Finding a good neighborhood in New York City

Ivan Souza

September 12, 2019

1 Introduction

1.1 Background

Very often, people need to move from one place to another. A new job, raising a family and college admission, for example, are some of the most common reasons for moving around. A change can also take place due to external problems: armed conflicts, poverty, natural disasters, political persecution, etc. But, when arriving at the new location, a question that may eventually arise is: where is the best place to live in this city?

Each individual has unique preferences and needs, which may vary over time, but it's reasonable and legitimate to think that everyone would like to live in a neighborhood that best suits their current expectations. A married person with children may prefer to live closer to where there are more schools and parks, for example. A young single person may prefer somewhere better served by public transport. A couple with no children may prefer to live near restaurants. If the couple is Italian, probably their favourite restaurants would be Italian, not Indian, for instance. The desired combinations of amenities are virtually endless.

Living close to places that are most compatible with current needs and preferences means maximizing personal and familiar happiness. From a philosophical point of view, the pursuit of happiness has always been the subject of study by great philosophers, from Aristoteles, through Kant to Stuart Mill, just to name a few.

Thus, in addition to the philosophical aspect, and now dealing with a more practical and rational approach, moving from one place to another can result in more or less personal and / or family problems. Lower work productivity (or even unemployment), emotional instability, disagreements with spouse / children, among other problems, can be related to the non-adaptation to a location due to the lack of essential structures needed by the individuals or families. On the other hand, good adaptation means greater personal and familiar fulfillment.

It is true, however, that happiness depends on many other aspects that are not related to living in a good neighborhood, but all these aspects are beyond the scope of this work. Having said that, the question to be answered by this project is: **Considering a person's needs, which New York City (NYC) neighborhoods would be most compatible with him/her?**

1.2 Problem

Based on the user's needs and the data provided by Foursquare API, we should be able to locate neighborhoods in NYC that satisfy those criterias.

1.3 Interest

The answer to this question may be of interest not only to those who seek a place for themselves or their families to live in, but also to public offices working with the establishment of immigrants or refugees, as well as to private businesses such as a real estate offices, or companies seeking professionals abroad, for example. Any entity that is responsible for advising someone to obtain a residence in NYC may be interested in this project.

2 Data acquisition and cleaning

2.1 Required Data

For the execution of this project, we will need:

- Data on the New York boroughs and its neighborhoods, which have already been made available throughout the course (https://cocl.us/new_york_dataset);
- **Foursquare** venues category list, which can be obtained by simply calling an endpoint (https://api.foursquare.com/v2/venues/categories);
- List of places mapped by **Foursquare** in each NYC neighborhood, obtained through **Foursquare API** calls (https://api.foursquare.com/v2/venues/search);
- List of priorities within **Foursquare** categories, informed by the user and which will be used to generate a score for each neighborhood.

2.2 How data will be used

- The user will inform the categories he or she thinks are important in a neighborhood, indicating their priority:

  3: Very important;

  2: Important;

  1: Not so important

- He or she can choose as many categories as he or she likes, indicating their priorities
- The system will obtain the list of boroughs of the city, and for each borough its neighborhoods;
- For each neighborhood, we will get the list of places within the categories that the user chose
- After gathering all data necessary, the system will group the data by neighborhood, summing up the amount of venues of each category, normalizing the data and applying the priority informed by the user to calculate the final rating for each neighborhood
- With this, it will be possible to create clusters using **KMeans clustering** and create groups of neighborhoods based on its calculated ratings

- Two maps will be plotted:
    1. Clusters based on the similarities of the neighborhoods
    2. Groups based on the final ratings of the neighborhoods

2.3 Important notice about Foursquare venue categories

- Foursquare advices that the list of categories may slightly change over time, so it can't be a fixed list and we need to call the API every time we run the application in order to bring all the categories up to date;
- Each venue in Foursquare has at least a main category and a subcategory. For example:
    - "Travel & Transport" = main category
        - "Metro Station" = subcategory
- However, there are some subcategories that are subdivided into narrower categories:
    - "Food" = main category
        - "Italian Restaurant" = subcategory
            - "Calabria Restaurant" = subcategory within "Italian Restaurant"
            - "Venetto Restaurant" = subcategory within "Italian Restaurant"
            - "Puglia Restaurant" = subcategory within "Italian Restaurant"
            - etc.;
- In an extreme case, categories can reach up to four levels:
    - "Outdoors & Recreation" = main category
        - "Athletics & Sports" = subcategory
            - "Gym / Fitness Center" = subcategory within "Athletics & Sports"
                - "Boxing Gym" = subcatgegory within "Gym / Fitness Center"
                - "Climbing Gym" = subcatgegory within "Gym / Fitness Center"
                - "Cycle Studio" = subcatgegory within "Gym / Fitness Center"
                - "Gym Pool" = subcatgegory within "Gym / Fitness Center"
                - etc.;
- It's not a farfetched idea that Foursquare can come up with a five level category or even a higher level category
- When we use Foursquare to search for venues from a certain location, following some category criteria, it might bring categories that were not specified, but having some relation to one of the selected. For instance, if you ask Foursquare to bring all "Italian Restaurants", it will bring all of them and includes all "Pizza Places" as well, but "Pizza Place" subcategory does not belongs to "Italian Restaurant" category. It's an independent main category, but somehow it relates to "Italian Restaurant" (**IT'S NOT CLEAR ON FOURSQUARE ENDPOINT RESPONSE HOW SOME CATEGORIES ARE LINKED**);
- In this project, we are considering that user will be able to select only subcategories (level = 2) from categories (level = 1), but not subcategories (level = 3) from subcategories (level = 2). However, the system will handle the entire category chain brought by Foursquare by assigning each subcategory (second, third and fourth levels) to its second level subcategory.

