Business Recommendations

CMPE 256 Spring 2019 Project Report
Team: T-Recs

Team Members:
Matthew Kwong <matthew.kwong@sjsu.edu>
Yi Lai <yi.lai@sjsu.edu>
Yuanjian Gao <yuanjian.gao@sjsu.edu>

1. Introduction

Whether it is eating, shopping, fixing one's car, going to the dentist, or any other activity, it is undeniable that there are an overwhelmingly large number of options to choose from even in one's local neighborhood. Our project aims to provide a solution to this problem, by using Google Local reviews to generate business recommendations. We also aim to provide users with a map tool with which they can search for businesses by specifying a category or location, with the ability to filter by average rating and distance. In conjunction with the generated business recommendations, we believe that this will help solve problems of indecision for users when they are considering where to bring their business.

In section 2 we describe the project design and implementation. Section 3 discusses in detail the methods we used for the analysis and evaluation of the machine learning portion of the project. Section 4 covers our conclusions and the lessons we learned through this project. Finally, section 5 contains a breakdown of the project work among us team members.

2. System Design & Implementation

2.1. Architecture

The whole system includes three main components: data preprocessing, the business search engine, and the business recommendation system. All components are written in Python, leveraging several libraries described later in section 2.4. The components and their interactions are described in Figure 1 below.

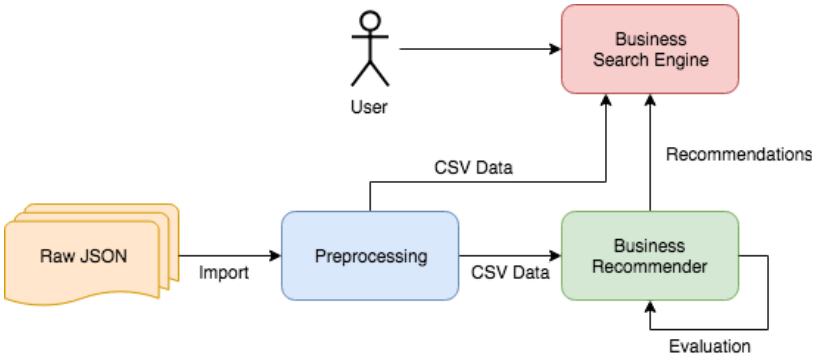

Figure 1. Project architecture

The primary purpose of the data preprocessing component is to transform the raw JSON data into a set of csv files, which are more suitable for handling by common data analysis tools in Python. For our project, we have decided to extract only the data pertaining to Santa Clara County. We will discuss this decision more in-depth in section 3.2.

Following data processing, the main function of the business search engine is to provide and intuitive interface for searching for a business. Using the business search engine, a user can specify business attributes such as location, category, average rating, number of reviews, or any combination of the aforementioned items, when searching for a business of interest.

Finally, the business recommendation system exists to provide users with recommended businesses based on their past reviews of other businesses. For our project, we tried several different algorithms and compared their results through cycles of evaluation in order to find both the optimal algorithm as well as the optimal algorithm parameters for the problem.

2.2. Algorithms

2.2.1. Business Search Engine

The business search engine is responsible for displaying businesses that satisfy a location category, average rating, and/or review count criteria. One challenge is that two similar businesses may be classified into slightly different categories in the raw data. For example, given two dessert shops, one may be categorized as “Dessert Shop” and the other as “Dessert Restaurant”. A second challenge is that the majority of the businesses are tagged by several categories. For the sake of the end user, we certainly want to ensure that we display all businesses that are similar to the target category. Therefore, we need to find a way to expand each category to include closely associated categories. In order to complete this expansion, we designed our own algorithms based on K-Means clustering. The procedure is given below:

1. generate a sparse matrix using one-hot encoding of the category
2. use the K-Means clustering algorithm to group these businesses into 220 subtypes based on their categories’ information, and obtain a list of categories for each subtype
3. use the K-Means clustering algorithm a second time to group the 220 subtypes into 30 higher-level types by text clustering
4. when a category is inputted, predict the type ID (0-29) using the trained model
5. expand the input category to all the categories belonging to the same type ID, then obtain the list of businesses that fall into those categories, and which exist within a certain threshold distance from a given location

2.2.2. Business Recommendation System

For the recommendation system, the first step was to try several different algorithms, tuning the parameters for each until a (near) optimal evaluation metric value was reached. We used the following algorithms, all provided by the Python Surprise library.

