Tinder Eats

Bring people and food together!

Akshay Adiga
Computer Science
University of Southern California
Los Angeles, USA
aadiga@usc.edu

Anish Narang
Computer Science
University of Southern California
Los Angeles, USA
anishnar@usc.edu

Arjun Mohan
Computer Science
University of Southern California
Los Angeles, USA
mohanarj@usc.edu

ABSTRACT

Seldom one prefers to eat alone. Many recent studies have shown how people with similar food preference tend to stay together. Looking from both a restaurant and customer's perspective, eating together has always been economical and saves lot of food. Keeping these points in mind, we came up with a recommender system which can recommend people with similar food preferences and further, suggest restaurants nearby, which the two users would enjoy visiting together. We use user-feedback in the form of reviews to extract food preferences of users. With the advancement in Natural Language Processing and Data Mining, we are able to extract useful information from user reviews and recommend restaurants and similar users using Machine Learning based Collaborative Filtering. In this paper, we discuss the intuition behind the approach used, in the '*Introduction'* section and give a glimpse of the dataset used and discuss various pre-processing and algorithmic methods incorporated, in *'Method'* section. We finally conclude with information about accuracy and other statistics, in the '*Results and Discussion'* section.

INTRODUCTION

Abundant supply of reviews provide useful information and feedback by users for a business. When these reviews are grouped by users, they give a good idea of a user's taste and preferences. With large number of reviews available on restaurants by users, we decided to build a user profile to understand each user's food preference. On researching about review quality, we found almost 80% of the reviews spoke about food items which are inferred from the graph shown in Fig 1. We use this data of how people speak about different food, to infer similar users.

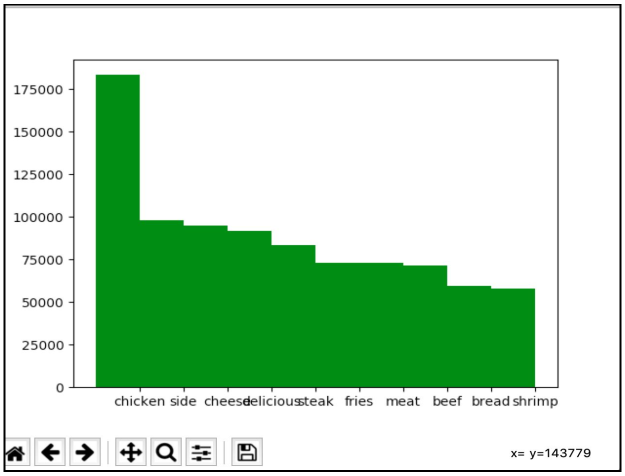


Figure 1: Frequently occuring food items

Collaborative Filtering has shown a lot of success when it comes to recommendation systems. We decided to incorporate Model based Collaborative Filtering on user to food ratings to find similar users and recommend food items for them. Though our recommendation system recommends food items for the chosen pair of users, our ultimate aim was to recommend a restaurant to visit. We thus, built a mapping of food to businesses based on the rating of each business and food items extracted from reviews of each business. We use this mapping to filter out businesses which serve those food items and recommend top k businesses which have high ratings for the recommended food.

DATASET

The dataset is from the Yelp Dataset Challenge. The raw data was preprocessed to extract a subset of the data to work with for our use case. We identified Las Vegas to be a good training ground due to the number of restaurants, visitors, reviews, and variety in the data. Exploratory data analysis was performed to get insights such as the average rating for a restaurant, popular cuisines in the city and percentage of missing data. Figure 2 shows the most popular categories in Las Vegas. Since the data provided by Yelp was already clean with minimal missingness, we did not impute any data.

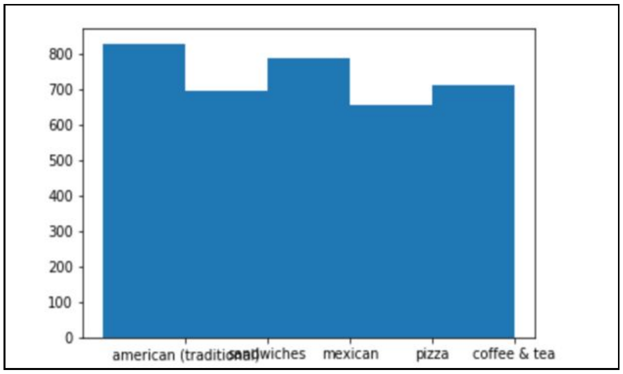


Figure 2: Frequently occuring business categories

The final dataset consists of 317240 users that have rated 28865 restaurants. To perform sentiment analysis, we used around 100,000 reviews.

METHOD

Figure 3 gives an overview of components of the system and modules involved in the flow of a recommendation. The two main components are Food Preference Extraction and Recommendation System. Each of these modules are further described in detail in this section.

