**Eating Mate Recommendation for Yelp Users**

Hao Xu, Zehao Song, Yutong Tie

Department of Computer Science

Columbia University

{hx2208, zs2324, yt2549}@columbia.edu

*Abstract*—**Yelp has been more and more prevalent in making recommendations for people’s daily life, like where to eat or where to shop, however, little has been done regards making recommendations for who they could go with. We proposed that eating mate could be recommended based on the review history of each user. Therefore, we collected review data from Yelp to make eating mate recommendations, which included a collaborative filtering of the review data, linear regression on a single user and business data, and finding K nearest neighbors of each user. The experiment on data points confirms that it is a good way to make recommendations for eating mate. Since the current experiment is based on review data on partial cities in the United States, more tests are needed for other cities, and in order to have a better representation for recommendation, a User Interface is needed in the future.**

Keywords-Yelp; eating mate; recommendation; collaborative filtering

# Introduction

Yelp is a platform about sharing reviews and making recommendations. On the one hand, users of Yelp could submit reviews on different shops and restaurants using a one to five-star rating. On the other hand, Yelp would make recommendations for users based on their past review history. McNichol states Yelp has become “one of the most import sites on the Internet” [1]. According to the statistics provided by Yelp on 2016, it had about 102 million total reviews and a monthly average of 92 million unique visitors in the second quarter of 2016 [2].

Though Yelp has strong functionalities in helping people find a place to eat, it has less tricks in finding people an eating mate. It is a great business loss since under most circumstance, people would not like to go eating alone, no matter how good the business is.

Based on this situation, this project is intended to make recommendations for people who share similar preferences to have meal together based on each individual’s review history. We are dedicated to dig into each user’s tastes which hides behind their reviews, and hopefully, to analysis taste distributions of each city.

Finally, the application should work like: 1). For a new user, we would like this user answer some question so we can know about his preference and make eating mate recommendation based on his answers. 2). For an existed user, we would make recommendation directly based on his rating history. The final out would be several eating mates and corresponding restaurant.

# Related Works

I do not think there is any previous work that makes recommendation for eating mates, however, “people you may know” is a section that has been widely implemented in social websites. Besides, most E-commerce websites, like Amazon and eBay, make recommendations for goods that people would like to shop. These kinds of recommendation systems are in different tunes rendered with equal skill as eating mate recommendation. I would list a few websites and how they make recommendations in the following paragraphs.

Facebook [3] is dedicated to connect everyone who is using their website. According to Vignesh’s answer on Quora [4], Facebook makes friend recommendations on following 5 aspects: 1). Right after registration, every user would be asked to fill in a questionnaire about his background, like education, region, interest, etc. Based on that information, Facebook would recommend people who went to same school or work at the same place to that user. 2). Once has some friends, Facebook would recommend one’s friends to him. Say if A and B are friends, B and C are friends, then Facebook would recommend A to C and C to A at the same time. 3). Facebook would recommend two people in the same group to be friend. For example, if user A is in Columbia Engineering group and user B is also in that group, then Facebook would recommend A to B and B to A at the same time. 4). Facebook would also recommend two people like same public page to be friend. For example, if user A likes Bayern Munich and user B likes Bayern Munich as well, then Facebook would recommend them to be friend, however, if A likes Bayern Munich and B likes Borussia Dortmund, it might not recommend them to be friend due to the counter relationship between these two teams.

LinkedIn [5] is designed for people to build up their career circles and trying to help people in job-hunting. Linkedin makes recommendations for their user based on following 3 aspects: 1). Like Facebook, LinkedIn makes friend recommendation based on users’ basic information, like education, region, etc. It would try to connect users from same school, same area or same company. 2). Unlike Facebook, LinkedIn is intended to build up a more real network among people than Facebook, which guarantees the quality of connections in LinkedIn. LinkedIn would import contacts via each users’ email address and recommend friends’ connections to each user.

Amazon’s [6] recommendation starts from finding a set of customers who have items overlap with the user’s purchase and rate history. Then it uses different algorithms aggregates items from those customers, remove the items that has already been purchased or rated, and recommends items that are left to the user. There are basically 2 algorithms being applied in Amazon’s recommendation. 1). Traditional collaborative filtering. Each customer is represented as an N-dimensional vector. For example, there are two users A and B, the similarity of A and B is calculated by cos(A, B). 2). There are many attributes that described a specific user and the recommendation is based on different cluster that the user belongs to. Therefore, the recommendation problem is simplified to a classification problem. Amazon uses those attributes to divide customers into different clusters.