- SVD
- KNN basic (both user- and item-based)
- KNN with mean-centering (both user- and item-based)
- KNN with z-score normalization (both user- and item-based)
- Co-Clustering

We chose SVD because it performed well in previous recommendation system tasks. We chose KNN and its variations because it is an intuitive solution to the problem of generating

ratings-based recommendations, and because we were interested in seeing if it could outperform SVD. Finally, we chose to also include co-clustering as an alternative algorithm, as a learning experience.

2.3. Technologies & Tools

Following is a list of Python libraries used in each of our components, as well as what each was used for. We also used Jupyter Notebook as the platform for running our Python code.

2.3.1. Data Preprocessing

- **json:** for reading the raw JSON data files
- **pandas:** for handling data using its versatile Dataframe class
- **csv**: for writing data to a csv file
- **ast:** for converting the string dictionary objects to a Python dictionary, and for solving the unicode format problem

2.3.2. Business Search Engine

- **pandas:** for data analysis, such as calculating the number of reviews for each business, the average rating for each business, and also sorting, splitting, and merging the data
- **sklearn:** for category expansion, we used the **K-Means** algorithm and the **Calinski Harabaz Score** metric from sklearn
- **folium:** for showing a map on which to visualize the business search results or recommendations

2.3.3. Business Recommendations

- **pandas:** for data analysis & parsing input files
- **surprise:** for collaborative filtering and other algorithms, which handily expect data to be input as (user ID, business ID, rating) triplets
- **matplotlib:** for visualization of the results

2.4. Visualizations

2.4.1. Location Distribution

Figure 2 shows the distribution of locations for the businesses in the dataset. The left diagram shows the top 25 countries as represented in the raw business data. Since the vast majority of businesses reviewed were in the U.S., we also looked at the top 25 states, which is shown in the diagram on the right.