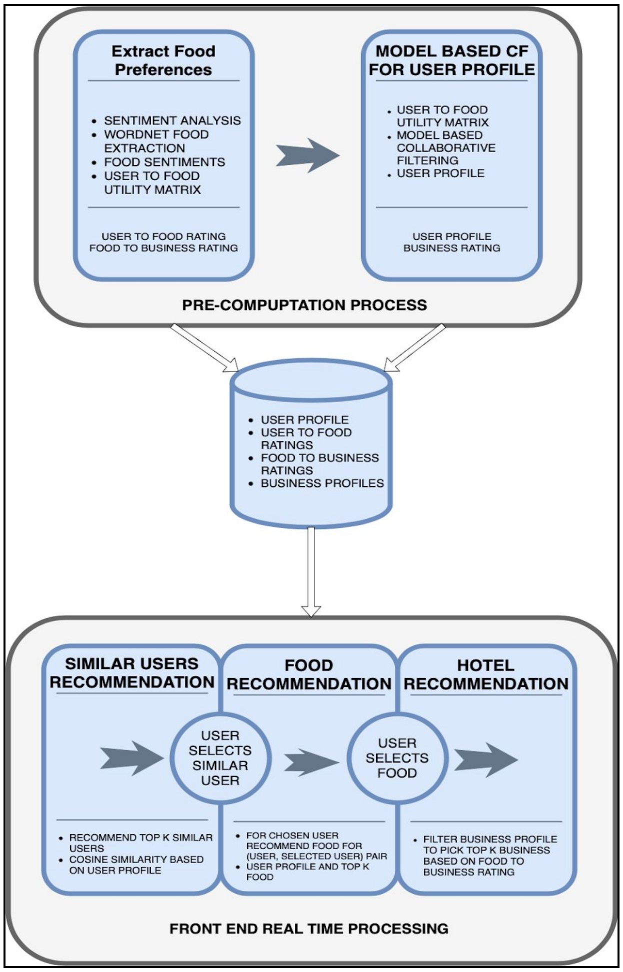

Extract Food Preferences
SENTIMENT ANALYSIS
WORDNET FOOD EXTRACTION
FOOD SENTIMENTS
USER TO FOOD UTILITY MATRIX
USER TO FOOD RATING
FOOD TO BUSINESS RATING
MODEL BASED CF FOR USER PROFILE
USER TO FOOD UTILITY MATRIX
MODEL BASED COLLABORATIVE FILTERING
USER PROFILE
USER PROFILE
BUSINESS RATING
PRE-COMPUTATION PROCESS
USER PROFILE
USER TO FOOD RATINGS
FOOD TO BUSINESS RATINGS
BUSINESS PROFILES
SIMILAR USERS RECOMMENDATION
FOOD RECOMMENDATION
HOTEL RECOMMENDATION
USER SELECTS SIMILAR USER
USER SELECTS FOOD
RECOMMEND TOP K SIMILAR USERS
COSINE SIMILARITY BASED ON USER PROFILE
FOR CHOSEN USER RECOMMEND FOOD FOR (USER, SELECTED USER) PAIR
USER PROFILE AND TOP K FOOD
FILTER BUSINESS PROFILE TO PICK TOP K BUSINESS BASED ON FOOD TO BUSINESS RATING
FRONT END REAL TIME PROCESSING

Figure 3: System overview

1 Food Preference Extraction

Food preference extraction is a key step in our pipeline which helps understand what the user likes and what cuisines the restaurant is known for. After data preprocessing is performed in the above method, we have a subset of reviews to be analyzed. We convert this into the following 3 step process - Food item extraction, Sentiment extraction, Food rating extraction. The reviews are split into sentences and each sentence is individually analyzed.

1.1 Food Item Extraction

For each sentence extracted from the review, we devise a method to extract all the food items mentioned by the user. We use ‘nltk’, a Natural Language Toolkit which provides a dictionary called wordnet containing words related to food items, to extract all food related words from the sentence. This step provides information on the food the user is referencing.

1.2 Sentiment Extraction

Sentiment Extraction is another key aspect of the food preference extraction pipeline once food extraction is complete. This module captures the sentiment of a user towards a food item - positive or negative. We make use of the sentiment analysis feature of TextBlob, a simplified text processing module, to get the polarity of what user is trying to say. This outputs a score in the range of -1 to +1, where -1 signifies negative, +1 positive and 0 neutral sentiments respectively. We use this information to assign an appropriate rating for user to food in the review.

1.3 Food Rating Extraction

For every review, we have an associated rating as mentioned in the Dataset section. So when assigning rating of each user to a specific food, we use the rating given for the review being analysed, list of all food items being spoken about and its sentiment. We take a product of sentiment and rating value to get the final rating of a user to specific food.

