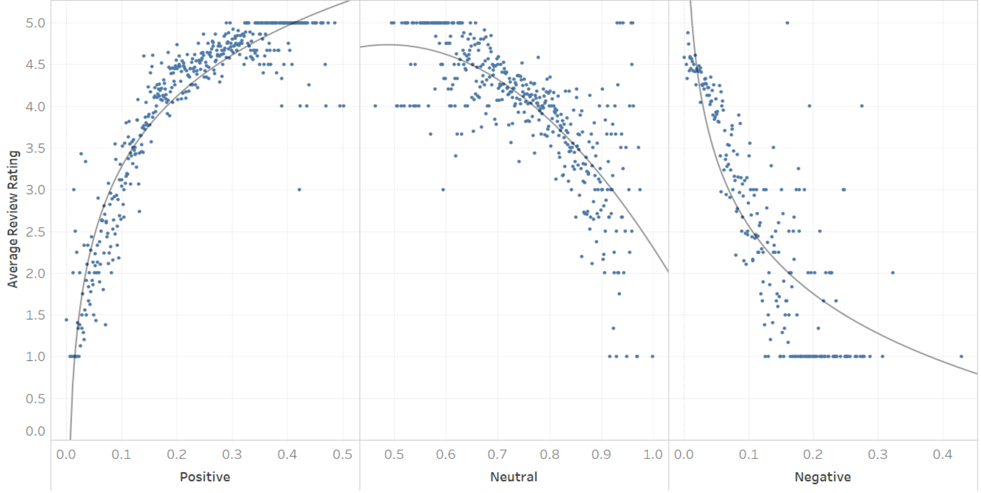
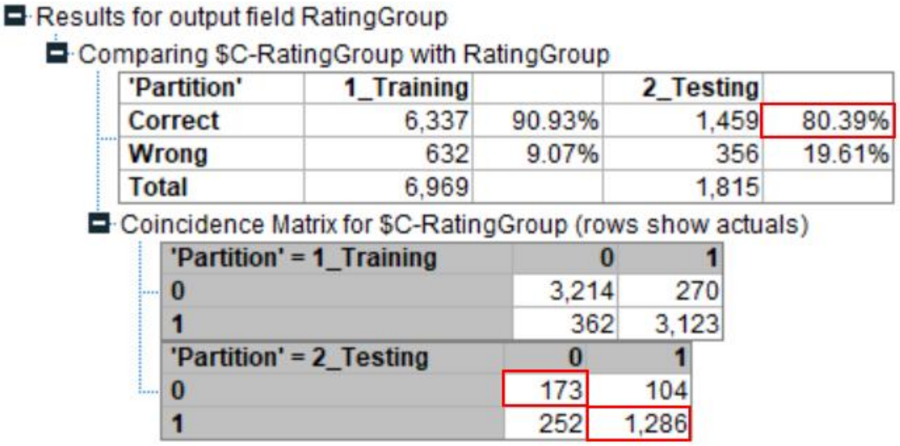


Figure 13. Sentiment Scores VS. Review Rating

Figure 14 shows the performance of the Decision Tree. The accuracy of the testing set is 80.39%. The precision is TP/(TP+FP)=1286/(1286+104)=0.93, and the recall is TP/(TP+FN)=1286/(1286+252)=0.84, and the F1 score is 2*precision*recall/(precision+recall)=0.88. However, the model predicts the high rating group better than the low rating group, which may be due to the imbalance data.



Results for output field RatingGroup

Comparing $C-RatingGroup with RatingGroup

'Partition'	1_Training		2_Testing	
Correct	6,337	90.93%	1,459	80.39%
Wrong	632	9.07%	356	19.61%
Total	6,969		1,815	

Coincidence Matrix for $C-RatingGroup (rows show actuals)

'Partition' = 1_Training	0	1
0	3,214	270
1	362	3,123

'Partition' = 2_Testing	0	1
0	173	104
1	252	1,286

Figure 14. Performance of Decision Tree

Figure 15 shows the most important predictors in this model. Review_count is the most important predictor in this model, followed by the negative score. Meanwhile, among the top 5 important predictors, three of them are topics. Topic 2, 1, and 5 all mention food, time, place, and service.

