12/7/17

CASE STUDY 4: YELP RECOMMENDER SYSTEM

TEAM 12

DS 501: INTRODUCTION TO DATA SCIENCE

TEAM MEMBERS

Chu Wang

Saranya Manoharan

Rishitha Kiran

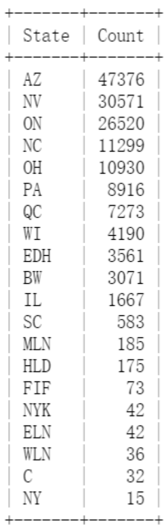
Di You

1. INTRODUCTION

Writing online reviews about restaurants, spas, shopping etc. has become popular over the years. These reviews benefit in promoting businesses and also help users to evaluate the business and make a choice. Yelp is a social networking website that publishes crowd-sourced reviews about businesses. Popularity of restaurants has grown over time and has become more important than ever. People prefer to try restaurants with good reviews and ratings, while planning a trip. In this project we use the datasets that we collected from Yelp's Dataset Challenge and work with the necessary dataset. Yelp provided us with five different datasets which includes business, check-in, review, tip, photos and user. In this project we employed two approaches: Top restaurant recommendation in an area, recommend potential friends to users based on similarity metrics and second level relationship respectively and Recommending potential users to business owners. The structure of the remaining paper is as follows: the first module of our project discusses about data cleaning and filtering. The second module talks about basic Data analysis process which is followed by Sentimental analysis for predicting whether a review is good or bad. Then, the fourth module discusses about the various recommendation techniques.

2. DATA CLEANING AND FILTERING

Since the original dataset from YELP is too large to process and there are too many noisy within the dataset. We went through the distribution of original dataset and decide to select a particular state as our final dataset. Eventually, the whole project is based on a sample of North Carolina State, which contains 11,299 business_ids, around 70,000 users and 270,000 reviews. For computation convenient for recommend potential friends to users based on similarity metrics and recommending potential users to business owners, we transform original string user_id into numbers.



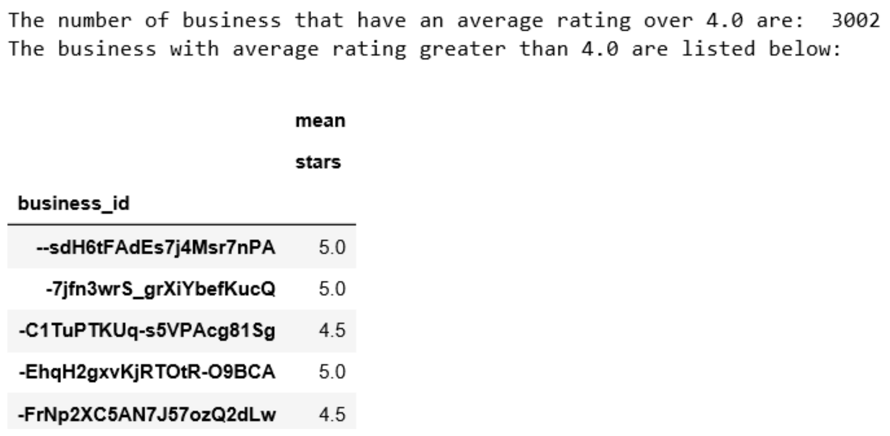
```
+-------+-------+
| State | Count |
+-------+-------+
| AZ    | 47376 |
| NV    | 30571 |
| ON    | 26520 |
| NC    | 11299 |
| OH    | 10930 |
| PA    |  8916 |
| QC    |  7273 |
| WI    |  4190 |
| EDH   |  3561 |
| BW    |  3071 |
| IL    |  1667 |
| SC    |   583 |
| MLN   |   185 |
| HLD   |   175 |
| FIF   |    73 |
| NYK   |    42 |
| ELN   |    42 |
| WLN   |    36 |
| C     |    32 |
| NY    |    15 |
+-------+-------+
```