Once the user ratings to a food is extracted from reviews, there may be multiple entries of user ratings for a food across multiple reviews. There might also be instances where a user speaks positively about a food in one review and negatively about the same food in another review. To account for this, we take the average for a user and food pair, and provide a single average rating for user to food. This will be used in the utility matrix during the recommendation step.

2 Recommendation System

The recommendation system was implemented with three different approaches for collaborative filtering. The input to each of these algorithms is the result of the food preference extraction. A set of 1,745,248 tuples of the form <UserID, FoodItem, Rating> is split into training and testing sets.

A. User based CF

A memory-based approach where a subset of other users is chosen based on their similarity to the active user. A weighted combination of their ratings is used to make predictions for the active user. We used our own implementation of this algorithm using Pearson Correlation to find similar users with co-rated food items.

B. Matrix Factorization

One widely-used way of matrix factorization for recommendation system is singular value decomposition, which can be expressed as X = USV, where U and V are orthogonal matrices, and S is a diagonal matrix with the singular values of X on the diagonal. By the factorization, we extract the useful information from X into U, S and V, and then use their product as the estimates for the missing values in X. We used an implementation of this algorithm from the ‘surprise’ package in Python.

C. Alternating Least Squares (ALS)

ALS models the rating matrix as the multiplication of low-rank user and product factors, and learns these factors by minimizing the reconstruction error of the observed ratings. The unknown ratings can subsequently be computed by multiplying these factors. We used Spark MLLib’s implementation of this algorithm along with grid search to identify the optimal hyperparameters for the algorithm.

One of the main challenges was to compute similarity of each user with every other user. Due to the size of data, calculating this was too expensive, so we came up with a solution to prune the candidate neighbors for each user. This was achieved as follows :

1. Calculate top k food items for each user based on the ratings calculated.
2. Obtain list of unique businesses containing the food items.
3. Get list of food items pertaining to each of the businesses listed above. This will be a

smaller set compared to the original list of food items.

4. The final step is to obtain a list of users who have reviewed these food items and consider them as candidate neighbors for the given user.

The correlation for each user with their neighbors was then calculated using cosine similarity.

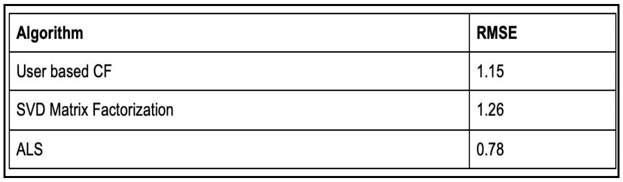
Next, for each pair of users, a paired set of food items are suggested based on common interests calculated from the CF utility matrix.

We then filter these food items to obtain top 5 food item combinations. This is achieved by calculating the sum of ratings for each food item pair and obtaining the top 5 food item pairs.

Finally, we suggest top 5 businesses for each food item pair. An inverted rdd mapping of food to business helps obtain list of candidate businesses and we further filter based on score for the business, along with a threshold for number of ratings given for the business.

RESULTS AND DISCUSSION

All experiments for the recommendation system were first executed on our laptops. We then setup a Spark cluster in the cloud with 3 nodes to achieve the reported results. Figure 4 shows a comparison of these results. Amongst the models we have built, the best performance was given by ALS with an RMSE of 0.78. The other models also have good performance with the testing RMSE between 1.1 and 1.3.



Algorithm	RMSE
User based CF	1.15
SVD Matrix Factorization	1.26
ALS	0.78

Figure 4: Collaborative filtering performance

Through this project, we have built a recommendation system that suggests users with similar food preferences to visit nearby restaurants together. The recommendations are provided in 3 stages based on the user's choice at each stage - first similar users, then food items and finally restaurants.

To verify if our suggestions are correct, we plan to use user feedback to evaluate our matches. After an analysis of this feedback, the algorithm to provide suggestions can be improved.

APPENDIX

The latest copy of code for the above mentioned methods are available in the following github repository, https://github.com/anishnarang/data_mining_project.git. This was implemented in a group of 3 members. Anish Narang implemented the initial preprocessing and analysis on the data set and implemented Collaborative Filtering. Akshay Adiga worked on Food preference extraction using nltk and textblob, and created intermediate mappings. Arjun Mohan worked on optimizing the process of finding similar users and recommending businesses from food items using PySpark.

REFERENCES

[1] Sentiment Analysis using TextBlob, https://textblob.readthedocs.io/en/dev/api_reference.html#textblob.blob.TextBlob.sentiment

[2] Model Based Collaborative Filtering using PySpark, https://spark.apache.org/docs/2.2.0/mllib-collaborative-filtering.html