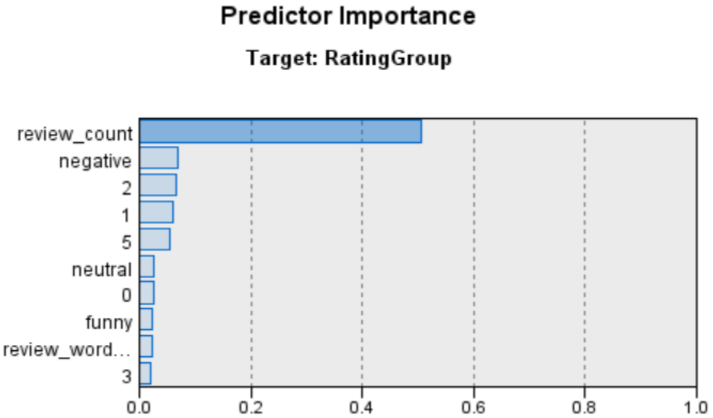
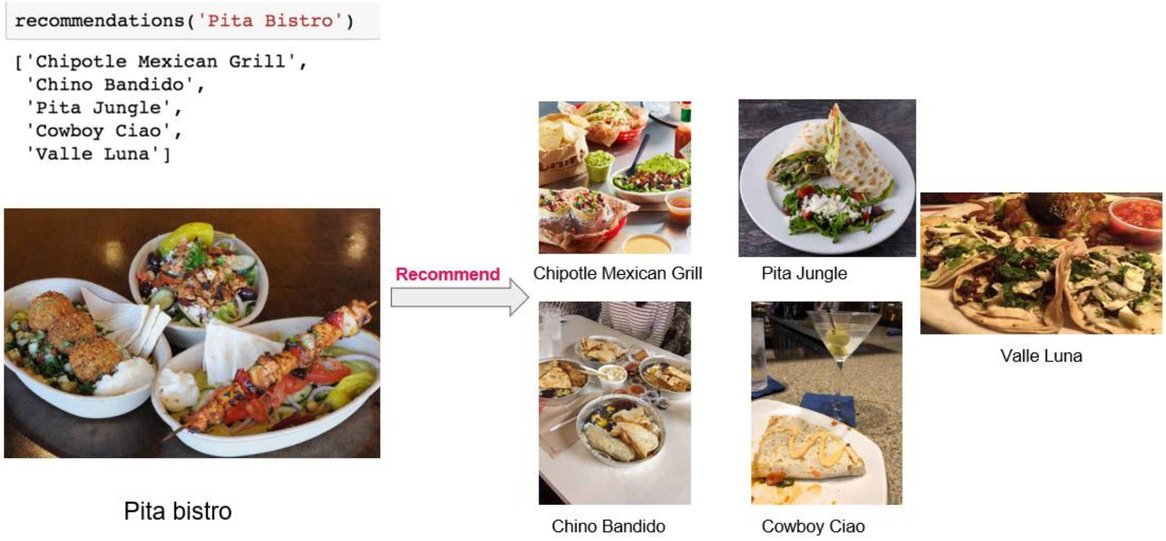


Figure 15. Predictor Importance

ii. Content-Based Recommender System

1. Approach 1 - Recommending based on the similarity of item contents

With the recommender system based on the similarity of item contents, when customers input a specific restaurant's name, top 5 similar restaurants will be recommended to them based on the review words similarities.



```
recommendations('Pita Bistro')

['Chipotle Mexican Grill',
 'Chino Bandido',
 'Pita Jungle',
 'Cowboy Ciao',
 'Valle Luna']
```

Figure 16. Recommendation with Top 5 Similar Restaurants of Pita Bistro

2. Approach 2 - Recommending based on the prediction of user's rating

There are 14510 customers in the dataset before selecting reviews with 70-300 words and special years, but 10842 customers (75%) only have one review.

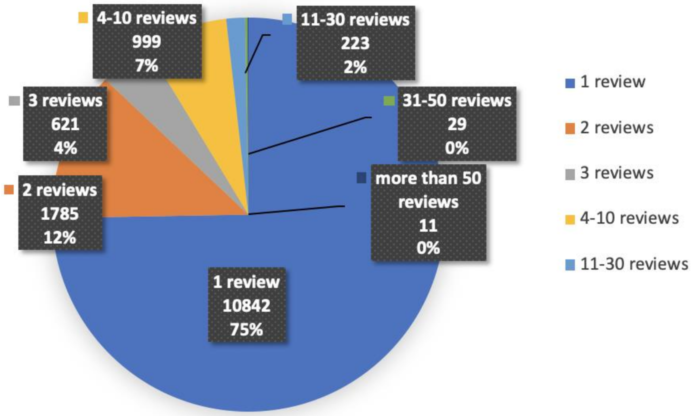


Figure 17. Number of Reviews in User Profile

We defined that when the predicted rating is over 3, the restaurant can be recommended to the customer. Among these 10842 customers, this recommender system makes recommendations to only 21.5% of them. However, as the user provides more inputs, the engine becomes more and more effective. When the customers have more than 10 reviews, this recommender system can find and recommend restaurants to almost everyone.

