Review Sentiment Analysis & Recommendation System

Group 3: Weiwei Wan | Harika Rallapalli
Ranran Shi | Xingzi Sun

Abstract

As the quick development of the internet, there has been a rapid increase in the number of online businesses. People freely express their views and comments on online businesses, and at the same time their potential choices may also be influenced by others. On one hand, the review sentiment analysis with high accuracy could help online businesses to understand their customers' opinion and get timely feedback about their products. We proposed various machine learning classifiers (such as random forest, neural networks, gradient boosting) with different feature extraction methods classified the Yelp review sentiment. The best classifier neural networks with hasingvetorizer could achieve 91.5% accuracy for sentiment classification. On the other hand, recommendation systems were highly demanded by both users and businesses. We developed 4 recommender systems (simple, content-Based, collaborative filtering, hybrid) recommended businesses on Yelp to users based on their personal interests and biases, which could improve customer's experience as well as the business advertising.

1. Introduction

Yelp provides a business directory, publishes crowd-sourced reviews about local businesses, and people can make online reservation services through Yelp. People could search on Yelp for gathering information regarding restaurants, moving companies, hospitals and so on. Yelp grew in usage and in 2019 it reported having a monthly average of 76.7 million unique visitors using its APP [3]. Therefore, an effective and efficient recommendation to customers is crucial for optimizing user experience, as well as improving business advertising. Furthermore, plentiful users comment on business and share their opinions with others on Yelp. Accurate classification of user-review sentiments would help the online businesses to understand customer' preferences and get timely feedback. This valuable sentiment information would also help enhance the recommendation system.

In this project, we explored the Yelp restaurant business dataset and built below two systems: Yelp business recommendation system and review sentiment analysis system. Generally, this project is a combination of a practicing work of data filtering and natural language processing. On one hand, several machine learning classifiers were built with different feature extraction methods to classify the review sentiment. By exploring different algorithms and feature extraction approaches, this study tried to find out the best classifier with high accuracy. On the other hand, four recommendation engines were built based on weighted rating, content, collaborative filtering, and hybrid methods.

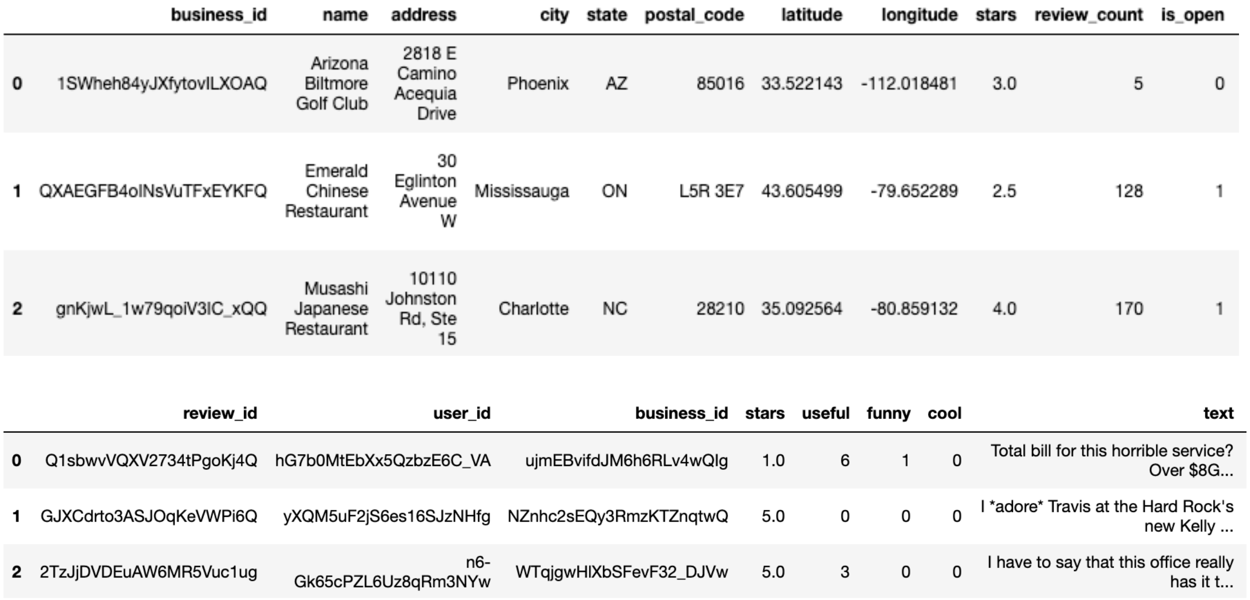
2. Literature Reviews

There is a huge explosion today of sentiments available from social media including Twitter, Facebook, Amazon, Yelp. Ronen Feldman [5] reviewed some of the main research problems within the field of sentiment analysis and discussed several algorithms that aim to solve each of these problems. He also described the basic processes and some major applications of sentiment analysis. It is a good review paper for beginners to understand this area. Andreea et. al. [6] proposed several approaches for automatic sentiment classification, using two feature extraction methods and four machine learning models (Naive Bayes, Support Vector Machine, Logistic Regression and Stochastic Gradient Descent). Their best classifiers Linear SVC and SGD have obtained an accuracy of 94.4%. In the paper [4], Basilico J and Hofmann T. have built a sentiment model to predict a simple positive or negative evaluation on the part of the customer's review and employed methods to determine the rating ranging from 1 to 5 stars.

Recommender systems have been widely used in online businesses, such as Amazon, Netflix, and Yelp, which help users discover new and relevant items (products, videos, jobs, music), creating a delightful user experience while driving incremental revenue. The recommendation systems based on collaborative filtering are widely used by online business, which predicts a user's interest in some item on the basis of the scores generated and the correlation calculated between the users. U. Farooque et al. proposed a basic structure and steps of designing a recommender system that uses collaborative filtering (user-based) , and designed a Restaurant Recommender System. N. Jonnalagedda and S. Gauch [8] developed a hybrid personalized news recommender system that can recommend interesting new articles based on user's profile settings. L. Zhang et at [11] have proposed an assortment optimization method: Assortment Balancing for Two-Stage collaborative filtering (DBTS), to enhance the robustness of the recommendation system by using social relationship filtering connected with collaborative filtering. We could adopt the above approaches to build our own Yelp recommendation systems.

