Chicago Restaurant Recommender System

Table of Contents

Introduction

Anyone who has watched movies on Netflix or shopped on Amazon is very familiar with the recommendations provided as you browse. The systems underpinning these suggestions have become an important feature for online businesses wishing to provide the best possible user experience. The systems can reduce an overwhelming variety of choices to a small group of carefully chosen items. They can also introduce users to items that they would not have searched for on their own. By employing recommender systems, businesses hope to entice users to watch more movies and buy more products.

Foursquare is another company that provides recommendations to its users. For the sake of this project, however, we will assume this is not the case. In that world, Foursquare would be in severe jeopardy of losing users to competitors who can offer useful suggestions. As a starting point, we will create a recommender system for restaurants in the city of Chicago, Illinois, USA.

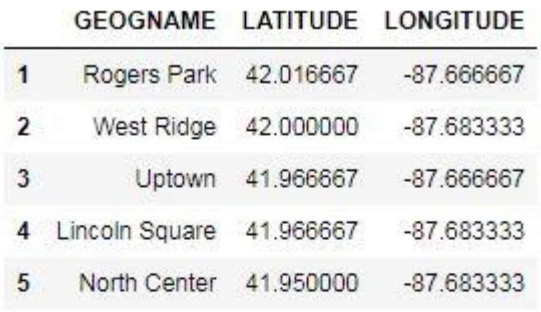
We will employ data from Foursquare and the city of Chicago to build a content-based recommender system. This is a method that builds profiles for each user based on their previous ratings. It then compares a prospective restaurant against the user's past experiences to estimate a rating. When you repeat this for several eateries within an area, the recommender system will provide a list sorted from the highest estimated rating to the lowest. The hope is that this will greatly improve customer satisfaction with Foursquare, so that the company can continue to be the leading location technology platform.

Methodology

Chicago Neighborhoods

The starting point for this task was with the city of Chicago. The city is well-known for its neighborhoods, officially called community areas. Many people tend to frequent restaurants within a handful of neighborhoods due to the logistical challenges of going to places far from home or work. With this in mind we have used the neighborhoods as features for finding recommendations.

While the city publicizes a rich assortment of data on its neighborhoods, none of them included the longitude/latitude information required for Foursquare API queries. We therefore collected and entered this information manually into a spread sheet of neighborhood census data, then deleted the extraneous columns.



	GEOGNAME	LATITUDE	LONGITUDE
1	Rogers Park	42.016667	-87.666667
2	West Ridge	42.000000	-87.683333
3	Uptown	41.966667	-87.666667
4	Lincoln Square	41.966667	-87.683333
5	North Center	41.950000	-87.683333

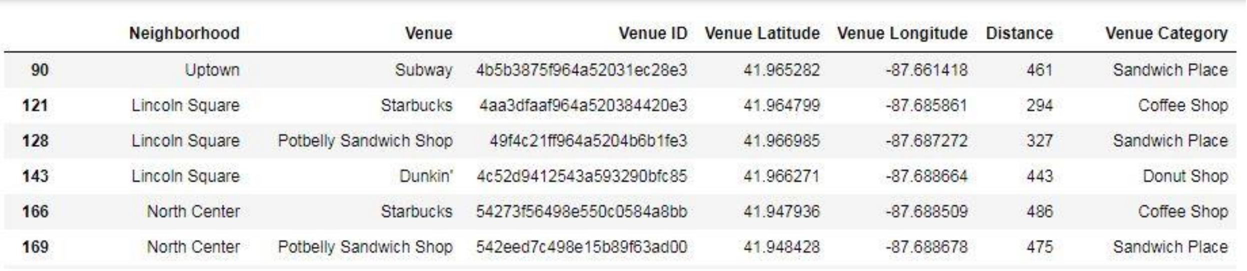
Foursquare Data

Over the years Foursquare has built an extremely rich dataset that is centered on venues and users. For each venue this set includes details such as the name, location and category. While Foursquare gathers a plethora of data on its users, most of that information is not available to the public. For this project, we are limited to identifying the users who liked each venue.

