Traveling Location Recommendation System

I. Introduction.

In this project, I will use machine learning techniques to see what city a traveler would like to visit. The audience of this program will be travel agencies, who can take advantage of the program and give recommendations to their customers, i.e. the travelers. This will help the agencies target their customers, and could even help them explore new tourist attractions.

You can think of this program as a content-based recommendation system, and what the users have to do is simply rate the cities they've been to. The program will first ask the user to input ratings for 5 locations of his/her choice. The program will analyze the locations, exploring nearby venues and finding the pattern. After the program gets a hint of what the user likes, it will apply the criteria to other cities, and see if the user will like it.

II. Data.

I will leverage the power of Foursquare, in which detailed location data can be found. I will use the url like https://api.foursquare.com/v2/venues/explore to explore nearby venues of the locations that the user input. The output we are interested will be each venue's name, its coordinates, and the category it belongs to. This can be further analyzed and be turned into useful information that I am looking for. See the methodology section for more details.

III. Methodology.

After categorizing the data I got from Foursquare, I can ultimately generate a venues metric that lets me know what venues are nearby the location. The categorizing process consists of three steps. First, clean the information we get from Foursquare to get rid of unnecessary information as well as to make the output readable. This will be done by the function *getNearbyVenues()* in the program. Second, we want to use one hot encoding techniques to see whether or not each location has restaurants, shopping mall, bar… nearby. Third, we will group the venues by their locations. The *grouped* data frame in the program represents the result of this part.

Now, we can generate the user profile table by calculating the dot function of the venues' appearance frequency and the user's rating. This gives us an understanding of what the user likes. After having the resulting metric, I can see the user's preference, and use it to calculate the possibility that the user will like other locations.

Let's look at a simple example. Say Bob has been to Rouge Hill, Scarborough. Bob gave the place a rating of 10, which means he loved it. Now, the program can find data from Foursquare that tells us what are the most abundant venues in Rouge Hill, specifically, bars, Vietnamese restaurants, and coffee shops. With this information, I will boldly guess that Bob enjoys cities that have bars, Vietnamese restaurants, and coffee shops. Then, I will find cities that meet this criteria, and recommend it to Bob. Of course, this recommendation does not seem very convincing when it is based on only one rating. That's why I programmed it to ask for five. In fact, the more information we have about the user, the better recommendation we can give to him/her. This scale of this program can be expanded for more accurate predictions if needed in the future.

This program used the machine learning technique of a content-based recommendation system. This is because it is more suitable when we do not have a great database, compared to collaborative filtering. The reason to use a recommendation system is because the results it generates is easily understandable and very useful. Thus, it fits the need of the audience more easily.

IV. Results.

In the demonstration notebook, I used the data of five ratings, which shows as follows:

	Latitude	*Longitude*	*Rating*
0	*40.8766*	*-73.9107*	*8.0*
1	*40.7156*	*-73.9943*	*9.0*
02	*40.8519*	*-73.9369*	*8.0*
3	*40.8677*	*-73.9212*	*10.0*
4	*40.8236*	*-73.9497*	*7.0*

After calculations, we acquired the user profile. The profile is rather long, so I

will only list the first five here.

Accessories Store	*0.095238*
American Restaurant	*1.051732*
Arepa Restaurant	*0.095238*
Asian Restaurant	*0.160000*
Austrian Restaurant	*0.080000*

Finally, I ask what place the user might want to visit. Using the coordinates 40.877531, -73.905582 to demonstrate, the final result prints "The possibility that you like this place is 1.77 %".

V. Discussion.

The accuracy of this program might be limited because it only asked for five ratings. But since most parts of this program can be used on larger scales without changing the code, it can adapt to a bigger scale of analysis. This program can be combined with feedback systems provided by traveling agencies. Travelers, after travelling to a city, can rate the city and thus generate a big database. Traveling agencies can input the locations where they now have traveling plans, and see if the user will like the plan.

VI. Conclusion.

Using data from Foursquare, combined with simple calculations, the program can get a hint of what travelers like and recommend them to visit other locations. The scale of the program can be expanded easily and is suitable for corporate use.