3. Data Preprocessing

We obtained the dataset from an open source website at Kaggle: Yelp Dataset. The original datasets file contains 6 files, 8GB in total. This dataset is Yelp businesses, reviews, and user data. In total, there are 5,200,000 user reviews, information on 174,000 businesses, and the data span of 1200 cities. The original JSON files were converted to CSV files for further analysis.
Our project involved two datasets: "business" and "review", as shown in Figure 1. The business dataset contains information for each business, including business id, name, address, categories, star ratings, etc. The review dataset contains information for each review: review id, user id, business id, review text, rating, date, and tag counts for "cool", "funny", and "useful". Yelp uses a 5-star rating system by which users can review local business on a 5-star scale, where 5 is for the best and 1 is for the worst.



	business_id	name	address	city	state	postal_code	latitude	longitude	stars	review_count	is_open
0	1SWheh84yJXfytovILXOAQ	Arizona Biltmore Golf Club	2818 E Camino Acequia Drive	Phoenix	AZ	85016	33.522143	-112.018481	3.0	5	0
1	QXAEGFB4oINsVuTFxEYKFQ	Emerald Chinese Restaurant	30 Eglinton Avenue W	Mississauga	ON	L5R 3E7	43.605499	-79.652289	2.5	128	1
2	gnKjwL_1w79qoiV3IC_xQQ	Musashi Japanese Restaurant	10110 Johnston Rd, Ste 15	Charlotte	NC	28210	35.092564	-80.859132	4.0	170	1

	review_id	user_id	business_id	stars	useful	funny	cool	text
0	Q1sbwvVQXV2734tPgoKj4Q	hG7b0MtEbXx5QzbzE6C_VA	ujmEBvifdJM6h6RLv4wQIg	1.0	6	1	0	Total bill for this horrible service? Over $8G...
1	GJXCdrto3ASJOqKeVWPi6Q	yXQM5uF2jS6es16SJzNHfg	NZnhc2sEQy3RmzKTZnqtwQ	5.0	0	0	0	I *adore* Travis at the Hard Rock's new Kelly ...
2	2TzJjDVDEuAW6MR5Vuc1ug	n6-Gk65cPZL6Uz8qRm3NYw	WTqjgwHlXbSFevF32_DJVw	5.0	3	0	0	I have to say that this office really has it t...

Figure 1. The snapshots of the business (top) and review (bottom) datasets.

The data was cleaned by filling missing text values with empty string and dropping other missing values. The useless attributes such as “is_open”, “hours”, “date”, and review tags were dropped as they are not used in our analysis. The original user_id and business_id is difficult to read, thus they were encoded with integers. A new column denoting price range was created by extracting price information in the “attributes” column. For the purpose of the collaborative recommendation, we created new data frames that merged business information and review information on business id.

4. Exploration Data Analysis

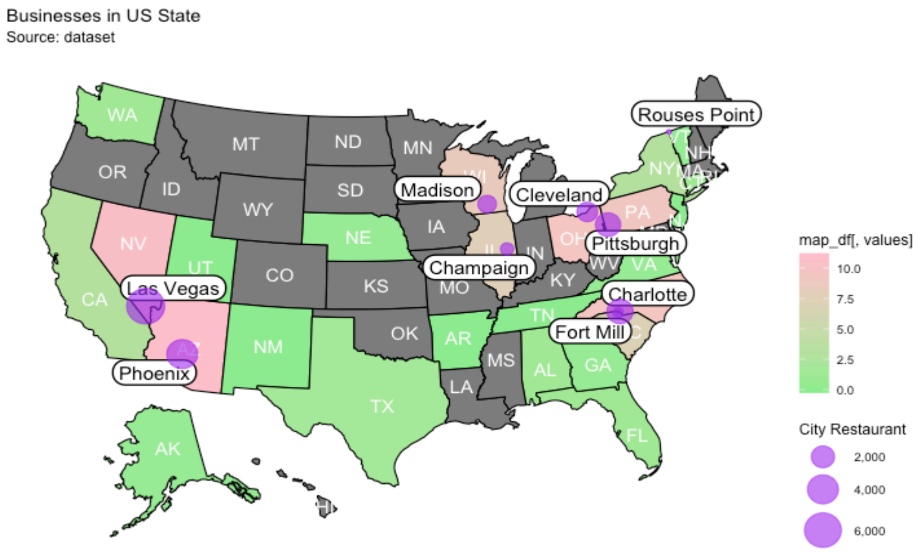


Figure 2. The business distribution in the United State.

Figure 2 shows the businesses distribution (with natural logarithm transformation) in the United State. There were 141,900 businesses in 818 cities of 25 states (colored). The states with pink/light green colors had a large/small number of businesses. There was no business data in the states with the gray color. The states AZ and NV had the greatest number of businesses (56686 and 36312, respectively).
Figure 3 shows the most popular business categories in Yelp, such as "Restaurants", "Shopping", "food", "Home Services", "Beauty Spas", "Home services", and "Health Medical". We can see that the businesses on Yelp covered most aspects of our daily life, and Yelp could facilitate our lives.

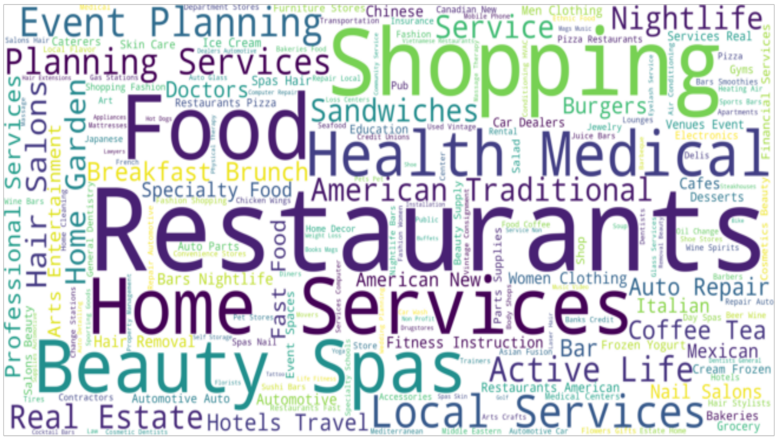


Figure 3. Popular business categories.

In this project, we focused on the top popular business, i.e., restaurants. The distribution of restaurants in 9 cities are indicated by the purple circles in Figure 2. The city of Las Vegas had the greatest number of restaurants (6450), followed by the city of Phoenix (3999).

Figure 4 shows the reviews distribution regarding the restaurants in different cities. The city of Las Vegas had the largest number of reviews (1204772).

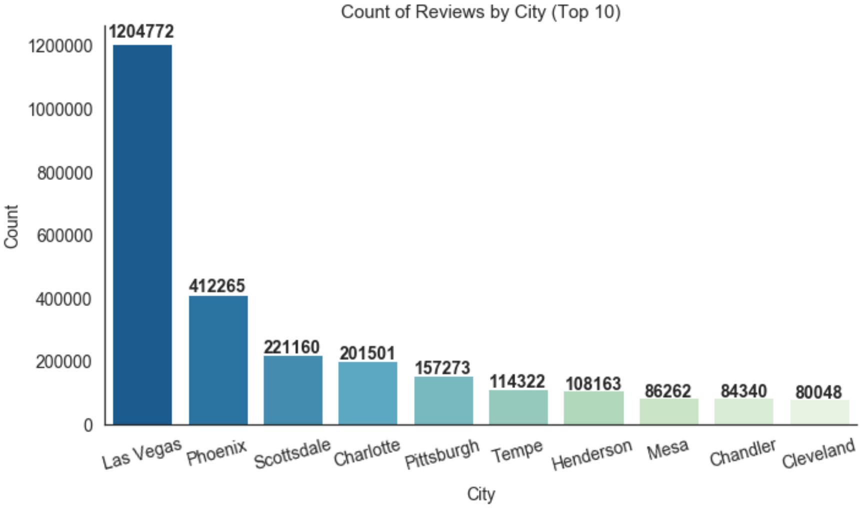


Figure 4. The total number of reviews for different cities.