Figure 1: Business dataset with states and count

3. DATA ANALYSIS

3.1. Filter restaurants whose star rating is more than 4.0

From the North Carolina dataset which is obtained after cleaning and filtering, we did few basic data analysis of finding restaurants whose star rating is more than 4.0. We displayed all those restaurants which have average rating more than 4.0 stars using pivot table. Figure 2 shoes the top businesses whose star rating is more than 4.0.



```
The number of business that have an average rating over 4.0 are:  3002
The business with average rating greater than 4.0 are listed below:
```

	mean
	stars
business_id	
--sdH6tFAdEs7j4Msr7nPA	5.0
-7jfn3wrS_grXiYbefKucQ	5.0
-C1TuPTKUq-s5VPAcg81Sg	4.5
-EhqH2gxvKjRTOtR-O9BCA	5.0
-FrNp2XC5AN7J57ozQ2dLw	4.5

Figure 2: Businesses whose star rating is >4.0

3.2. Sort the restaurants based on votes

Next, we sorted the businesses based on various votes such as cool, funny, useful and also stars. Figure 3 shows the bar plot of the sorted businesses.

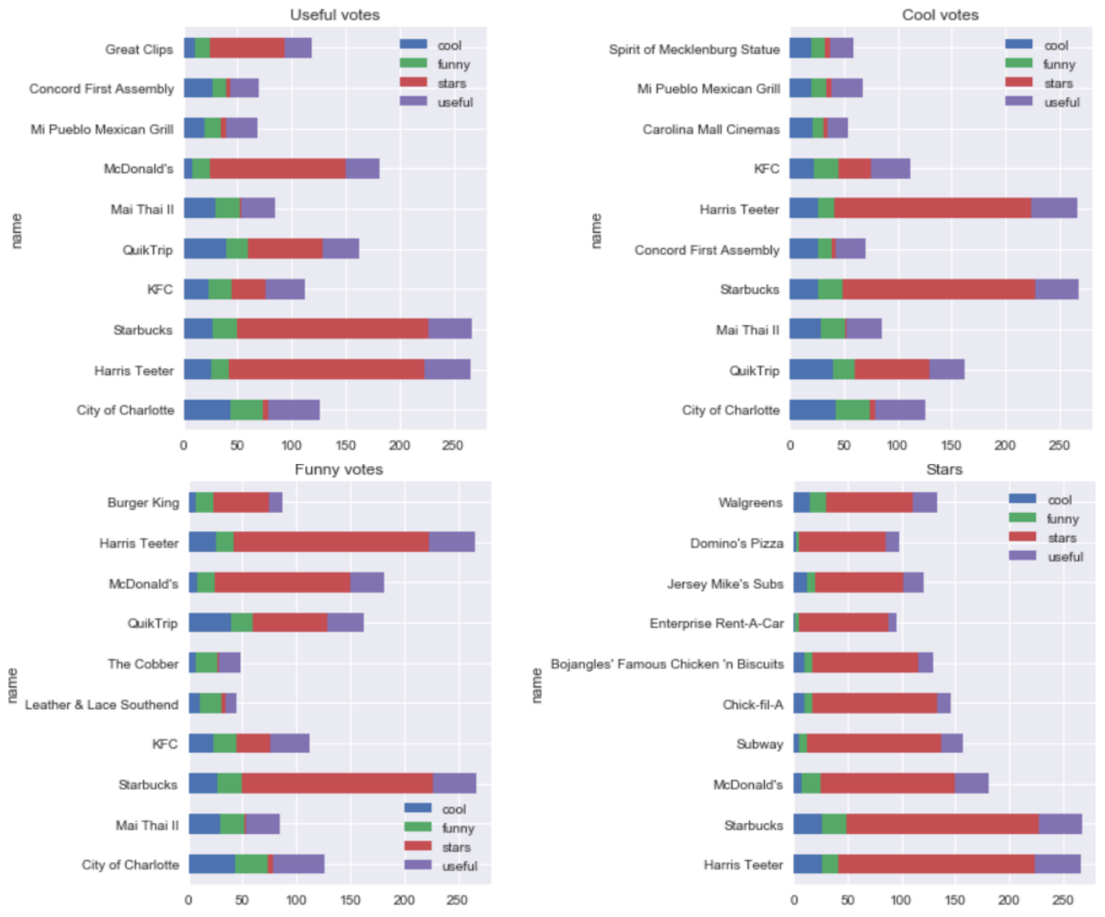


Figure 3: Sorted businesses based on votes

3.3. Find the most frequent and unique word in the reviews

Next, we created a word cloud which contains the most frequently used words in the reviews. Figure 4 shows the Word Cloud.