Building a Venue List

Working from our Chicago neighborhood data, we have iterated through each neighborhood and queried Foursquare for venues within it. We also collected the name, identification number, longitude and latitude for each venue. The most important piece of information for our purposes is the category. The reasoning is that if a user likes a lot of Thai restaurants, he is more likely to enjoy others as well. We limited our search results to those within the Food category and collected the subcategory for each. This became the second feature for generating recommendations.

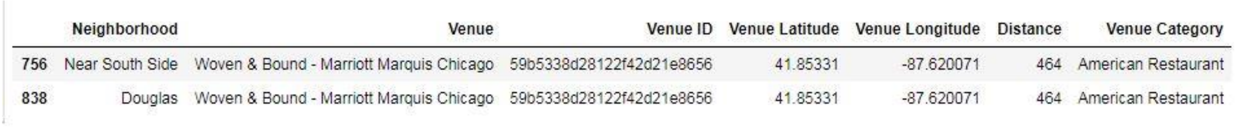
One unexpected aspect of the data was the presence of chain restaurants. While a user who likes one Starbucks will undoubtedly like another, we did not find value in this type of recommendation. Chain restaurants, especially for locations within the same city, will offer an almost identical experience. The purpose of this recommender system is to enhance the user experience. Including chains could skew results to try to repeat the same experiences. For those reasons, we have removed all venue names that appear more than five times.



	Neighborhood	Venue	Venue ID	Venue Latitude	Venue Longitude	Distance	Venue Category
90	Uptown	Subway	4b5b3875f964a52031ec28e3	41.965282	-87.661418	461	Sandwich Place
121	Lincoln Square	Starbucks	4aa3dfaaf964a520384420e3	41.964799	-87.685861	294	Coffee Shop
128	Lincoln Square	Potbelly Sandwich Shop	49f4c21ff964a5204b6b1fe3	41.966985	-87.687272	327	Sandwich Place
143	Lincoln Square	Dunkin'	4c52d9412543a593290bfc85	41.966271	-87.688664	443	Donut Shop
166	North Center	Starbucks	54273f56498e550c0584a8bb	41.947936	-87.688509	486	Coffee Shop
169	North Center	Potbelly Sandwich Shop	542eed7c498e15b89f63ad00	41.948428	-87.688678	475	Sandwich Place

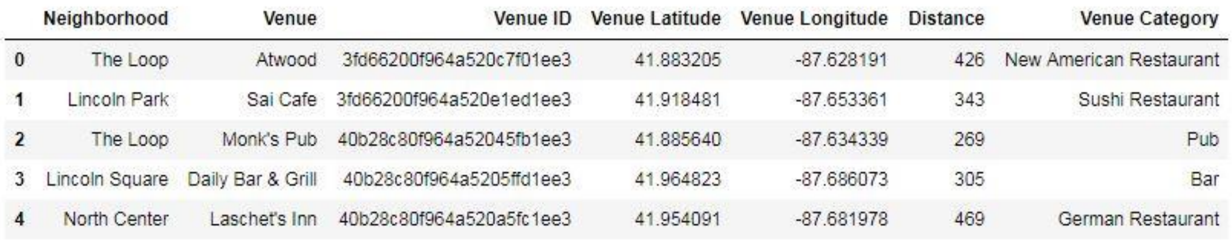
Another thing that could skew our data is the possibility of duplicates. Our Foursquare query used a radius from the center of each neighborhood. Since neighborhoods come in a variety of shapes and sizes, this creates the possibility of retrieving the same establishment for multiple neighborhoods. We therefore checked for duplicates based on the venue identification number.

We found sixty-one duplicates in our dataset. Because it would be difficult to accurately identify the neighborhood for each of these, we used the distance from the neighborhood center as a measurement. The first step was to sort our dataset on the venue identifier and distance. Then we removed all duplicates aside from the first entry. We used a sample entry to check before and after to confirm that only one remains.