The star ratings in Yelp reviews had 5 levels. As shown in Figure 5, most of the restaurant reviews had star ratings from 3 to 4. Only a few restaurants had star ratings of 1 or 5.

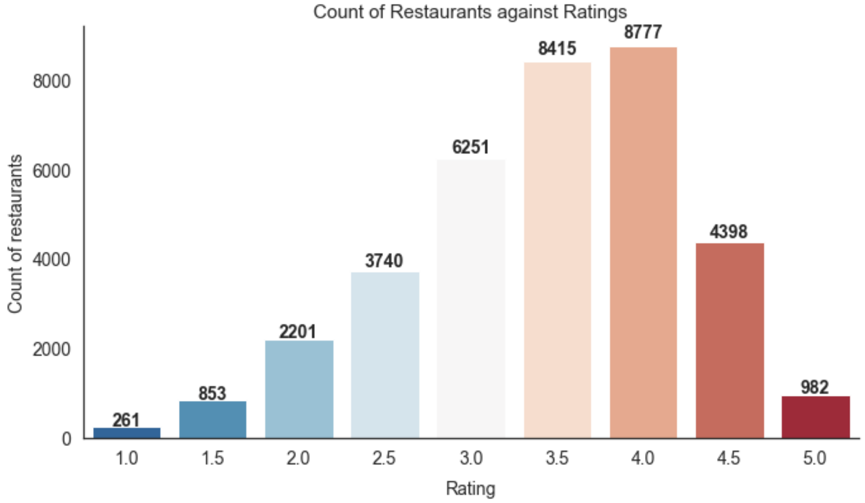


Figure 5. The distribution of star ratings.

The popularity of cuisine types in the United State was also explored in Figure 6. The most popular cuisine is American, followed by Mexican, Italian, Chinese, Japanese food.

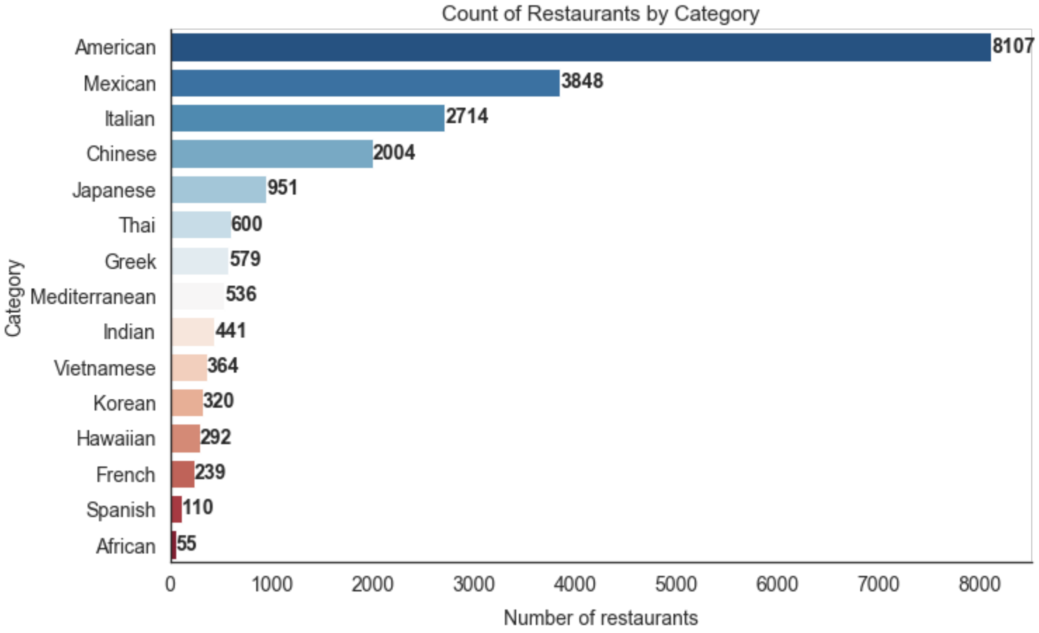


Figure 6. The most popular cuisine types.

5. Methodology

5.1 Review Sentiment Analysis

Review sentiment analysis is text mining, which is a subcategory of natural language processing. In order to identify sentiment in reviews, we divided the tasks into two parts: feature extraction and classifiers building.

5.1.1 Feature Extraction

Review data is a text data. It cannot be used directly since machine learning models can process only numerical data. Therefore, the text data should be converted to numbers.

The review text is treated as a bag of words model, in which a text is represented as the bag of its words, disregarding grammar and even word order but keeping the occurrence of words in a document. After tokenization preprocessing (removing stop words, stemming), the words were encoded into numeric vectors by using three different vectorizers. This process is called feature extraction (or vectorization).

Word Counts with CountVectorizer

CountVectorizer counts the number of times a word appears in each review and builds a count matrix. Each row of the count matrix represents a review, and the columns (features) are the unique words that appear in all the reviews.

Word Frequencies with TfidfVectorizer

One limitation of CountVectorizer is that some words like "*the*", "a" will appear many times and there is no meaning for their large counts in the encoded vectors. An alternative is to use word frequency inverse document frequency (TFIDF). TfidfVectorizer weights the word counts by a measure of how often they appear in the documents. A count matrix could be transformed to a TFIDF matrix by applying the below formula.

$$w_{ik} = ft_{ik} * \log\left(\frac{N}{n_k}\right)$$

$T_k = term\ k\ in\ document\ D_i$

$tf_{ik} = frequency\ of\ term\ T_k in\ document\ D_i$

$idf_k = inverse\ document\ frequenct\ of\ term\ Tk\ in\ C$

N=total number of documents in the Collection C

$n_k = the\ number\ of\ documents\ in\ C\ that\ contain\ T_k$

$idf_k = \log\left(\frac{N}{n_k}\right)$

Hashing with HashingVectorizer

CountVectorizer and **TfidfVectorizer** can be very useful. However, as the number of reviews increases, the vocabulary and the encoded vectors could become very large. The large data size with increasing instances and features would result in large requirements on memory and hence slow down the algorithms. HashingVectorizer was designed to be memory efficient, which did not compute a dictionary mapping terms but used hash functions to hash each term. The HashingVectorizer could fix the feature size (we fix it to 1000) and hence reduce the data dimension. The downside of this method is that it is a one-way vectorizer, the original features can no longer be retrieved (which may not matter for many machine learning tasks).