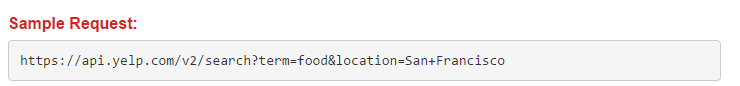
In our project, we choose to do similar as what Amazon does. We represent each user in Yelp as feature vectors and make recommendations based on the nearest neighbor of that user.

# System Overview

In this section, we would ramp up on how we scarp data from Yelp, how our recommendation system works and how the dataset looks like.

Data Collection.

In order to run scarpy, we would need to generate the URL that contains the data we want. In this case, we need to get business\_id for each city. Yelp provides its Search API for developers to search business\_ids that they are interested.



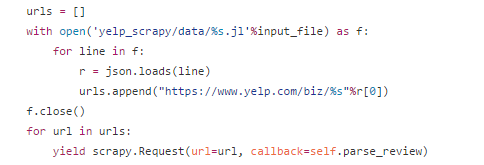
This is how the sample request looks like. In this request, we are fetching business ids in San Francisco that sell food. We fetch data by changing corresponding attributes in the URL. The response from Yelp server is a JSON object. Below is a sample response:



This JSON object contains the import information we want, like business id, name, geo location, etc.

After getting enough number of businesses, we start to gather the users’ reviews of each business by a python crawler implemented via Scrapy Framework.

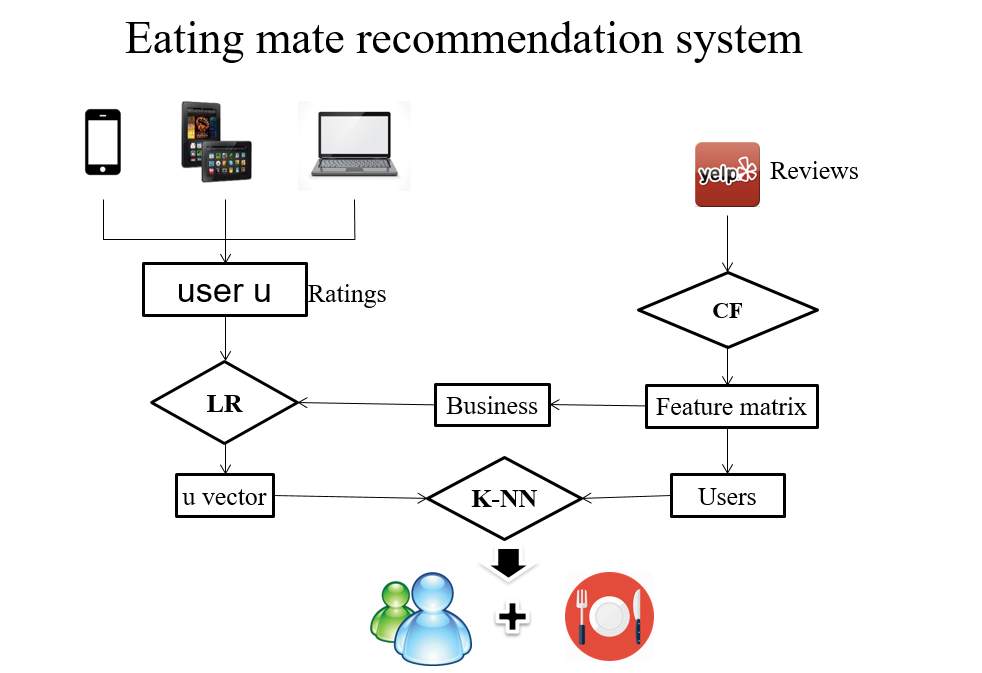
Below is how we generate the target URL and fire the request:



The code above reads lines in generated business id lists in previous step and append it to regular Yelp URL and fire the URL. Then the code parses the HTML code and find the reviews we want in each single page to collect review data. For more detailed code, take a look at our Github: http://www.gethub.com/tsszh/yelp\_recommendation/blob/master/scrapy/yelp\_scrapy/spiders/review\_spider.py

Workflow.