	Neighborhood	Venue	Venue ID	Venue Latitude	Venue Longitude	Distance	Venue Category
756	Near South Side	Woven & Bound - Marriott Marquis Chicago	59b5338d28122f42d21e8656	41.85331	-87.620071	464	American Restaurant
838	Douglas	Woven & Bound - Marriott Marquis Chicago	59b5338d28122f42d21e8656	41.85331	-87.620071	464	American Restaurant

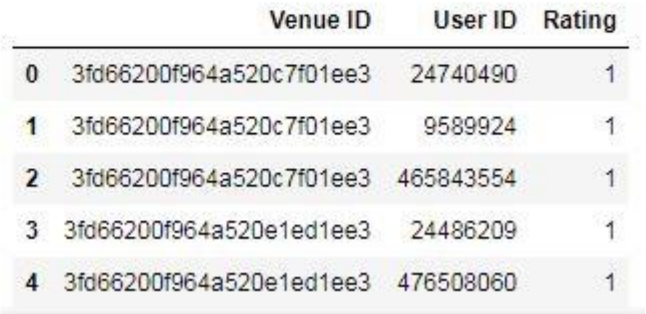
After the removal of chains and duplicates, we were left with a working dataset of Chicago restaurants.



	Neighborhood	Venue	Venue ID	Venue Latitude	Venue Longitude	Distance	Venue Category
0	The Loop	Atwood	3fd66200f964a520c7f01ee3	41.883205	-87.628191	426	New American Restaurant
1	Lincoln Park	Sai Cafe	3fd66200f964a520e1ed1ee3	41.918481	-87.653361	343	Sushi Restaurant
2	The Loop	Monk's Pub	40b28c80f964a52045fb1ee3	41.885640	-87.634339	269	Pub
3	Lincoln Square	Daily Bar & Grill	40b28c80f964a5205ffd1ee3	41.964823	-87.686073	305	Bar
4	North Center	Laschet's Inn	40b28c80f964a520a5fc1ee3	41.954091	-87.681978	469	German Restaurant

Building a User List

With our list of viable restaurants in hand, the next step was to build a list of users. As mentioned above, Foursquare limits the amount of data available to the public. For this system we were limited to finding the users who liked each of our venues. We looped through our list of venues and built a list of users and the venues they liked. We discovered that some venues have not been liked by any users, so we skipped those. However, those venues are still useful in our dataset as possible recommendations. Due to the limitations of the available data, we recorded the rating of each liked venue as one (1). Since we cannot access data on dislikes or how often a user has visited the venue, our results may not be as rich as they could be.

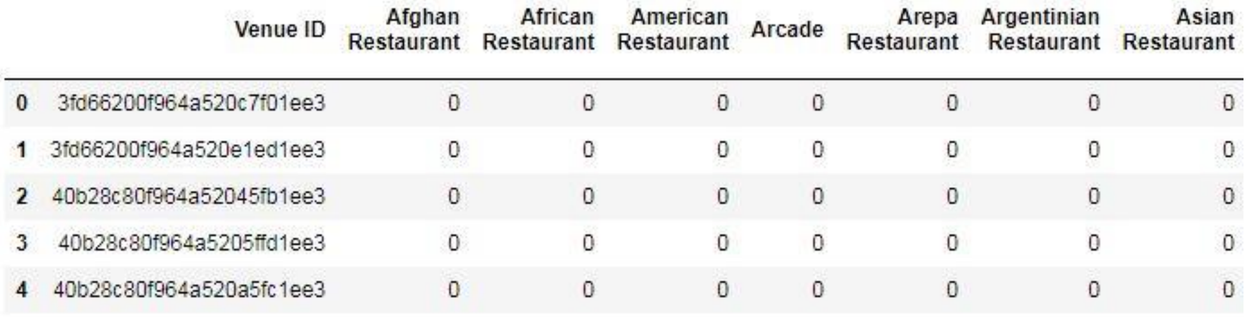


	Venue ID	User ID	Rating
0	3fd66200f964a520c7f01ee3	24740490	1
1	3fd66200f964a520c7f01ee3	9589924	1
2	3fd66200f964a520c7f01ee3	465843554	1
3	3fd66200f964a520e1ed1ee3	24486209	1
4	3fd66200f964a520e1ed1ee3	476508060	1