5.1.2 Build classifiers.

Review sentiment analysis is actually a classification problem. We labeled whether a review was positive or negative based on the star ratings. Initially, all the reviews with star ratings >=4 were labelled as positive, and other reviews with star ratings <=3 were labelled as negative.
The dataset was split into a training set and a test set with a ratio of 70%/30%. The training set was used to fit the models, and the test set was used to evaluate the model performance with respect to accuracy, recall, precision, F1 score.
For the binary classification, we built 6 machine learning classifiers based on Decision Tree, Random Forest (RF), Gradient Boosting, Ada Boost, Neural Networks, and Support Vector Machine (SVM) algorithms.

5.2 Recommendation Engines

Four recommendation engines were built in this project: simple recommendations based on users' input, content-based systems, collaborative filtering systems, and hybrid systems (which use a combination of the other two).

Simply Recommendation
Simply Recommendation makes recommendations based on the interests of users, such as food type, food price, food rating. A good restaurant should not only have a high star rating but also sufficient rating counts. For an extreme example, a restaurant with only 1 review would not be recommended to users even though it had a 5-star rating. The minimum votes required for recommendation could be determined by the default or user-define percentiles. The weighted rating formula is shown as below [2]:

$$\left(\frac{v}{v+m} * R\right) + (\frac{m}{v+m} * C)$$

where,

- v is the number of votes for the restaurant
- m is the minimum votes required to be listed in the chart

- R is the average rating of the restaurant
- C is the mean vote across the whole report

Content filtering approach

Content filtering approach recommends similar items to users based on their preference (such as user inputs). As shown in Figure 7(a), assuming item coca cola is similar to pizza. If a user likes coca cola, the content filtering approach would recommend him/her the pizza.

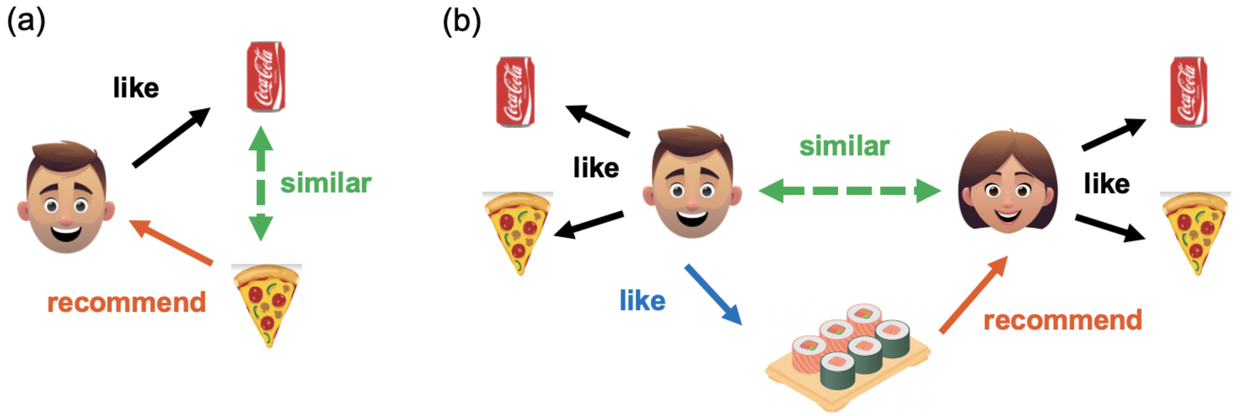


Figure 7. (a) Content filtering approach. (b) Collaborative filtering approach

Cosine similarity is the most common similarity metric, which calculates similarity by measuring the cosine of angle between two vectors. This is calculated as:

$$\text{Similarity}=\cos(\theta) = \frac{A*B}{||A||*||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}*\sqrt{\sum_{i=1}^{n} B_i^2}}$$

In this project, the similarity between businesses was calculated based on the “categories”. The words in “categories” were converted into numeric vectors by using a bag of words with TF-IDF.

Collaborative filtering approach

Content filtering approach requires a good amount of information about items' own features. On the other hand, collaborative filtering doesn’t need anything except users’ interactions and feedback, as shown in Figure 7(b). Assuming both users A and B like coca cola and pizza (i.e., A and B are similar), if user A also likes sushi, user B is expected to like sushi too, and the collaborative filtering approach would recommend sushi to B.

Collaborative filtering approach uses the user-item rating matrix to calculate similarity between users or items. There are two kinds of Collaborative filtering. 1, user-based**:** find similar users who have rated the item, using the weighted users’ rating to predict the rating. 2, item-based: find similar items rated by the same user. Using the weighted items’ rating to predict the rating.

Usually, the user-item rating matrix is very sparse, and Dimensionality Reduction (such as Matrix factorization, PCA) can improve the performance of the algorithm in terms of both space and time.

In this project, matrix factorization has been used to reduce the dimension by decomposing the user-item interaction matrix into the product of two lower dimensionality matrices (user matrix and item matrix). We used a popular algorithm of Singular Value Decomposition (SVD) to factorize the user-item matrix.

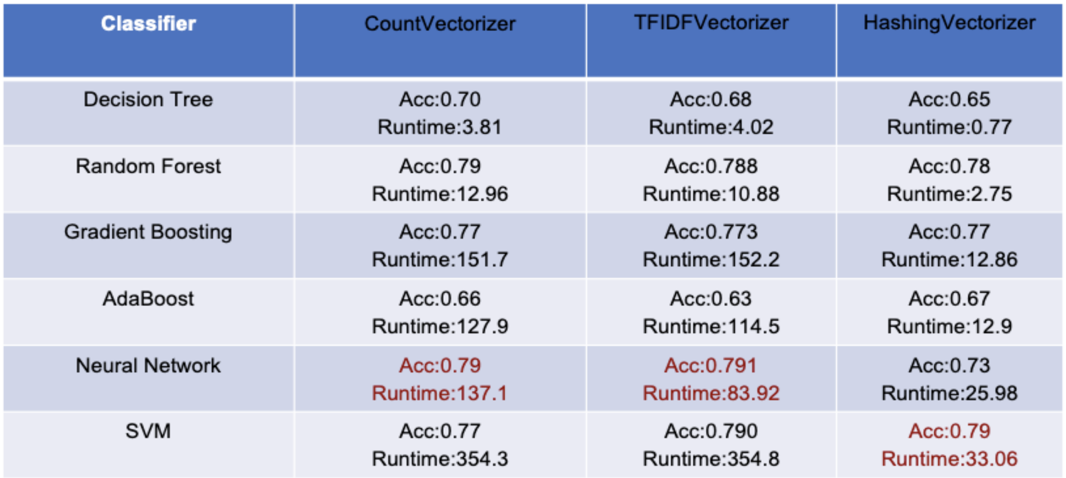
Hybrid approach

Hybrid approach brings together techniques we have implemented in the content based and collaborative filter-based engines. Compared to other base recommendation systems, the hybrid approach combines two or more recommendation techniques to gain performance with fewer of the drawbacks. Hybrid recommendation engines could recommend personalized items to different users.