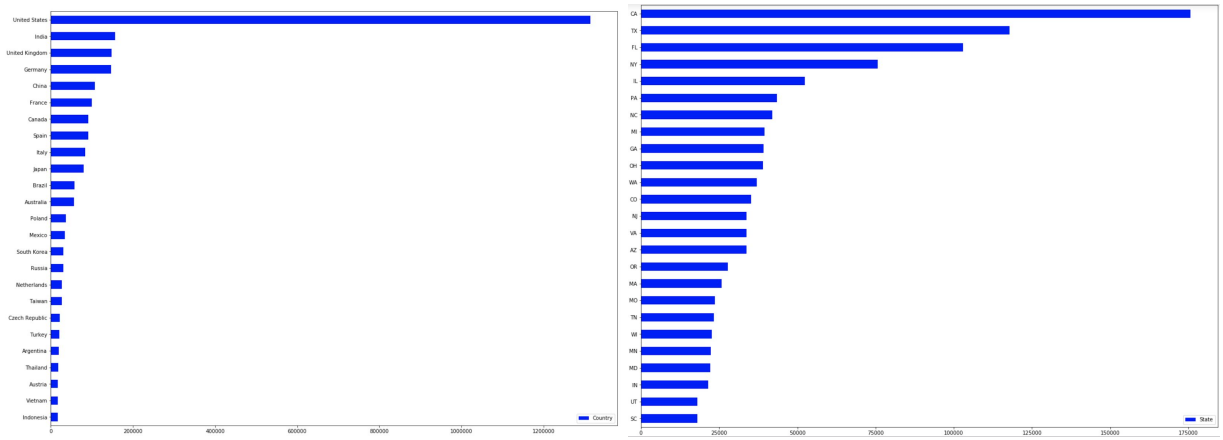

Figure 2. Business location distribution

2.4.2. Search Engine Category Expansion

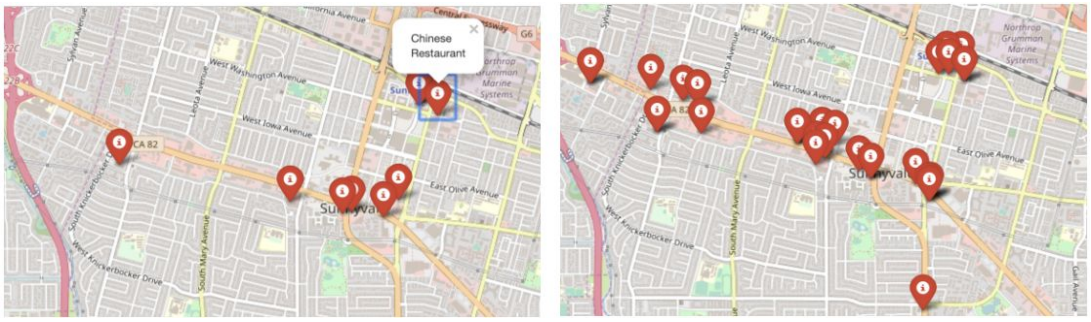

Figure 3. Category expansion

Figure 3 shows the effect of the category expansion. If we input “Chinese Restaurant” as the category filter, we only obtain several target businesses (left). After applying our category expansion algorithm, more businesses with similar categories are included in the target set, which results in a broader higher-level category such as “Asian Restaurant” in this case. Such category expansion drastically improves the quality and ease of searching for businesses.

3. Experiments & Evaluation

3.1. Dataset

The dataset our group used is the Google Local review data from the University of California San Diego. The raw data contains reviews of businesses from Google Local (Google Maps), provided as 3 JSON files. Due to the limitation of our hardware, we decided that we could not use the entire data. Instead, we decided to use the data pertaining to Santa Clara County, which also removes language issues and ensures that the problem is relevant to us.

- places.json (11m+ records): business ID, name, price, address, phone, hours, GPS.
- Users.json (4m+ records): user ID, name, current place, education, jobs, previous places, user name. The current place and the previous place are the present and past

locations of the user, respectively. Aside from the ID and name, most of the attributes are missing values, possibly due to privacy issues.

- Reviews.json (3m+ records): user ID, business ID, rating from 1 to 5, user name, comment, categories (a string representing a list), date, and unix timestamp.

3.2. Preprocessing

All three raw JSON files, and particularly the review data, were much too large for our limited hardware to handle, likely to cause an out-of-memory error or simply take much too long to process if we were to attempt to use the entire dataset. As mentioned previously, we therefore extracted only the data from Santa Clara County, performing feature selection and feature transformation to obtain the attributes we could use.

For the place data, we kept the business ID, name, GPS coordinates (splitting it into two new attributes - latitude and longitude), and address. The address was used not only to display on the map, but also to filter the businesses by city.

For the user data, we only kept the user ID and the current location information.

For the reviews data, kept the user ID, the business ID, the rating, and only the unix timestamp as it would be easier to compare/sort than the raw date. We also extracted the categories list into a separate csv file with its corresponding business ID for one-hot encoding. The IDs, rating, and timestamp were used for training our model and performing verification.

At first, in order to reduce the data size, we extracted data from the place dataset, as it was the easiest of the three from which we could extract only Santa Clara County data. We accomplished this simply by keeping all businesses whose city was one of the 15 in Santa Clara County. We used the JSON library to read the JSON file and convert the list of JSON objects into a pandas Dataframe, for ease of processing and exporting to csv for future use. We followed the same procedure for the user dataset, extracting only the user ID and the current place of users whose current place was one of Santa Clara County's 15 cities.

Second, we used the business data to obtain the list of unique business IDs, representing all the businesses in Santa Clara County. We used this list to extract the reviews of all those businesses, as well as their categories, saving them to two separate files.

Finally, since most users did not have a current place specified as mentioned previously, we needed a way of expanding the user list by inferring which users likely lived in Santa Clara County. We accomplished this by counting the percentage of reviews written by user, which were written for Santa Clara County businesses. By setting a threshold rating count and then only considering users whose Santa Clara County reviews surpassed that threshold, we were able to add many more users to the user list.