One-Hot Encoding

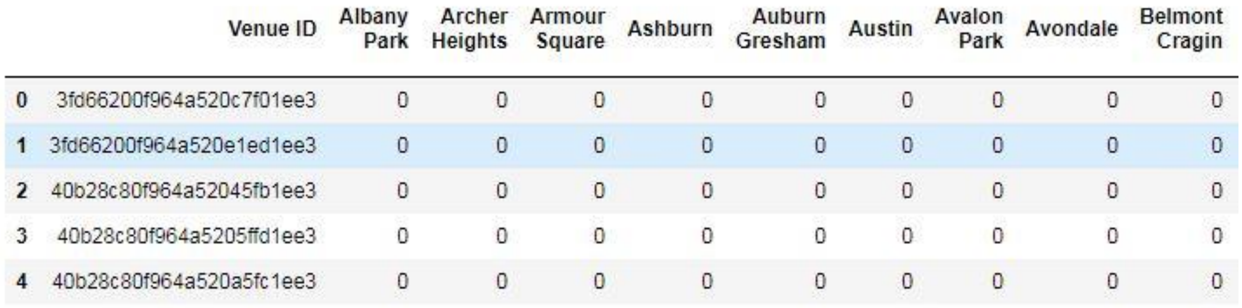
In order to use machine learning to produce our recommendations, we employed the one-hot encoding technique. This converts the features that we wish to build our model on – restaurant category and neighborhood – into numerical values based on user likes. That allows our model to easily compare venues to find those that are most similar to ones a user has liked in the past.

Since we chose to use two pieces of data, we performed this exercise twice. First, we created a new list of the venues where all the different types of restaurants became individual columns. Each venue was classified as one if it matched the type and zero if not.



	Venue ID	Afghan Restaurant	African Restaurant	American Restaurant	Arcade	Arepa Restaurant	Argentinian Restaurant	Asian Restaurant
0	3fd66200f964a520c7f01ee3	0	0	0	0	0	0	0
1	3fd66200f964a520e1ed1ee3	0	0	0	0	0	0	0
2	40b28c80f964a52045fb1ee3	0	0	0	0	0	0	0
3	40b28c80f964a5205ffd1ee3	0	0	0	0	0	0	0
4	40b28c80f964a520a5fc1ee3	0	0	0	0	0	0	0

Next, we repeated this process for the neighborhoods. In both cases we also added the venue identifier to the data and made sure we still had the same number as in our original venue dataset.

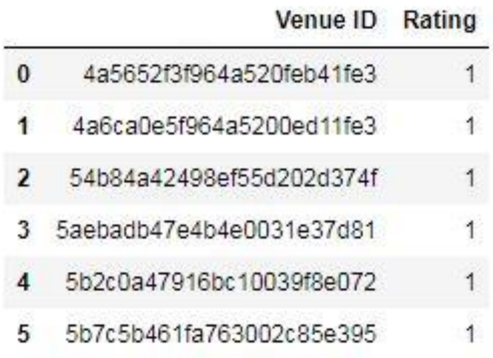


	Venue ID	Albany Park	Archer Heights	Armour Square	Ashburn	Auburn Gresham	Austin	Avalon Park	Avondale	Belmont Cragin
0	3fd66200f964a520c7f01ee3	0	0	0	0	0	0	0	0	0
1	3fd66200f964a520e1ed1ee3	0	0	0	0	0	0	0	0	0
2	40b28c80f964a52045fb1ee3	0	0	0	0	0	0	0	0	0
3	40b28c80f964a5205ffd1ee3	0	0	0	0	0	0	0	0	0
4	40b28c80f964a520a5fc1ee3	0	0	0	0	0	0	0	0	0

Finally, we combined both of the one-hot encoding lists for categories and neighborhoods into a single dataset.

Results

In order to obtain the results from our recommender model we chose a test user. To get a better result, we first identified users who have liked more than five venues. One drawback to content-based recommender systems is that users with few ratings produce poor results. Our minimum of five likes was to ensure a better outcome. Then we chose a user at random and extracted the venues they liked.