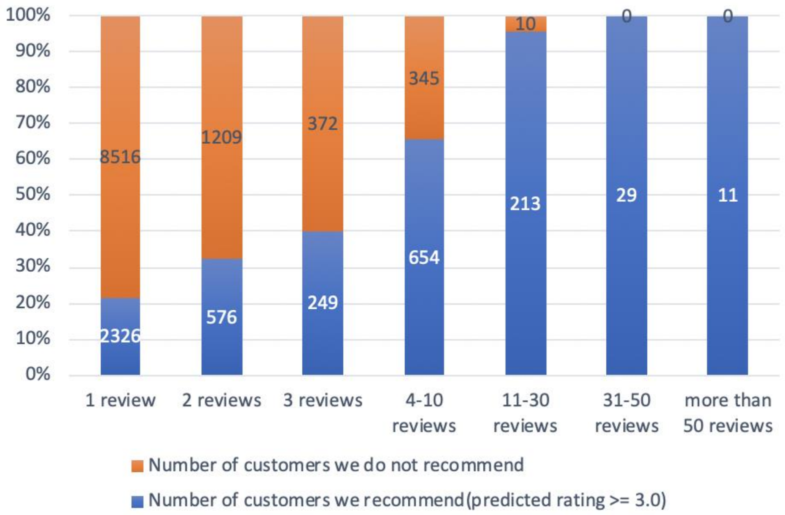


Figure 18. Number of Customers We Recommend

Next, we evaluated the restaurants that are recommended to two users. The first user Taylor has 80 records in our user profile. From the line chart of the number of reviews she has given to restaurants, we can see that she often writes reviews for Fast Food and American restaurants, but her average rating of fast food is only about 3.5. The average ratings show that Taylor loves Vietnamese, Thai, Steakhouse, and Japanese foods most.

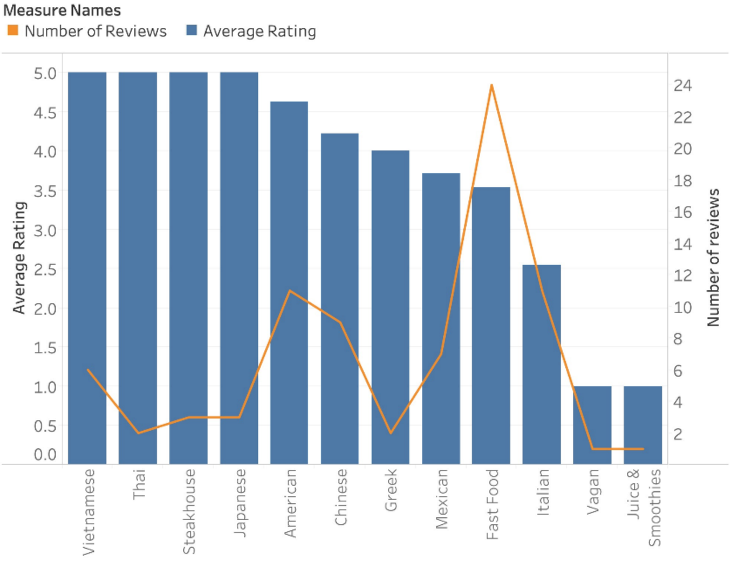
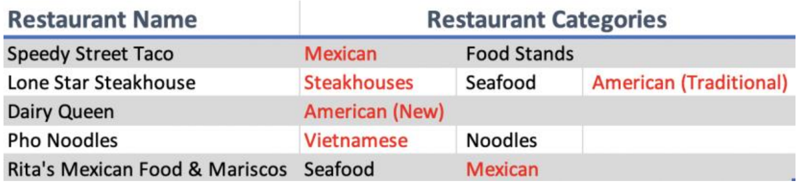


Figure 19. Taylor’s Average Rating vs. Number of Reviews in Each Category

According to the predicted rating, our recommender system recommended 5 restaurants to Taylor as below. The categories of these restaurants all belong to the categories that Taylor has been to and rated with scores. From this perspective, Tyler might love these recommended restaurants.



Restaurant Name	Restaurant Categories		
Speedy Street Taco	Mexican	Food Stands	
Lone Star Steakhouse	Steakhouses	Seafood	American (Traditional)
Dairy Queen	American (New)		
Pho Noodles	Vietnamese	Noodles	
Rita's Mexican Food & Mariscos	Seafood	Mexican	

Figure 20. Restaurants Recommended to Taylor

The second user is Cris. We have 28 reviews of him in our user profile. According to the line chart of Cris's records, he often writes reviews for Bar and Italian restaurants. Additionally, he had high average ratings of Mexican food, but he did not have good experience at Japanese and Fast food restaurants.