3.3. Methodology

3.3.1. Business Search Engine

There are a total of 1573 distinct categories in the raw data, with most businesses belonging to more than one category. Since we want to filter the businesses by category, we

cannot use the original 1573 categories directly, since most of the businesses that would intuitively fall into the same category are separated by slight category differences. Thus, our motivation is to group the 1573 categories into clusters, where the cluster number K is key factor. We tried K from 3 to 300 and used the Calinski-Harabaz score as shown in Figure 4:

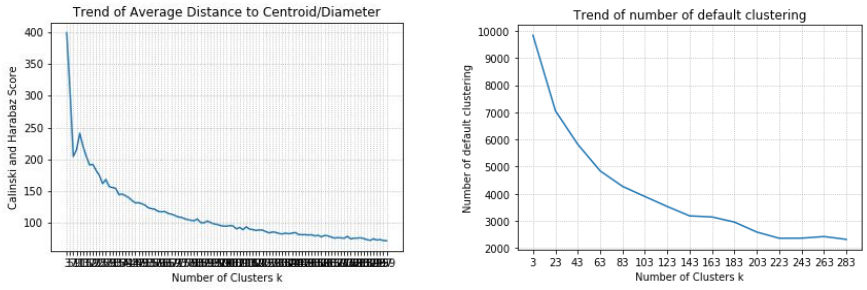

Figure 4. Calinski Harabaz score && number of Default clustering with K

However, since the data is uneven (some categories like “Restaurant” have a large amount of businesses belonging to it, while other categories like “Plumber” have far fewer), a smaller K is a benefit to larger categories, but most of the small categories end up in the default cluster, which means that a small K is not able to group small categories well. When we increase K, the small categories are clustered better, but businesses that are supposed to belong to the same category are split apart. In order to solve this problem, we first use a large K = 220 (which has the smallest size of default cluster) to guarantee fewer categories falling into the default cluster. Then, we get a list of categories (text) for each group and use K-Means to cluster the 220 groups into 30 supertypes.

3.3.2. Business Recommendations

As mentioned previously, the first step in generating business recommendations was to try several different algorithms, evaluating them to find an optimal parameter set. To do this, we ran the GridSearchCV() function provided by the Surprise library, in order to efficiently test multiple combinations of parameters, using 5-fold cross-validation, RMSE and MAE to evaluate them. The results are shown below in Figure 5.

Algorithm	RMSE	MAE	Parameters
SVD	1.000104164	---	lr_bu=0.005, lr_bi=0.005, lr_pu=0.005, lr_qi=0.001, reg_bu=0.05, reg_bi=0.02, reg_pu=0.05, reg_qi=0.05
KNN basic user	1.102765627	0.853629175	k=35, min_k=5
KNN basic item	1.106378672	0.8514354132	k=35, min_k=5
KNN z-score item	1.107823041	0.852417666	k=40, min_k=7
KNN means user	1.108768277	0.8420876861	k=50, min_k=5
KNN means item	1.111083018	0.8487991044	k=50, min_k=7
KNN z-score user	1.111791521	0.8393765511	k=45, min_k=7
Co-Clustering	1.124588327	0.8544638485	n_cltr_u=3, n_cltr_i=3

Figure 5. GridSearchCV results

SVD is clearly the best algorithm, outperforming the others by a full 0.01 RMSE. Thus, we chose to move forward with SVD. For the next step, we generated the “anti-testset” - all combinations of user-business pairs NOT in the training data - then applied some modifications to the raw ratings, and compared the results of running SVD on each set of ratings. We used

the raw ratings, the mean-centered ratings, the timestamp-normalized rating described below in Figure 6, and finally ratings that were both mean-centered and timestamp-normalized.

$$\hat{r}_i = r_i \cdot \frac{t_i - t_{min}}{t_{max} - t_{min}}$$

Figure 6. Timestamp normalization formula

The intuition behind timestamp normalization is that more recent reviews should have a greater influence/weight on the final rating. Therefore, by multiplying the ratings times the min-max normalized timestamp, we slightly adjust the ratings such that more recent ratings are proportionately higher than older ratings.

3.4. Graphs

Figure 7 below shows the comparison of algorithms run using different parameters. RMSE and MAE are the axes, so in all graphs except SVD, the data point closest to the lower left corner represents the best parameter pair, which has been marked and corresponds to Figure 5.