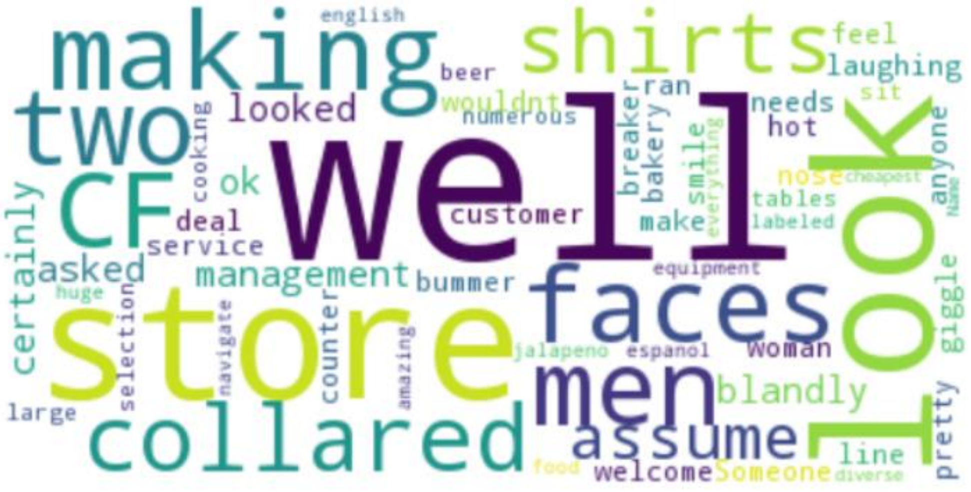


Figure 4: Word Cloud of frequent words

4. SENTIMENTAL ANALYSIS

We build a simple text classifier using Python's Pandas, NLTK and Scikit-learn libraries. Our goal is to build a sentiment analysis model that predicts whether a user liked a local business or not, based on their review on Yelp. The subset of review dataset containing only those reviews from the above selected restaurants is used for this purpose. We calculate the length of each text review in order to analyze the relationship between text length and the ratings. We then make use of Seaborn's FacetGrid which allows us to create a grid of histograms placed side by side. We can use FacetGrid to see if there's any relationship between our newly created text length feature and the stars rating. From the results (Figure 5) we see that the distribution of text length across all five ratings are as shown below.

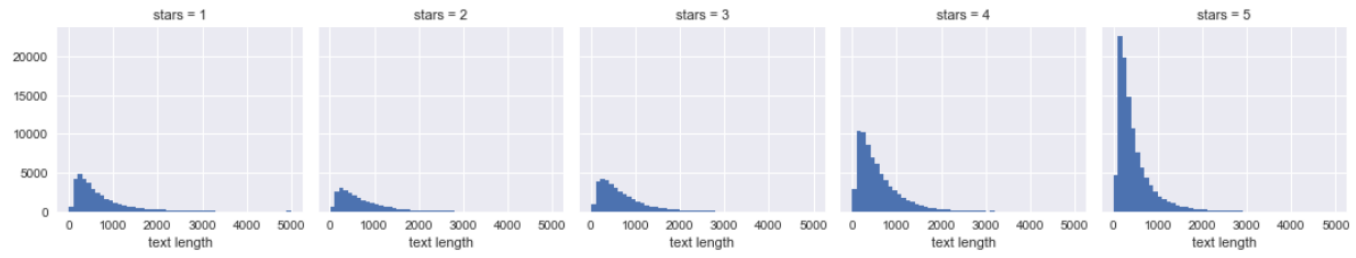


Figure 5: Distribution of text length across all five ratings

The number of text reviews seems to be skewed a lot higher towards the 4-star and 5-star ratings. Next, we create a box plot of the text length for each star rating (Figure 6)