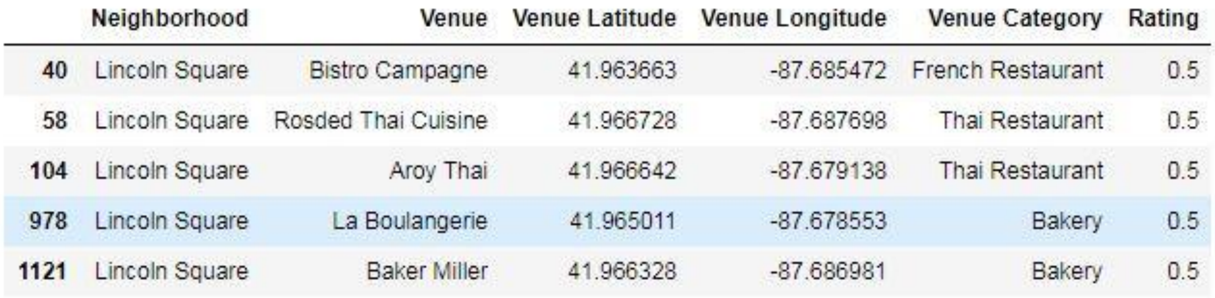
	Venue ID	Rating
0	4a5652f3f964a520feb41fe3	1
1	4a6ca0e5f964a5200ed11fe3	1
2	54b84a42498ef55d202d374f	1
3	5aebadb47e4b4e0031e37d81	1
4	5b2c0a47916bc10039f8e072	1
5	5b7c5b461fa763002c85e395	1

Next, we extracted those venues from our one-hot encoding list and removed the unnecessary columns.

To create a user profile, we then calculated a dot product of the one-hot encoding list and a vector of the user's ratings. This produced a weighted rating of each of our features. A reminder that the feature set is a combination of restaurant categories and neighborhoods.

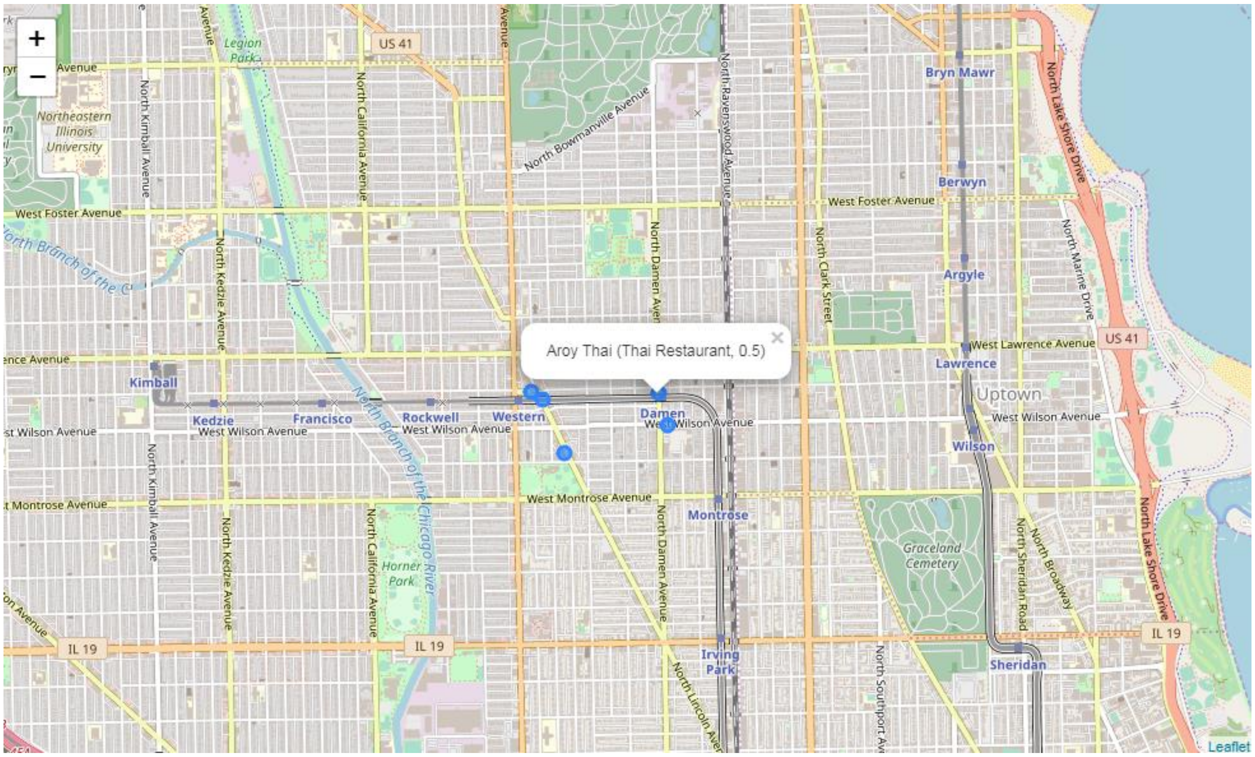
The next step was to create a clean dataset of our features.