6. Results and discussion

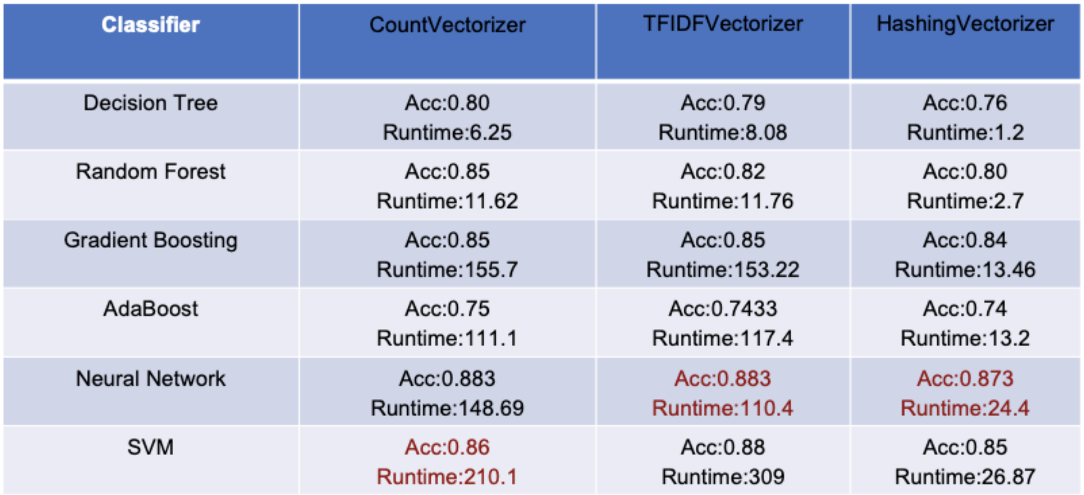
6.1 Review Sentiment Analysis

The star ratings have 5 levels of stars. We expected that the sentiment of the reviews with 3 stars were neutral, and the model performance could improve without considering the neutral reviews. In order to check our assumption, two different methods were used to label the review sentiment. The first method included the neutral reviews, and all the reviews with star rating <=3 were considered as negative reviews, while others were considered as positive reviews. The second method dropped all the neutral reviews. The reviews with star ratings <3 are labelled as negative reviews, and those with star ratings >3 were labelled as positive reviews.



Classifier	CountVectorizer	TFIDFVectorizer	HashingVectorizer
Decision Tree	Acc:0.70 Runtime:3.81	Acc:0.68 Runtime:4.02	Acc:0.65 Runtime:0.77
Random Forest	Acc:0.79 Runtime:12.96	Acc:0.788 Runtime:10.88	Acc:0.78 Runtime:2.75
Gradient Boosting	Acc:0.77 Runtime:151.7	Acc:0.773 Runtime:152.2	Acc:0.77 Runtime:12.86
AdaBoost	Acc:0.66 Runtime:127.9	Acc:0.63 Runtime:114.5	Acc:0.67 Runtime:12.9
Neural Network	Acc:0.79 Runtime:137.1	Acc:0.791 Runtime:83.92	Acc:0.73 Runtime:25.98
SVM	Acc:0.77 Runtime:354.3	Acc:0.790 Runtime:354.8	Acc:0.79 Runtime:33.06

Figure 8. The model performance with the neutral reviews. The total number of reviews is 2000.

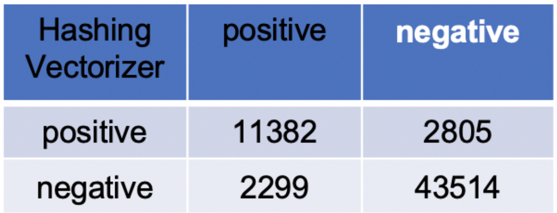


Classifier	CountVectorizer	TFIDFVectorizer	HashingVectorizer
Decision Tree	Acc:0.80 Runtime:6.25	Acc:0.79 Runtime:8.08	Acc:0.76 Runtime:1.2
Random Forest	Acc:0.85 Runtime:11.62	Acc:0.82 Runtime:11.76	Acc:0.80 Runtime:2.7
Gradient Boosting	Acc:0.85 Runtime:155.7	Acc:0.85 Runtime:153.22	Acc:0.84 Runtime:13.46
AdaBoost	Acc:0.75 Runtime:111.1	Acc:0.7433 Runtime:117.4	Acc:0.74 Runtime:13.2
Neural Network	Acc:0.883 Runtime:148.69	Acc:0.883 Runtime:110.4	Acc:0.873 Runtime:24.4
SVM	Acc:0.86 Runtime:210.1	Acc:0.88 Runtime:309	Acc:0.85 Runtime:26.87

Figure 9. The model performance without the neutral reviews. The total number of reviews is 2000.

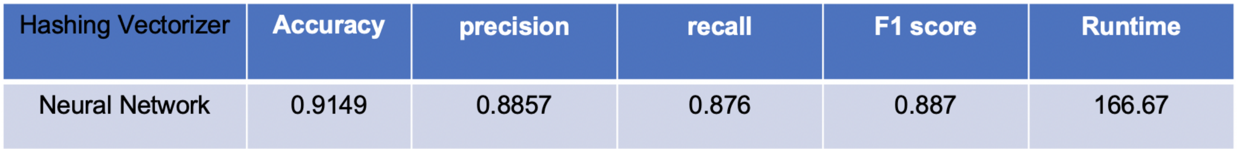
Figure 8 and Figure 9 show the model performance with and without the neutral reviews, respectively. The accuracy of all the six classifiers was improved about 0.1 if the neutral reviews were dropped, indicating that our expectation was true.

As shown in Figure 9, the neural networks, SVM, and gradient boosting have similar high accuracy. However, the neural network has the smallest run time for the 5-folder cross validation. Thus, in terms of both accuracy and runtime, the neural network is the best classifier. It is also noted that, compared to CountVectorizer and TfidfVectorizer, the HashingVectorizer has similar accuracy but a significantly smaller run time. Thus, HashingVectorizer is the best feature extraction approach. Next, the neural networks with HashingVectorizer were used to analyze 200,000 reviews. The confusion matrix and performance metric are shown in Figure 10 and 11. Our best classifier neural network model with the hashing vectorizer can handle 200000 review texts, which takes only 166 seconds for 5-folder cross validation using our personal laptop. The best neural network classifier obtained a high accuracy of 91.5%.