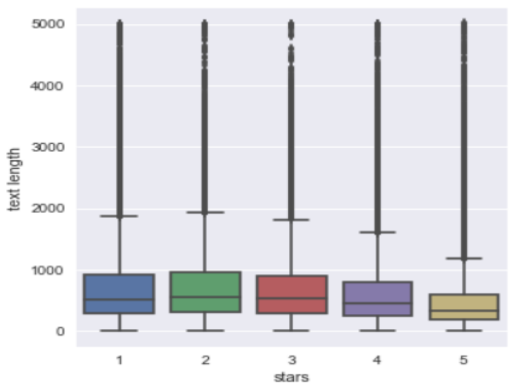


Figure 6: Box plot of the text length for each star rating

From the plot, looks like there are quite a few outliers (which can be seen as points above the boxes). Because of this, maybe text length won't be such a useful feature to consider after all. Next, we group the data by the star rating, and see if we can find a correlation between features such as cool, useful, and funny. We use the correlation method from Pandas to find any correlations in the data frame. Our task is to predict if a review is either bad or good, so we just grab reviews that are either 1 or 5 stars from the review data frame. We can store the resulting reviews in a new data frame called yelp_class. We make use a classification algorithm here, the classification algorithm will need some sort of feature vector in order to perform the classification task. The simplest way to convert a corpus to a vector format is the bag-of-words approach, where each unique word in a text will be represented by one number.First, let's write a function which splits the message into its individual words, and return a list. We will also remove the very common words (such as "the", "a", "an", etc.), also known as stopwords. To do so, we take advantage of the NLTK library stopwords. We remove punctuation, stopwords, and returns a list of the remaining words, or tokens.

4.1. Vectorization

Our reviews are present as lists of tokens. To make the Scikit-learn algorithms to work on our text, we need to convert each review into a vector. We use Scikit-learn's Count-Vectorizer to convert the text collection into a matrix of token counts. This can be imagined as a resulting 2-D matrix, where each row is a unique word, and each column is a review. Since the number of reviews are more, we can expect a lot of zero counts for the presence of a word in the collection. Because of this, Scikit-learn will output a sparse matrix.

4.2. Training data and test data

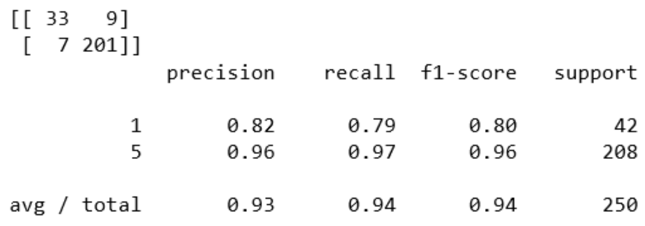
Once we have finished processing the review text, we split our X and y into a training and a test set using train_test_split from Scikit-learn. We will use 30% of the dataset for testing and 70% for training purpose.

4.3. Training our model

Multinomial Naive Bayes is a specialized version of Naive Bayes designed more for text documents. We build a Multinomial Naive Bayes model and fit it to our training set (X_train and y_train).

4.4. Testing and Evaluating models.

Once our model has been trained, it is time to see how well it predicts the ratings of previously unseen reviews (reviews from the test set). First, let's store the predictions as a separate dataframe called preds. We evaluate our predictions against the actual ratings (stored in y_test) using confusion_matrix and classification_report from Scikit-learn.



```
[[ 33   9]
 [  7 201]]
             precision    recall  f1-score   support

          1       0.82      0.79      0.80        42
          5       0.96      0.97      0.96       208

avg / total       0.93      0.94      0.94       250
```

Figure 7: Classification report for Multinomial NB

4.5. Predict whether a review is good or bad

From our model the reviews with star rating less than 3.0 are negative review and those greater than 3.5 are positive review. The below shows the prediction results:

Positive review:

"I am so glad that they got one of these stores here in Charlotte North Carolina I have been trying to find it I found you found Golden Krust Bakery you must eat their the best you're making pulled ever wish they had more big location that is just a little too far for me to get but I get there when I can try it you will love it Golden Krust the best ever"

Predicted result:

5

Negative review:

"This place is NOT Caribbean food! It's terrible. You must ask for a drink separate from the combo even though you've ordered a combo. They're not going to offer it to you either. The food tastes horrible. I wish I could sit outside and turn people around to save their souls. But unfortunately, this is all I can do. Take heed and go elsewhere!!!!!"

Predicted result:

1

4.6. Reflection of Incorrect Classification Instance

Since our model is more biased towards positive reviews compared to negative review. Hence, our model predicts wrong.

Negative review:

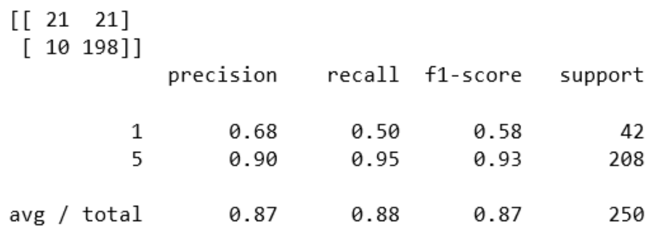
"What a dump! I rented this place because it had a convenient location and the price wasn't too bad. Although it says it's non-smoking, my room reeked of cigarette smoke! So bad that I thought I would have an asthma attack. I don't mind that everything is outdated, it was a great price, but I seriously feel like I need to get a cootie shot. The front desk staff was disheveled and very sloppy looking, and they were a direct representation of the hotel itself. Don't be fooled by the outside because the outside isn't too bad. Once you get inside, be prepared for filth on every kind of level. I will never go back! Management needs to wake up."

Predicted result:

5

4.7. Comparison of Classification Algorithms

We used different classification algorithms to study their comparative results of their confusion matrix. We used Gaussian NB, SVC and K-Neighbors classifiers additional to Multinomial NB. The results are given below in Figure 8.

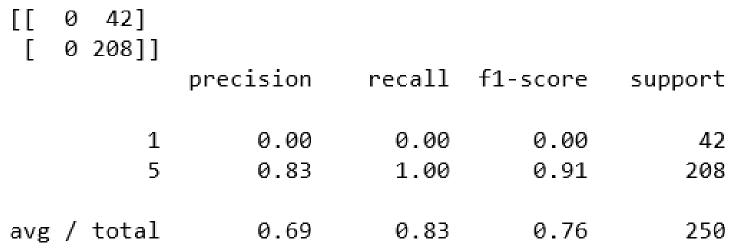


```
[[ 21  21]
 [ 10 198]]
             precision    recall  f1-score   support

          1       0.68      0.50      0.58        42
          5       0.90      0.95      0.93       208

avg / total       0.87      0.88      0.87       250
```

Gaussian NB

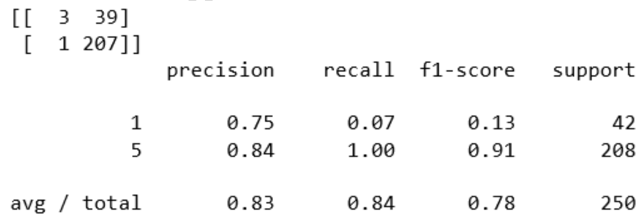


```
[[  0  42]
 [  0 208]]
             precision    recall  f1-score   support

          1       0.00      0.00      0.00        42
          5       0.83      1.00      0.91       208

avg / total       0.69      0.83      0.76       250
```

C-Support Vector Classification



```
[[  3  39]
 [  1 207]]
             precision    recall  f1-score   support

          1       0.75      0.07      0.13        42
          5       0.84      1.00      0.91       208

avg / total       0.83      0.84      0.78       250
```

