# The battle of the Neighborhoods

By Matthias Sauer, 2019-05-07

As part of the Coursera IBM Data Science Professional Specialization, this report describes a recommender system for location data

## 1. Problem Description

Whenever I am coming to a new city, I have a hard time to access which location I should select. This is true for me as a tourist as well as if I am (like myself recently) moved away to a new location. Hence, I would like to have a service that recommends me a certain area in a city.

### 1.1. Background

Typically, I as a user know my own preferences and what I expect from an area. For instance, my personal preference is a lively region with a lot of shops, bars and recreational areas. A got medical system with many doctors may not be of much importance for me. Hence, it would be nice to get, based on a set of preferences, a recommendation for which area in a city to go for.

This is exactly what the described service does! The service is aiming at people visiting or moving a new city. They are presented a set of predefined personas they can personalize to match their own interest.

### 1.2. Business Model

The service can the capitalized by using targeted ads. As the type of the user is known by its persona, a pin-pointed marketing can happen. E.g. hotels in the most matching area can be advertised.

## 2. Data

In order to recommend a certain area of a city, two main types of data is needed:

### 2.1. User Preferences rating a set of attributes

The user preferences are taken from a user persona, which is the input to the approach. I as a user have to enter that the importance for e.g. a Bar is 5 and so on.

### 2.2. Location Data describing the quality of a region with respect to the set of attributes

The fundamentals of the location data is retrieved using the FourSquareAPI for a given area. Using the API, all entries in a circular area around the location center are collected and the number of entries for the various categories are counted.

## 3. Methodology

In order to allow a recommendation, the collected data is preprocessed.

### 3.1. Generation of the regional frequency map

At first, an area grid is created forming a regular grid of locations matching the size of the target size. For each part of the grid, the FourSquareAPI is used to get a list of locations returned as a JSON data entry. As such, the results from FourSqure need to be extracted. For each location entry, the main information used in the type of the location, e.g. a Bar or a Restaurant.

| | Neighborhood | Neighborhood Latitude | Neighborhood Longitude | Venue | Venue Latitude | Venue Longitude | Venue Category |
|---|---|---|---|---|---|---|---|
| 0 | Montréal 0,0 | 45.35 | -73.73 | Morsels By Mark | 45.351379 | -73.734732 | Dessert Shop |
| 1 | Montréal 0,0 | 45.35 | -73.73 | Plomberie Michel Labelle Inc | 45.350510 | -73.735576 | Construction & Landscaping |
| 2 | Montréal 0,8 | 45.35 | -73.49 | Montreal South KOA | 45.345979 | -73.489062 | Campground |
| 3 | Montréal 1,0 | 45.38 | -73.73 | Salsa Verdun - Bord de l'eau | 45.377136 | -73.731081 | Dance Studio |
| 4 | Montréal 1,4 | 45.38 | -73.61 | Boutique dollar et plus | 45.378564 | -73.614433 | Department Store |
| 5 | Montréal 1,6 | 45.38 | -73.55 | Cinéma Cineplex Odeon Delson | 45.381242 | -73.550163 | Multiplex |
| 6 | Montréal 1,6 | 45.38 | -73.55 | Tim Hortons | 45.383611 | -73.548364 | Coffee Shop |
| 7 | Montréal 1,6 | 45.38 | -73.55 | Pharmaprix | 45.384264 | -73.549216 | Pharmacy |
| 8 | Montréal 1,6 | 45.38 | -73.55 | SAQ | 45.382287 | -73.548842 | Liquor Store |
| 9 | Montréal 1,7 | 45.38 | -73.52 | Parc Montcalm | 45.378834 | -73.520790 | Soccer Field |

After extracting the types of locations for each gird entry, the data is grouped and the number of each location type for each location is kept. Afterwards, the locational data is preprocessed by normalizing the number of entries for each category using the MinMaxScaler. This is in order to avoid the different scales in the entries, while some elements like Coffee Shops tend to have a high occurrence, other entries like specialized restaurants have fewer entries. Hence, in order to compare the scale, the numbers are normalized for each categories over all grid cells.