Using the user's profile and the complete list of venue features, we took the weighted average of every venue based on the user profile and recommended the top five restaurants.

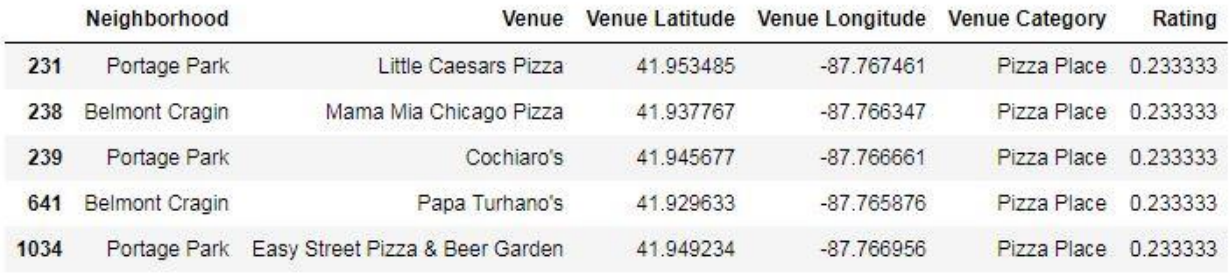


	Neighborhood	Venue	Venue Latitude	Venue Longitude	Venue Category	Rating
40	Lincoln Square	Bistro Campagne	41.963663	-87.685472	French Restaurant	0.5
58	Lincoln Square	Rosded Thai Cuisine	41.966728	-87.687698	Thai Restaurant	0.5
104	Lincoln Square	Aroy Thai	41.966642	-87.679138	Thai Restaurant	0.5
978	Lincoln Square	La Boulangerie	41.965011	-87.678553	Bakery	0.5
1121	Lincoln Square	Baker Miller	41.966328	-87.686981	Bakery	0.5