KNeighbors Classifier

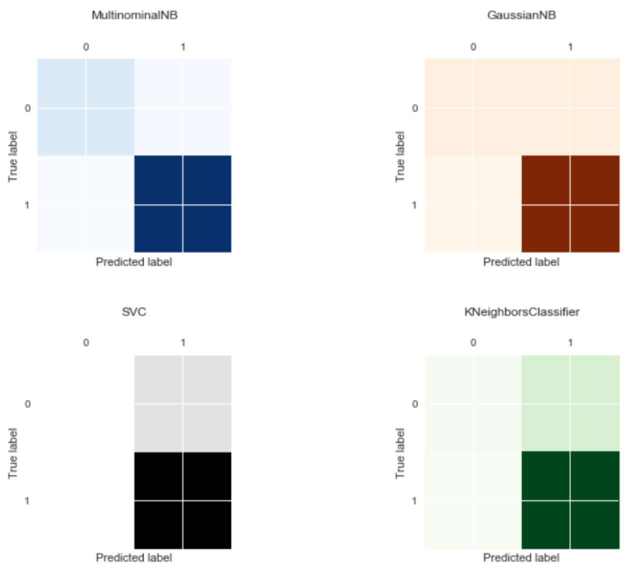


Figure 8: Comparison of Classification algorithms

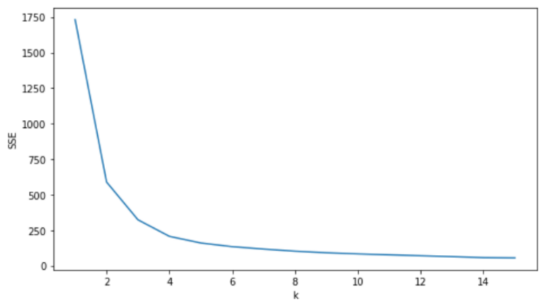
5. TOP RESTAURANT RECOMMENDATION IN AN AREA

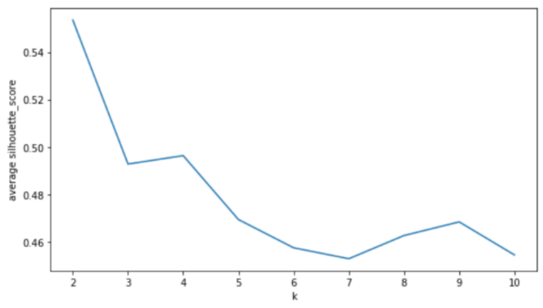
5.1. Business Intuition

Yelp contains review data of various restaurants in a city and helps the user in choosing a restaurant. But, it doesn't recommend any restaurants to a specific user. However, people are always interested in the most popular restaurant in town, and it is difficult to define the most popular restaurants.