Hashing Vectorizer	positive	negative
positive	11382	2805
negative	2299	43514

Figure 10. Confusion Matrix.



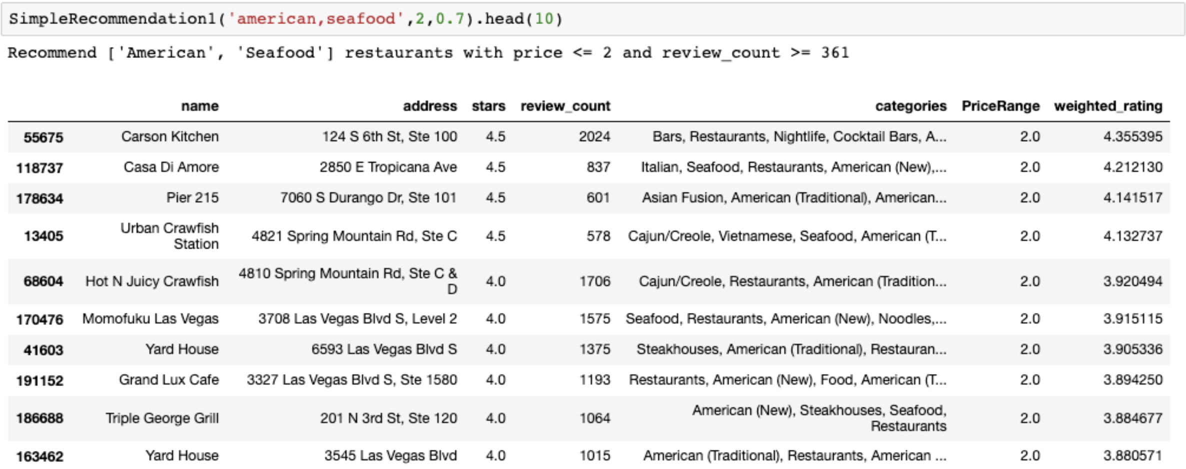
Hashing Vectorizer	Accuracy	precision	recall	F1 score	Runtime
Neural Network	0.9149	0.8857	0.876	0.887	166.67

Figure 11. Performance metric of the best model.

6.2 Recommendation System

Simple recommendation

Figure 12 shows the top 10 recommended restaurants in Las Vegas based on users' input "American, Seafood" and with the price range <=2. The restaurants are listed in the descending order of the weighted rating. All the recommended restaurants have large star ratings and review counts. Their "categories" contain the words of "American" and "Seafood".



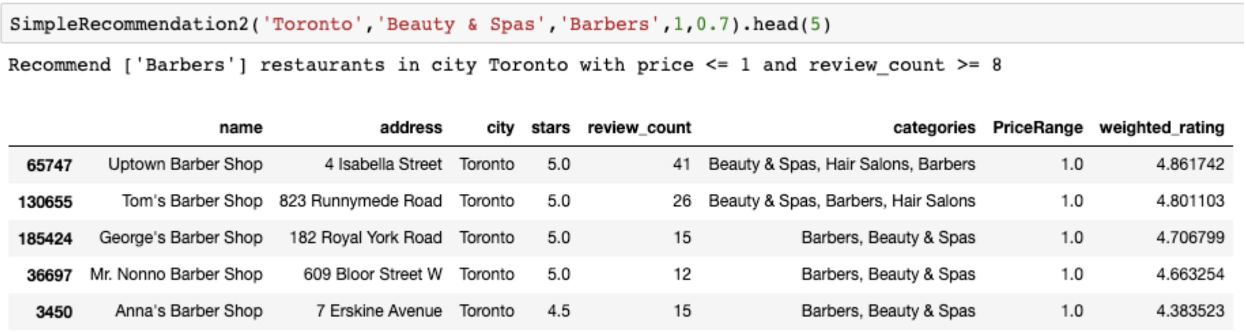
```
SimpleRecommendation1('american,seafood',2,0.7).head(10)
```

```
Recommend ['American', 'Seafood'] restaurants with price <= 2 and review_count >= 361
```

	name	address	stars	review_count	categories	PriceRange	weighted_rating
55675	Carson Kitchen	124 S 6th St, Ste 100	4.5	2024	Bars, Restaurants, Nightlife, Cocktail Bars, A...	2.0	4.355395
118737	Casa Di Amore	2850 E Tropicana Ave	4.5	837	Italian, Seafood, Restaurants, American (New),...	2.0	4.212130
178634	Pier 215	7060 S Durango Dr, Ste 101	4.5	601	Asian Fusion, American (Traditional), American...	2.0	4.141517
13405	Urban Crawfish Station	4821 Spring Mountain Rd, Ste C	4.5	578	Cajun/Creole, Vietnamese, Seafood, American (T...	2.0	4.132737
68604	Hot N Juicy Crawfish	4810 Spring Mountain Rd, Ste C & D	4.0	1706	Cajun/Creole, Restaurants, American (Tradition...	2.0	3.920494
170476	Momofuku Las Vegas	3708 Las Vegas Blvd S, Level 2	4.0	1575	Seafood, Restaurants, American (New), Noodles,...	2.0	3.915115
41603	Yard House	6593 Las Vegas Blvd S	4.0	1375	Steakhouses, American (Traditional), Restauran...	2.0	3.905336
191152	Grand Lux Cafe	3327 Las Vegas Blvd S, Ste 1580	4.0	1193	Restaurants, American (New), Food, American (T...	2.0	3.894250
186688	Triple George Grill	201 N 3rd St, Ste 120	4.0	1064	American (New), Steakhouses, Seafood, Restaurants	2.0	3.884677
163462	Yard House	3545 Las Vegas Blvd	4.0	1015	American (Traditional), Restaurants, American ...	2.0	3.880571

Figure 12. Simple recommendation engine based on users' input and weighted rating.