### 3.2. Building the User Persona

Building the user persona is done using a simple python array matching the name of the location and a score. For instance:

```
userPreferences = [];
userPreferences.append(['American Restaurant',4])
userPreferences.append(['Bar',11])
userPreferences.append(['Coffee Shop',10])
```

With the help of helper functions, the preference list is than matched to the scoring of the location data. For instance:

| | LocationType | Score |
|---|---|---|
| 0 | American Restaurant | 4.0 |
| 1 | Art Gallery | 0.0 |
| 2 | Asian Restaurant | 0.0 |
| 3 | Athletics & Sports | 0.0 |
| 4 | Auto Dealership | 0.0 |

### 3.3. Recommendation System

Then, the normalized data can be used in a recommender system by multiplying the user profile with the normalized gird data. As a result, a preference score is obtained for each grid entry as shown in the following example:

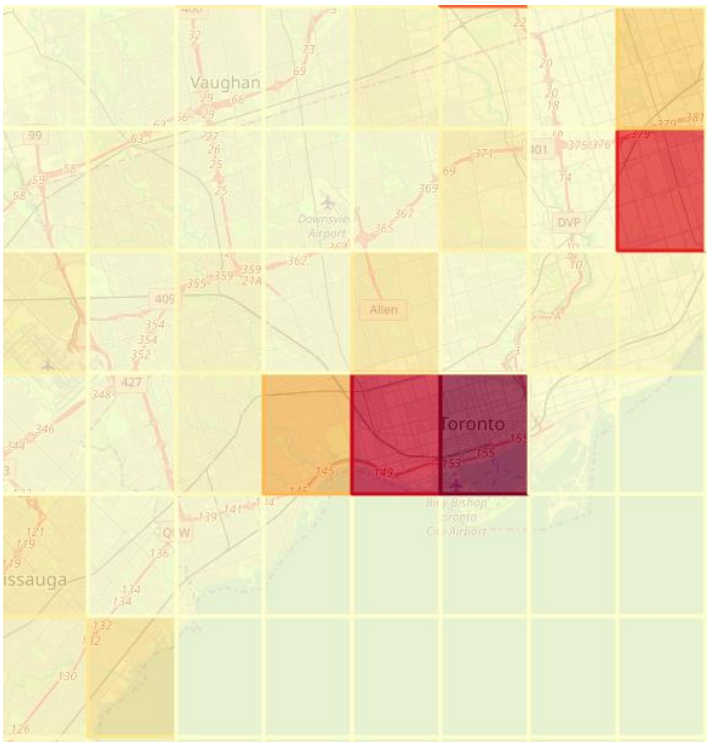
:

| | Neighborhood | Score |
|---|---|---|
| 0 | Toronto 0,0 | 0.00000 |
| 1 | Toronto 1,0 | 5.91716 |
| 2 | Toronto 2,1 | 5.91716 |
| 3 | Toronto 2,2 | 20.56213 |
| 4 | Toronto 3,0 | 0.00000 |

Hence, a recommendation score is computed by the recommender system and can be displayed to the user.