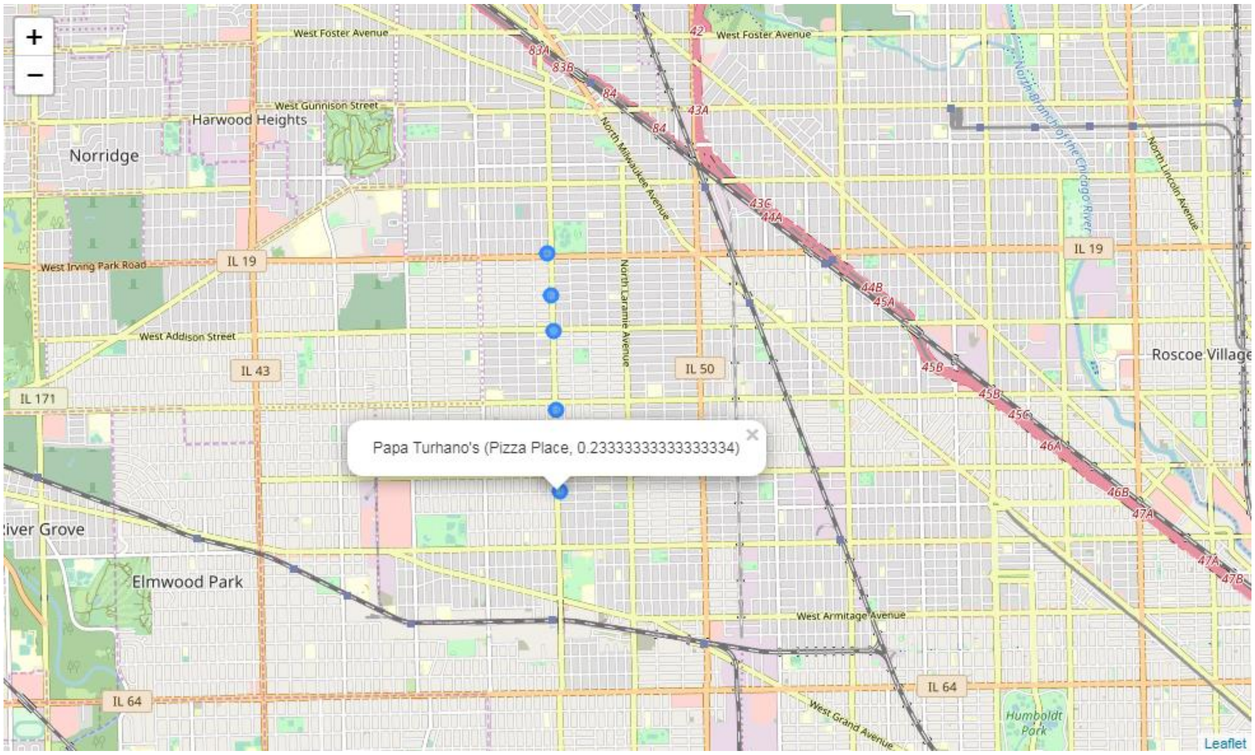
Now that we had our recommendations, we were able to plot them on a map with markers that indicated the venue name, category and expected rating.



We repeated the process for another user to show that the recommendations differ based on each user's profile.



	Neighborhood	Venue	Venue Latitude	Venue Longitude	Venue Category	Rating
231	Portage Park	Little Caesars Pizza	41.953485	-87.767461	Pizza Place	0.233333
238	Belmont Cragin	Mama Mia Chicago Pizza	41.937767	-87.766347	Pizza Place	0.233333
239	Portage Park	Cochiaro's	41.945677	-87.766661	Pizza Place	0.233333
641	Belmont Cragin	Papa Turhano's	41.929633	-87.765876	Pizza Place	0.233333
1034	Portage Park	Easy Street Pizza & Beer Garden	41.949234	-87.766956	Pizza Place	0.233333



Conclusion

The recommender system has proven useful for suggesting new eating venues based on each user's profile. Using restaurant categories allows the system to recommend other similar venues. Meanwhile, the neighborhood information makes it less likely that the system will make suggestions that are in far-flung parts of the city.

There are, however, some aspects of the system that could be improved. The limits of Foursquare pose the greatest challenge in this regard. For example, the initial plan also called for including the price points of each venue, e.g. cheap, moderate or expensive. Then we discovered a limit of five hundred requests per day for the queries required to fetch that data. Even after significantly reducing the number of neighborhoods, we exceeded this threshold.

The other major limitation of the Foursquare data is that we can only see when a user has liked a restaurant. The data provides no insight into whether the user disliked an establishment or how many times she has eaten there. That would be extremely valuable information for our system, as a simple yes/no rating does not provide much depth of knowledge. Of course, Foursquare does have access to all of this data and can, therefore, offer much better recommendations.

The exercise presented here offers a good example of using machine learning to provide recommendations to users. Past experiences and ratings allowed us to build a content-based recommender system for finding those suggestions. Unfortunately, our example is not as rich in data as we would like, and the recommendations provided may not be the best possible. Fortunately for Foursquare, in the real world they do have access to an extremely rich dataset that they can mine to offer much better recommendations.