5.2. Math Approach & Coding Implement

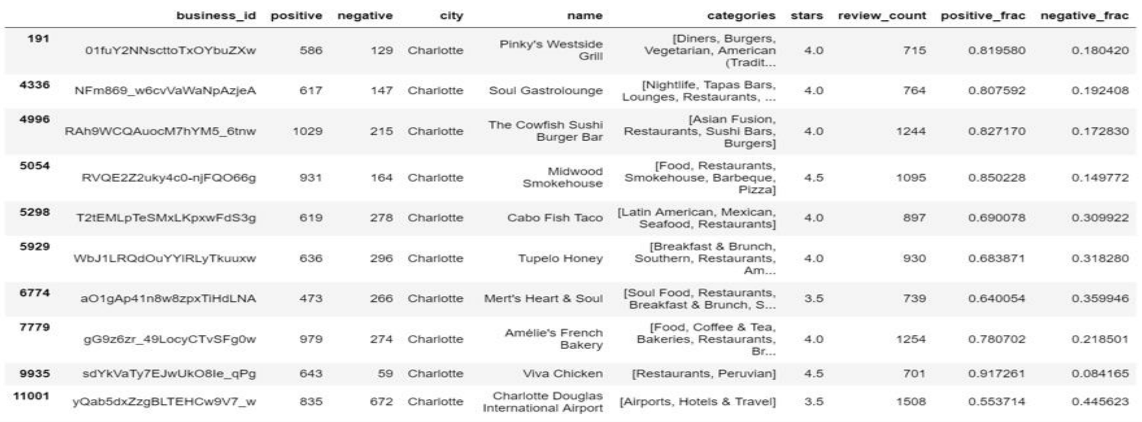
We define this problem as a clustering mathematic problem. The input of the clustering is review count, average rating, percentage of positive ratings and percentage of negative ratings. We define 4,5 as positive ratings and 1,2 as negative ratings. To determine the number of clusters in k-means, we employed 2 methods. One idea is to run k-means clustering on dataset for a range of values of k (say, k from 1 to 15 in this case), and calculate the SSE for each value of k. Then we plot a line chart of the result and obtain several values of k where SSE decreased significantly, which we call it 'elbow' point. We also calculate Silhouette coefficient for data without label and NMI for data with label respectively. And combining the results together to determine the best k value. The Silhouette coefficient will be high at first, so we try to find the local largest value instead of global largest value. According to analysis above, we cluster those business entities into 4 clusters, and we got a top-store list consists of 9 restaurants and 1 Airport. The second cluster contains 208 business entities and should be secondary choices of our customers.





5.3. Conjectures

All of the business entities on our top list locate in Charlotte, covering different types of cuisines and our customer is able to enjoy breakfast, brunch to nightlife from our list. (Maybe we can plot more with business entities in the first 2 clusters)



	business_id	positive	negative	city	name	categories	stars	review_count	positive_frac	negative_frac
191	01fuY2NNscttoTxOYbuZXw	586	129	Charlotte	Pinky's Westside Grill	[Diners, Burgers, Vegetarian, American (Tradit...	4.0	715	0.819580	0.180420
4336	NFm869_w6cvVaWaNpAzjeA	617	147	Charlotte	Soul Gastrolounge	[Nightlife, Tapas Bars, Lounges, Restaurants, ...	4.0	764	0.807592	0.192408
4996	RAh9WCQAuocM7hYM5_6tnw	1029	215	Charlotte	The Cowfish Sushi Burger Bar	[Asian Fusion, Restaurants, Sushi Bars, Burgers]	4.0	1244	0.827170	0.172830
5054	RVQE2Z2uky4c0-njFQO66g	931	164	Charlotte	Midwood Smokehouse	[Food, Restaurants, Smokehouse, Barbeque, Pizza]	4.5	1095	0.850228	0.149772
5298	T2tEMLpTeSMxLKpxwFdS3g	619	278	Charlotte	Cabo Fish Taco	[Latin American, Mexican, Seafood, Restaurants]	4.0	897	0.690078	0.309922
5929	WbJ1LRQdOuYYIRLyTkuuxw	636	296	Charlotte	Tupelo Honey	[Breakfast & Brunch, Southern, Restaurants, Am...	4.0	930	0.683871	0.318280
6774	aO1gAp41n8w8zpxTiHdLNA	473	266	Charlotte	Mert's Heart & Soul	[Soul Food, Restaurants, Breakfast & Brunch, S...	3.5	739	0.640054	0.359946
7779	gG9z6zr_49LocyCTvSFg0w	979	274	Charlotte	Amélie's French Bakery	[Food, Coffee & Tea, Bakeries, Restaurants, Br...	4.0	1254	0.780702	0.218501
9935	sdYkVaTy7EJwUkO8le_qPg	643	59	Charlotte	Viva Chicken	[Restaurants, Peruvian]	4.5	701	0.917261	0.084165
11001	yQab5dxZzgBLTEHCw9V7_w	835	672	Charlotte	Charlotte Douglas International Airport	[Airports, Hotels & Travel]	3.5	1508	0.553714	0.445623

6. RECOMMEND POTENTIAL FRIENDS TO USERS

6.1. Business Intuition

When it comes to Yelp, the first term occurs to most people is probably 'restaurant', it is known as a platform where people publishing crowd-sourced reviews about businesses. But it does focusing on social network as well. If you are familiar with Yelp, there are tons of ways to interact with your friends and this would definitely helpful to improve user loyalty towards YELP. So, in this question, we employed 2 approaches to recommend potential friends to users based on similarity metrics and second level relationship respectively.