The recommendation engine is further improved by extending its functions of searching different cities and different business types. As shown in Figure 13, the "SimpleRecommedation2" engine recommends "Beauty & Spas" and "Barbers" businesses with price range <=1 in the city of Toronto.



```
SimpleRecommendation2('Toronto','Beauty & Spas','Barbers',1,0.7).head(5)
```

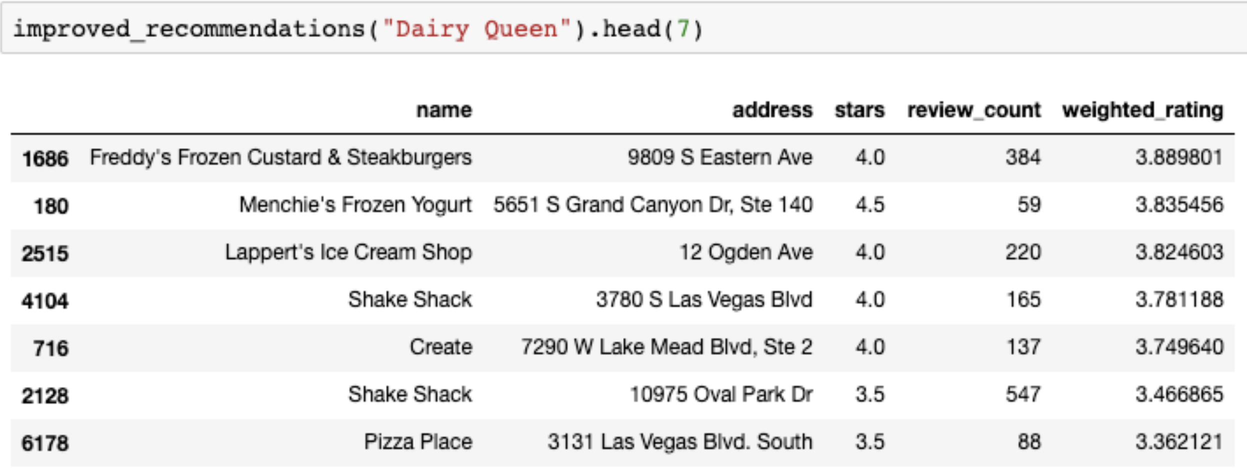
```
Recommend ['Barbers'] restaurants in city Toronto with price <= 1 and review_count >= 8
```

	name	address	city	stars	review_count	categories	PriceRange	weighted_rating
65747	Uptown Barber Shop	4 Isabella Street	Toronto	5.0	41	Beauty & Spas, Hair Salons, Barbers	1.0	4.861742
130655	Tom's Barber Shop	823 Runnymede Road	Toronto	5.0	26	Beauty & Spas, Barbers, Hair Salons	1.0	4.801103
185424	George's Barber Shop	182 Royal York Road	Toronto	5.0	15	Barbers, Beauty & Spas	1.0	4.706799
36697	Mr. Nonno Barber Shop	609 Bloor Street W	Toronto	5.0	12	Barbers, Beauty & Spas	1.0	4.663254
3450	Anna's Barber Shop	7 Erskine Avenue	Toronto	4.5	15	Barbers, Beauty & Spas	1.0	4.383523

Figure 13. Enhanced simple recommendation engine.

Content-filtering recommendation

The Content Based Recommender recommends restaurants based on the restaurant's similarity, calculated by the cosine similarity with TFIDF matrix. Figure 14 shows the recommended restaurants which are similar to the restaurant of "Dairy Queen".



```
improved_recommendations("Dairy Queen").head(7)
```

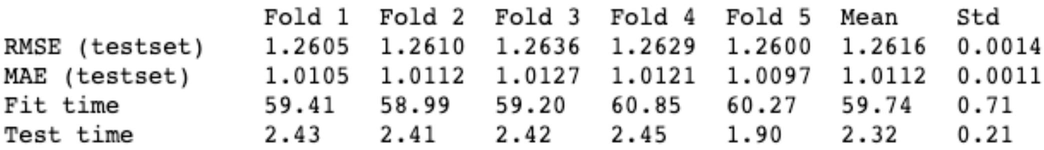
	name	address	stars	review_count	weighted_rating
1686	Freddy's Frozen Custard & Steakburgers	9809 S Eastern Ave	4.0	384	3.889801
180	Menchie's Frozen Yogurt	5651 S Grand Canyon Dr, Ste 140	4.5	59	3.835456
2515	Lappert's Ice Cream Shop	12 Ogden Ave	4.0	220	3.824603
4104	Shake Shack	3780 S Las Vegas Blvd	4.0	165	3.781188
716	Create	7290 W Lake Mead Blvd, Ste 2	4.0	137	3.749640
2128	Shake Shack	10975 Oval Park Dr	3.5	547	3.466865
6178	Pizza Place	3131 Las Vegas Blvd. South	3.5	88	3.362121

Figure 14. Content filtering recommendation engine.

Collaborative filtering recommendation

The content-based engine suffers from several limitations. The engine is not really personal in that it doesn't capture the personal tastes and biases of a user. Anyone querying the engine for recommendations based on a restaurant will receive the same recommendations for that restaurant. Collaborative recommendation is an alternative to achieve personalized recommendation based on users' personal tastes and biases.

The algorithm of SVD is used to predict the ratings of a user on his/her unrated restaurants. The SVD model is trained with 5-fold cross-validation, and the evaluation results are shown in Figure 15. The mean Root Mean Square Error (RMSE) of this SVD model is 1.26.

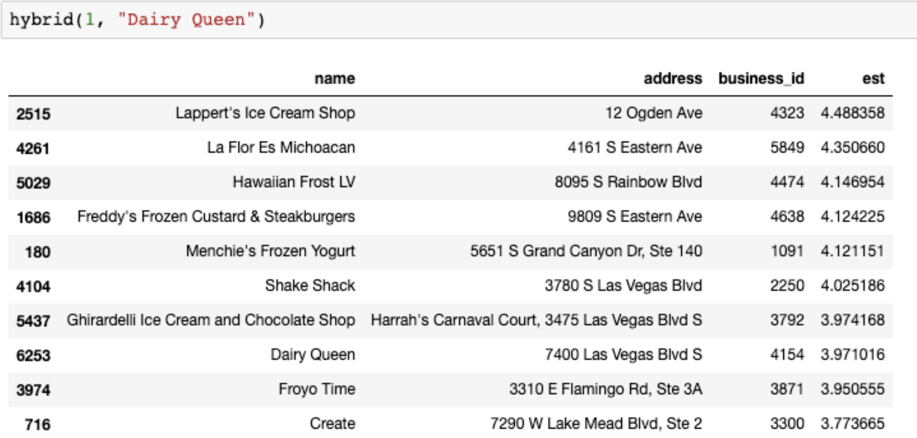


```
                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std     
RMSE (testset)    1.2605  1.2610  1.2636  1.2629  1.2600  1.2616  0.0014  
MAE (testset)     1.0105  1.0112  1.0127  1.0121  1.0097  1.0112  0.0011  
Fit time          59.41   58.99   59.20   60.85   60.27   59.74   0.71    
Test time         2.43    2.41    2.42    2.45    1.90    2.32    0.21    
```

Figure 15. The performance of the SVD model.