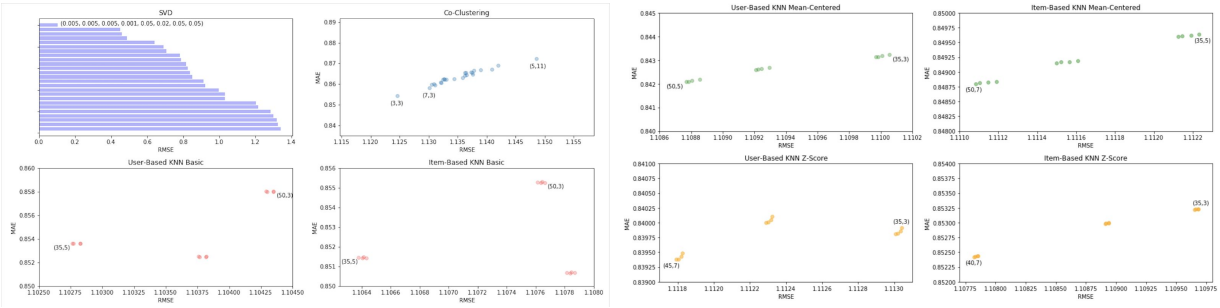

Figure 7. RMSE and MAE comparison of algorithms

3.5. Analysis of Results

3.5.1. Business Search Engine

After two rounds of clustering using K-Means (K1 = 220, K2 = 30) we obtain the table below in Figure 8. The first column is the supertype ranging from 0 to 29, and the second column is the subtype ranging from 0 to 219. The last column is the category list. Each time we input a category (e.g. “Church”), we return the corresponding super type (21 in this case). Also, because the second round of clustering is based on text, we can use the trained model to predict the supertype for any input word or even sentence. For example, if we input “where is the nearby Bank”, we get super type 25, which clearly contains bank-related categories.

super_types	subtype	categories_lists
21	18	['Church', 'Religious Organization', 'Abundant Life Church', 'Buddhist T
21	159	['Association or Organization', 'Culture', 'Entertainment', 'Jain Temple'
22	5	['Apartment Building', 'Apartment Complex', 'Property Management C
22	83	['Apartments', 'Apartment Rental Agency', 'Building']
22	106	['Apartment Rental Agency', 'Apartment Building', 'Furnished Apartme
22	122	['Apartment Building', 'Apartment Complex', 'Apartment Rental Agenc
23	77	['Optometrist', 'Eye Care Center', 'Optician', 'Contact Lenses Supplier',
23	98	['Optometrist', 'Contact Lenses Supplier', 'LASIK Surgeon', 'Ophthalmo
24	32	['Corporate Campus', 'Software Company', 'Cell Phone Store', 'Comput
24	51	['Extended Stay Hotel', 'Motel', 'Airport Shuttle Service', 'Boutique Hot
24	68	['Software Company', 'Marketing Consultant', 'Professional Services', '
24	123	['Movie Theater', 'Computer Consultant', 'Conference Center', 'Learnir
24	125	['Laundromat']
24	139	['Tax Preparation Service', 'Bookkeeping Service', 'Certified Public Acc
24	150	['Motel', 'Lodge']
24	192	['Financial Planner', 'Loan Agency', 'Marketing Consultant']
25	36	['Bank', 'ATM Location', 'Loan Agency', 'Check Cashing Service', 'Profes
25	66	['Credit Union', 'Mortgage Lender', 'ATM Location', 'Federal Credit Uni
25	78	['ATM Location', 'Bank', 'Mortgage Lender']
25	157	['Mortgage Lender', 'Mortgage Broker', 'Real Estate Agency', 'Loan Age
25	165	['Real Estate Consultant', 'Real Estate Developer', 'Commercial Real Es
25	182	['Bank', 'Mortgage Lender']

Figure 8: Parts of The final Type Cluster Result

3.5.2. Business Recommendations

An example of the top 20 recommendations for a user is given below in Figure 9. The recommendations are given for the algorithm run using the four types of ratings.