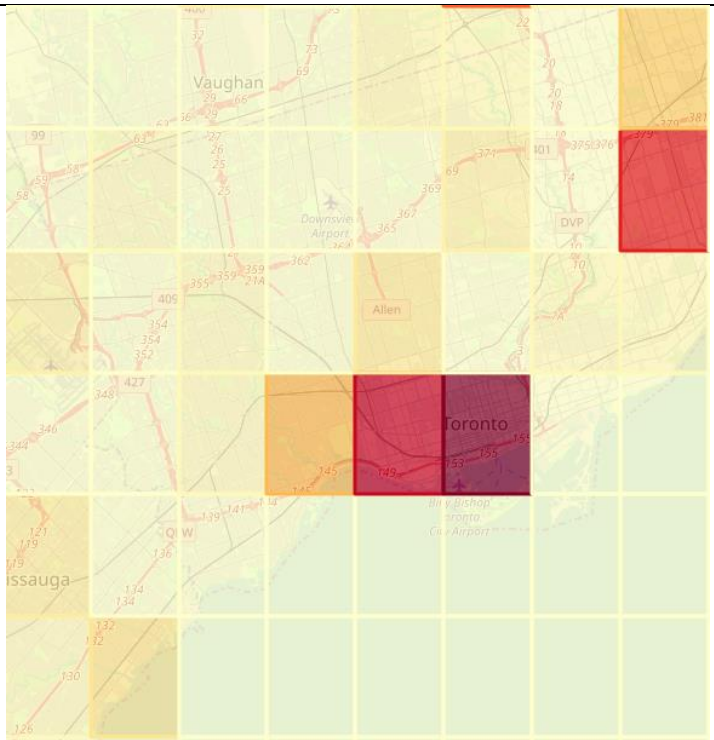
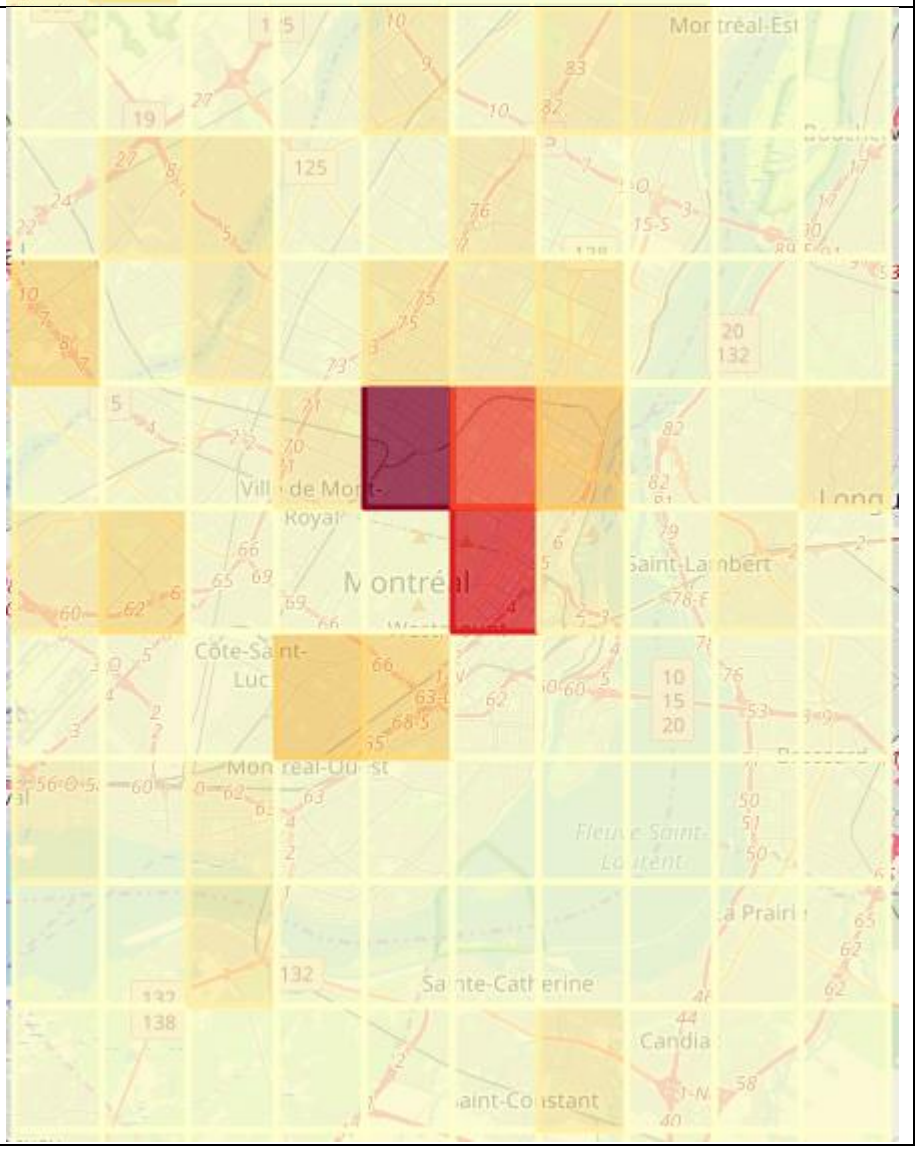
## 4. Results

Finally, the grid along with the score can be displayed on a map using folium. In order to visualize, a heatmap approach is used where an intensive red symbolizes a good score. An example is shown as follows:




## 5. Discussion

A very typical observation is that interesting regions tends to be in the city center as the density of interesting locations is typically higher than in an urban area. This is true for e.g. Toronto or Montreal.

| | |
|---|---|
| Toronto |  |
| Montréal |  |

## 6. Conclusions

The developed service allows to recommend regions of an area based on a set of user given preferences. Using the foursquare API, the locations of a grid are extracted and rated based on the preferences. Finally, the results are displayed nicely on a map allowing the user to choose its destination area.