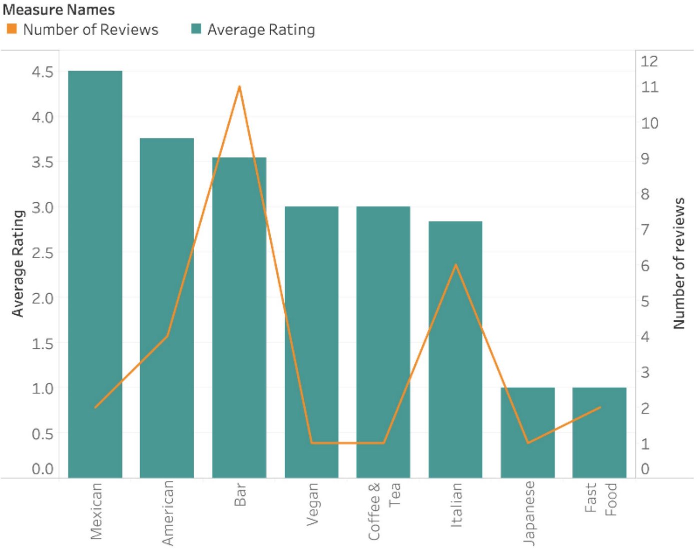
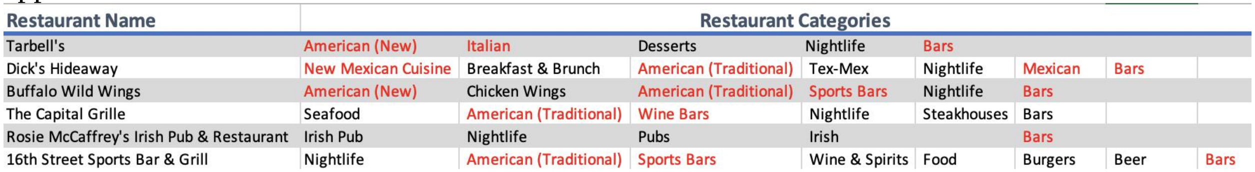


Figure 21. Cris's Average Rating vs. Number of Reviews in Each Category

We can see that the restaurants we recommend to Cris are all relevant to the restaurants that he has been to. In our recommendations, most of the restaurants are bars, which match Cris's appetite.



Restaurant Name	Restaurant Categories							
Tarbell's	American (New)	Italian	Desserts	Nightlife	Bars			
Dick's Hideaway	New Mexican Cuisine	Breakfast & Brunch	American (Traditional)	Tex-Mex	Nightlife	Mexican	Bars	
Buffalo Wild Wings	American (New)	Chicken Wings	American (Traditional)	Sports Bars	Nightlife	Bars		
The Capital Grille	Seafood	American (Traditional)	Wine Bars	Nightlife	Steakhouses	Bars		
Rosie McCaffrey's Irish Pub & Restaurant	Irish Pub	Nightlife	Pubs	Irish		Bars		
16th Street Sports Bar & Grill	Nightlife	American (Traditional)	Sports Bars	Wine & Spirits	Food	Burgers	Beer	Bars

Figure 22. Restaurants Recommended to Cris

VIII. Conclusions

1. The customers' sentiment in reviews is related to the rating. Positive score shows positive correlation, neutral and negative scores show negative correlation.

2. Topics about food, time, place, and service in the reviews are related to the rating.

3. The similarity between reviews of customers and restaurants, and the similarity among reviews of restaurants can help customers find more related restaurants.

4. Both two methods can build a useful content-based recommender system using the text data extracted from the reviews.

5. For customers who do not leave enough ratings to help the system understand their preference or new users, the recommender system which only analyzes the review text (approach 1) could be used to recommend them restaurants just based on the similarity of each restaurant's reviews.

6. For customers who have enough records in the user profile, the recommender system which creates a user profile and an item profile from user rated content (approach 2) should be used, since it takes user's preference into consideration.

IX. Recommendations

i. For Users

Since the reviw_count is the most important predictor for the restaurant rating, customers are recommended to write reviews about their opinions, which will contribute to not only the restaurant's improvement, but also the customers' future experience.

ii. For Restaurants

1. Negative score is the second important predictor, from the model details, we can see that the higher the negative score, the more likely the restaurants to be predicted as the low rating group. As a result, restaurants should spare no effort to solve all negative feedback from customers to improve their rating.

2. Topic 2, 1, and 5 all mention food, time, place, and service. Based on the model details, the more related to these three topics, the more likely a restaurant to belong to the low rating group. Therefore, those restaurants who get the low rating should pay more attention to these aspects and improve them.

X. References

[1] https://github.com/karantyagi/Restaurant-Recommendations-with-Yelp

[2] https://monkeylearn.com/text-mining/

[3]https://towardsdatascience.com/how-to-build-from-scratch-a-content-based-movie-recommender-with-natural-language-processing-25ad400eb243

[4]https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/