Rank	Raw	Time	Mean	Mean + Time
1	FCC Collision Centers	Public Storage	La Dolce Velo	Vasona Lake County Park
2	KAL Financial	Public Storage	California Playground Builders	Dittmer's Gourmet Meats and Wurst-Haus
3	Lexus of Stevens Creek	Public Storage	Dr. Martin Luther King, Jr. Library	Tootsie's At The Stanford Barn
4	European Wax Center - Mountain View	Public Storage	Gulzaar Halaal Restaurant & Bakery	Chromatic Coffee
5	Golden State Appliance Repair	Berg Injury Lawyers	Tootsie's At The Stanford Barn	Dr. Martin Luther King, Jr. Library
6	Aborn Properties	Public Storage	Tee Nee Thai	Quest Diagnostics
7	Gulzaar Halaal Restaurant & Bakery	Atlas Trillo Heating & Air Conditioning	Sunnyvale Public Library	Gaku Restaurant
8	Tin Pot Creamery	Public Storage	HSC Electronic Supply	Patxi's Chicago Pizza
9	Auto Care Specialists	Stevens Creek Subaru	Artisan Wine Depot	Trader Joe's
10	Public Storage	European Wax Center - San Jose The Plant	Summit Store Inc.	California Playground Builders
11	DataRetrieval Data Recovery Service San Jose	Public Storage	Patxi's Chicago Pizza	La Dolce Velo
12	Bumble	Public Storage	Mountain View Center for the Performing Arts	Flames Coffee Shop of San Jose
13	Pelle Heating & Air Conditioning	OnRevenue.com	Cafe Venetia	Los Gatos Creek County Park
14	Public Storage	Atlas Lock Change	Le Papillon	Supercuts
15	Any Water Sports	Delicious Crepes Bistro	Asena Restaurant	Starbucks
16	Sheila's Hair & Makeup	Kiddie Kountry Pre-School	Mozilla	San Jose BMW
17	Dr. Martin Luther King, Jr. Library	Schroeder's Garden Design	First Republic Bank	Sunnyvale Farmers Market
18	Kiddie Kountry Pre-School	First Bay	Chromatic Coffee	Summit Store Inc.
19	Peninsula Dental Excellence	Smythe European Maybach	Alexander's Steakhouse	Bombay Cash & Carry
20	OnRevenue.com	KAL Financial	Philz Coffee	Imagine Hair Salon

Figure 9: Example Recommendations

Businesses that appear in at least two lists are color-coded. One business, the Dr. Martin Luther King Jr. Library, appears in three of the four lists! Notably, the timestamp-normalized recommendations contain the fewest common businesses, and instead contain several instances of Public Storage, which is questionable. On the other hand, the mean-centered ratings perform much better, with the top 5 also being recommended by another method.

4. Discussion & Conclusions

- We had to drastically cut down on the size of the dataset in order to be able to handle it, and this proved to work out well for our team, since it was small enough to manage, while still large enough to provide a challenge when running certain algorithms.
- The dataset is so rich that we were unable given time constraints to make full use of certain attributes, i.e. the hours for a business, the previous locations of a user, or the review text. A larger scoped project would certainly benefit from such analysis.

- The novel timestamp normalization did not seem to increase the quality of the recommendations, especially compared to mean-centering.
- Overall, we enjoyed working with a new dataset, coming up with a problem, and approaching it using the techniques we learned in class.

5. Project Plan & Task Distribution

- Yuanjian Gao: Data Preprocessing
    - Leveraged desktop virtual memory to improve processing performance
    - Imported raw JSON files, handling unicode values gracefully
    - Feature selection: determined which attributes we would need
    - Feature transformation: split GPS into latitude and longitude; one-hot encoding for business categories
    - Exported data to several separate csv files
    - Data cleaning: used regular expressions to fix malformed GPS coordinates
- Yi Lai: Search Engine
    - Merged the review, business, and category data together to obtain a complete business record
    - Calculated the review count and average review rating for each business
    - Designed the algorithm that highly improved the coverage and accuracy of business search when using category information
    - Realized local business search engine with filters of location, minimal review count, minimal review rating, category and their combination
    - Visualized the search result on a map
- Matthew Kwong: Recommendations
    - Selected algorithms and optimized parameter combinations using GridSearchCV
    - Evaluated model using RMSE, MAE, and 5-fold cross-validation
    - Used the model and anti-testset to generate the top 20 recommendations for each user, exporting them to a csv file
    - Repeated the recommendation process for multiple rating types: raw, mean-centered, timestamp-normalized, and both mean-centered and timestamp-normalized