Also, YELP is providing offline activities to their users to participate such as check digitally on its mobile app and we would like to expand offline component further. Considering that tons of Yelpers are connecting online, people who hold opinions of local business and food could get together and have fun in real life.

6.2. Math Approach & Coding Implement

1. Based on similarity metrics

We use several attributes in user profile as the basis of similarity calculation, which are 'useful', 'funny', 'cool', 'friends', 'fans', 'review_count', 'compliment_photos','compliment_list','compliment_funny', 'compliment_note', ‘complement_plain’, ‘compliment_writer’, ‘compliment_cute’, ‘compliment_more’, ‘compliment_hot’, ‘compliment_profile’ and ‘compliment_cool’ and their proportion in all the compliment user received. We normalized those feature with L2 norm, then did K-means clustering(k=100) on these data instances to project them into a relatively low dimension and it is easier to compute similarity within the clusters.

The similarity calculated using cosine similarity metric and at the end of this process, we output 10 most similar friends to each user.

2. Based on second degree relationship

Indirect connections between people may also influence cohesion within a community and it is always an easy way to build up your network by knowing the friends of your friends. In this section, we try to explore 2nd friend network and make recommendations. Basically, the idea of this solution is based on graph model. Considering there are N people within a network, some of them are friends while some are not. The friendship is transitive in nature. If A is a direct friend of B and B is a direct friend of C, then A should be indirect friend of C. Also, since the friendship relationship should be a directed graph, we define the 1st degree friend as follow:

(1) if B is in A’s friend list, then B is A’s friend;

(2) if B is in A’s friend list, no matter whether A in or not in B’s friend list, then A is B’s friend; For the result of, we output all the 2nd friend for each user.

6.3. Conjectures

There is barely no overlapping between the lists we got by those 2 different approaches, but each of them is some kind inspiring.

7. RECOMMEND POTENTIAL USERS TO BUSINESS OWNERS USING COLLABORATIVE FILTERING MODEL BASED ON LFM

7.1. Business Intuition

The enormous growth in online availability of information content has made Recommender System an integral part of many online service portals. This technique enables those business entities to offer some form of recommendations to users with more clarity and accuracy by limiting the information needed to search through. However, there are hardly applications focusing on recommending users to business owners. We proposed this idea to sell a premium service to business owners that provide them with their potential customers. This would give business owners a chance to be more proactive in their promotion activities such as pushing promotion news more efficiently since they can distinguish their potential customers.

7.2. Math Approach & Coding Implement

A popular approach for Recommender System design is Collaborative Filtering, which uses the rating history of users to a subset of items in repository to make future recommendations. The biggest challenge in building an effective recommender system is probably to infer users’ preference from extremely limited information because it is hard to obtain rating information in real word. Thus, strategies are built on the principals of latent factor models (LFM). LFM are constructed on the assumption that a user's choice for an item is deeply influenced by a handful set of factors that are very specific to the domain, which we call them the latent factors. As the name suggested, these factors are in general not obvious and we might be able to think some of them but it's hard to estimate their impact on users' behavior. The goal is to infer those so called latent factors from the rating data by using mathematical techniques.

7.3. Conjectures

The LFM model also avoided cold-start problem [2] in some degree, but it still performs worse when the user-item matrix is too sparse. Also, the computation complexity is extremely high for LFM model, so we had to do sample the dataset and keeps only 1000 business entities and 6000 users.

REFERENCES

[1] http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[2] S. Rendle. Factorization machines with libfm. ACM Transactions on Intelligent Systems and Technology, 3(3):57:1–57:22, May 2012.

[3] https://medium.com/tensorist/classifying-yelp-reviews-using-nltk-and-scikit-learn-c58e71e962d9