After collecting large enough data, we start to analysis the data and make recommendation for each user. The whole process is as follow:



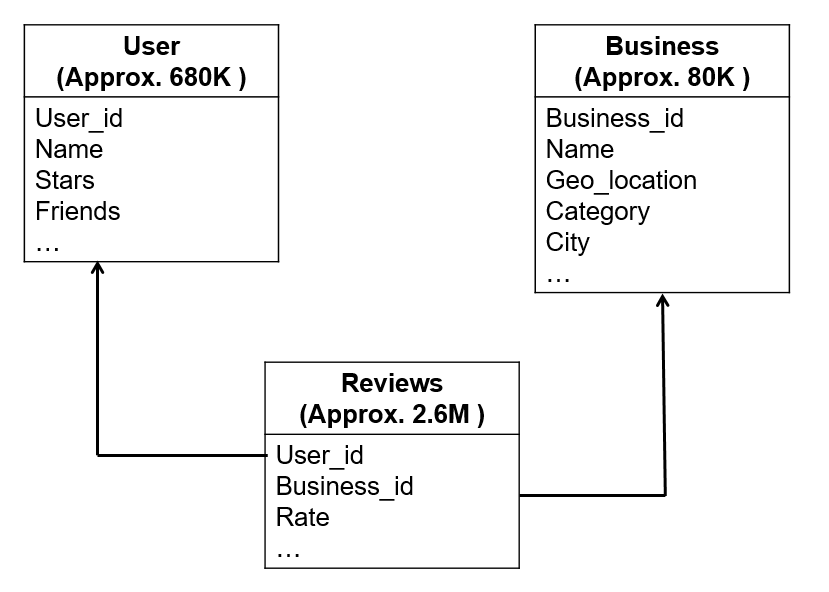
The analysis starts from the review data we collected from Yelp.

Firstly, we do a collaborative filtering over the review data (user\_id, business\_id and rate) to generate the feature matrixs. Save both business feature & user feature matrix for future usage. In principle, to make recommendation for eating mates, we just need to select users with the minimum distance in terms of user features. However, for new users, which is not included in the training sets, the user feature vector is not available.

Then, for each user we want to make recommendation for, we fetch his review history. By doing a linear regression on user’s rating history with Business data, we get a feature vector for that user. Then we make recommendation for eating mate by finding nearest neighbors of user’s feature vector in User data.

Dataset Overview

There are mainly 3 tables in our database system. They are User data, Business data, and Reivew data.



There are about 680K user tuples, 80K business tuples and 2.6M review tuple in total has been collected so far in order to make recommendations.

In User table, each user is assigned a unique key, which is user\_id (primary key). Other information about each user, like name, his friend, his location, etc. has also been included in the table defined above.

In Business table, each business is assigned a unique key, which is business\_id (primary key). Other information about each business, such as name, geolocation, category, etc. is also included in the table defined above.

In review table, each review record is defined by a combination of user\_id and business\_id (primary key). For instance, tuple (1, 2, 3, …) means that user whose id is 1 gives business whose id is 2 a rate 3. Other information like rate, comment, timestamp, etc. is also included in the table.

# Algorithm

Show algorithm and tools you have used to solve your

problem.

1. Collaborative filtering
2. Linear Regression
3. K-Nearest Neighbor

# Software Package Description

Describe the software package that is going to be open

sourced. Show some screen shots of how user use it and or

UI.

# Experiment Results

Describe the experiment results of your algorithm. Show

how you evaluate the performance of your algorithm.

# Conclusion

Conclusion.

In this paper, we covered the entire workflow of our project, dive deep into the algorithms we chose to use, the dataset we collected in our own hand, the process of hyper parameter selection and evaluation of algorithms.

To sum up, we have built an eating mate recommendation system successfully. The system only needs some piece of information about user’s reviews, then it would generate a feature vector of that user, find the nearest neighbor and finally output the eating mate that is recommended. We also deployed a site that supports eating mate recommendations. The result of recommendation is a combination of eating mate and a list of restaurant that could satisfy both users. According to several test results, the recommendation is reasonable and future customer feedback is to be collected.

Contributions.

Future works.

##### Acknowledgment

We would like to thank Prof. Lin for this fantastic semester, the informative lectures and relax atmosphere.

We would like to thank each Teaching Assistant for their helps during office hours.

We would like to every team member for the productive ideas and supportive for each other.

##### Appendix

##### References

1. T. McNichol, “Word on the streed,” NYSE magazine, 2012. Retrieved from http://www.nysemagazine.com/statics/NYSE\_fall\_20

12.pdf.

1. Yelp. (2016). *About us.* Retrieved from https://www.yelp.com/about.
2. Facebook. Avaliable: https://www.facebook.com.
3. N. Vignesh, “How does Facebook’s friend recommendation system work”. Retrieved from https://www.quora.com/How-does-Facebooks-friend-recommendation-system-work
4. Liknedin. Avaliable: https://www.linkedin.com
5. Amazon. Avaliable: https://www.amazon.com