Hybrid recommendation

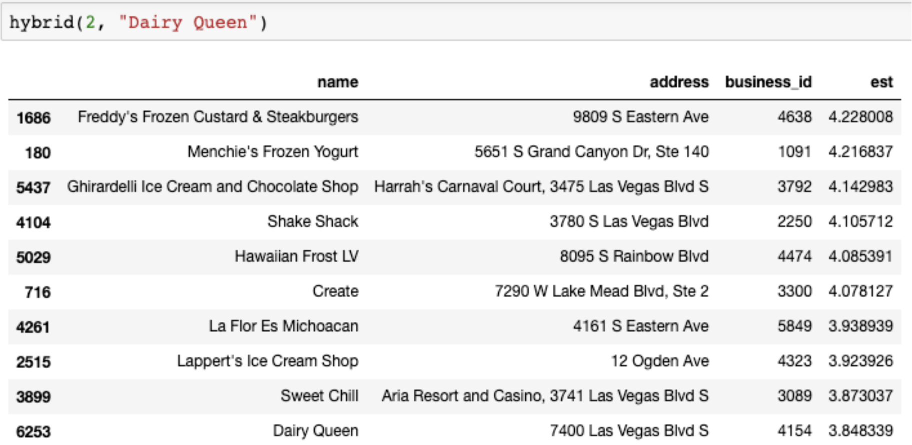
The hybrid recommender combines the content based and collaborative filter-based engines. It takes User ID and the name of a restaurant as input, and outputs similar restaurants sorted on the basis of expected ratings by that particular user. Figure 16 and 17 shows the hybrid recommendation results for different users (user 1 and user 2). For recommendation restaurants similar to “Dairy Queen”, the hybrid engine gives different recommendation lists. For user 1, the first recommended restaurant is “Lappert’s Ice Cream Shop”, and the rating of user 1 on this restaurant is predicted to be 4.49. For user 2, the “Lappert’s Ice Cream Shop” only appears at the 8th recommendation position, and user 2 is predicted to rate it with a star rating of 3.92. Therefore, our hybrid recommendation engine could make recommendations based on users’ personal tastes and biases.



```
hybrid(1, "Dairy Queen")
```

	name	address	business_id	est
2515	Lappert's Ice Cream Shop	12 Ogden Ave	4323	4.488358
4261	La Flor Es Michoacan	4161 S Eastern Ave	5849	4.350660
5029	Hawaiian Frost LV	8095 S Rainbow Blvd	4474	4.146954
1686	Freddy's Frozen Custard & Steakburgers	9809 S Eastern Ave	4638	4.124225
180	Menchie's Frozen Yogurt	5651 S Grand Canyon Dr, Ste 140	1091	4.121151
4104	Shake Shack	3780 S Las Vegas Blvd	2250	4.025186
5437	Ghirardelli Ice Cream and Chocolate Shop	Harrah's Carnaval Court, 3475 Las Vegas Blvd S	3792	3.974168
6253	Dairy Queen	7400 Las Vegas Blvd S	4154	3.971016
3974	Froyo Time	3310 E Flamingo Rd, Ste 3A	3871	3.950555
716	Create	7290 W Lake Mead Blvd, Ste 2	3300	3.773665

Figure 16. Hybrid recommendation engine for user 1.



```
hybrid(2, "Dairy Queen")
```

	name	address	business_id	est
1686	Freddy's Frozen Custard & Steakburgers	9809 S Eastern Ave	4638	4.228008
180	Menchie's Frozen Yogurt	5651 S Grand Canyon Dr, Ste 140	1091	4.216837
5437	Ghirardelli Ice Cream and Chocolate Shop	Harrah's Carnaval Court, 3475 Las Vegas Blvd S	3792	4.142983
4104	Shake Shack	3780 S Las Vegas Blvd	2250	4.105712
5029	Hawaiian Frost LV	8095 S Rainbow Blvd	4474	4.085391
716	Create	7290 W Lake Mead Blvd, Ste 2	3300	4.078127
4261	La Flor Es Michoacan	4161 S Eastern Ave	5849	3.938939
2515	Lappert's Ice Cream Shop	12 Ogden Ave	4323	3.923926
3899	Sweet Chill	Aria Resort and Casino, 3741 Las Vegas Blvd S	3089	3.873037
6253	Dairy Queen	7400 Las Vegas Blvd S	4154	3.848339

Figure 17. Hybrid recommendation engine for user 2.

7. Conclusion and Future work

This project explores the usage of 3 feature extraction methods and 6 classifiers for classifying Yelp review sentiment. We found the feature extraction approaches would greatly affect the model performance. Dropping neutral reviews could increase about 10% accuracy for all the classifiers. Compared to countvectorizer and tfidfvectorizer, hashingvectorizer is the best feature extraction approach for a large data size, since it could reduce the data dimension and the algorithm run time. Our best classifier neural network model with the hashing vectorizer can handle 200000 review texts, which takes only 166 seconds for 5-folder cross validation using our personal laptop. The best neural network classifier has obtained an accuracy of 91.5%. In terms of performance, random forest, gradient boosting, and SVM classifiers tend to have slightly worse results.

We also built 4 different recommendation engines based on different ideas and algorithms: simple recommender, content filtering recommender, collaborative filtering, and hybrid filtering engines. Our hybrid recommendation engine could make recommendations for particular users based on their own personal tastes and biases. The RMSE of the single value decomposition algorithm was about 1.26. Our recommendation engines could improve customer's experience as well as the business advertising.

Future work can focus on improving the methods by using a multitude of crowd-sourced data sets other than Yelp. Moreover, different hybrid recommendation system strategies can be built, and performance evaluation methods can be used to compare different hybrid recommendation strategies. Future studies can also examine applying our methods to a variety of different fields of research such as movies, and other fields.

Reference

1. Yelp dataset: https://www.kaggle.com/yelp-dataset/yelp-dataset
2. IMDb rating score formula:
http://answers.google.com/answers/threadview/id/507508.html
3. Yelp information: https://en.wikipedia.org/wiki/Yelp.

4. Umar Farooque, Bilal Khan, Abidullah Bin Junaid, Akash Gupta (2014). Collaborative Filtering based simple restaurant recommender In Proceedings of the **International Conference on Computing for Sustainable Global Development**.

5. Ronen Feldman**, Techniques and applications for sentiment analysis,** Communications of the ACM, 56, 82-89, 2013

6. Andreea Salinca, **Business Reviews Classification Using Sentiment Analysis,** 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 247-250, 2015.

8. Gauch, Susan. "**Personalized News Recommendation Using Twitter.**" *ResearchGate*, Nov. 2013, www.researchgate.net/publication/261199217_Personalized_News_Recommendation_Using_Twitter.

9. Basilico J, Hofmann T. (2005). Star Quality: Sentiment Categorization of Restaurant Reviews. In the Proceedings of the ACL. 115-124.

10. How to prepare text data for machine learning with scikit-learn. https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/

11. Zhang, Liang, et al. "Diversity Balancing for Two-Stage Collaborative Filtering in Recommender Systems." *MDPI*, Multidisciplinary Digital Publishing Institute, 13 Feb. 2020, www.mdpi.com/2076-3417/10/4/1257.