2.4 Data processing and feature selection

The basic data we acquired came from a JSON containing some information about NYC. However, we only imported into our dataframe information the information described in the Table 1 below.

Table 1 - Basic Information

Column name	Description	Type
Borough	Borough's name	String
Neighborhood	Neighborhood's name	String
Latitude	Geodesic latitude of the neighborhood	Float
Longitude	Geodesic longitude of the neighborhood	Float

The second set of data came from the user. First, he/she has defined a list of venue categories that is considered relevant when choosing a place to live in. An example of the item list used by the application is as follows.

['52f2ab2ebcbc57f1066b8b1c', 'Fruit & Vegetable Store', 3]
---------- First ---------- -------- Second ---------- - Third -

Each item of the list has three attributes (Table 2).

Table 2 - Selected categories and priorities

Position	Description	Type
First	Foursquare category ID	String
Second	Foursquare category name	String
Third	Importance of that category	Int

Then, he/she has selected boroughs in NYC that he/she thinks may have sufficient venues types. Currently, NYC has the following boroughs:

- Bronx
- Manhattan
- Brooklyn
- Queens
- Staten Island

Any combination of boroughs is allowed.

The third step consists in calling a Foursquare API and gather all the venues within the selected categories and boroughs. The response of that call includes several information that is not relevant to the project, so we only keep the following:

Table 3 – Information from Foursquare

Column name	Description	Type
Venue	Venue's name	String
Venue Latitude	Venue's geodesic latitude	Float
Venue Longitude	Venue's geodesic longitude	Float
Category Id	Venue's category code	String
Category Name	Venue's category name	String

Though we are not going to use venue's geodesic coordinates in this project, the tool has already gathered enough information to plot a map in case user wants to see where venues are located.

With the information provided by Foursquare we create another dataframe adding information about the borough and neighborhood.

Besides, venue's category may be one of third, forth or higher level. So we map venue's categories to the ones user has selected, which as second level categories. After that, we will have a new column containing the second level category of that venue, or *None* in case the category does not belong to the ones user selected.

Unfortunately, Foursquare sometimes return more categories than the ones we tried to filter, as we have explained earlier. We need to drop those lines to not mess with our purpose.

Then we add a new column for the second level category name. The final dataframe looks like that:

(Figure 1 – Final dataframe with all the information project needs)

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	Category Id	Subcategory	Subcategory Name
0	Marble Hill	40.876551	-73.91066	MTA Subway - 225th St/Marble Hill (1)	40.874486	-73.909589	Metro Station	4bf58dd8d48988d1fd931735	4bf58dd8d48988d1fd931735	Metro Station
1	Marble Hill	40.876551	-73.91066	Arturo's	40.874412	-73.910271	Pizza Place	4bf58dd8d48988d1ca941735	4bf58dd8d48988d1ca941735	Pizza Place
2	Marble Hill	40.876551	-73.91066	Bronx Boxing	40.875671	-73.908355	Boxing Gym	52f2ab2ebcbc57f1066b8b47	4f4528bc4b90abdf24c9de85	Athletics & Sports
3	Marble Hill	40.876551	-73.91066	marble hill pharmacy	40.875050	-73.909195	Pharmacy	4bf58dd8d48988d10f951735	4bf58dd8d48988d10f951735	Pharmacy
4	Marble Hill	40.876551	-73.91066	24 Hour Fitness	40.880592	-73.908255	Gym / Fitness Center	4bf58dd8d48988d175941735	4f4528bc4b90abdf24c9de85	Athletics & Sports

3 Exploring the data

Now, we start exploring the data.

First, we transform our dataframe into another one with selected categories as columns and venueses as index, indicating for each venue what category it is:

0 : Does not belong that category
1 : Belongs to that category

The result is this:

(Figure 2 – Type of category per venue)

	Neighborhood	Athletics & Sports	Bus Stop	Drugstore	Fruit & Vegetable Store	Italian Restaurant	Laundry Service	Market	Metro Station	Pharmacy	Pizza Place
0	Marble Hill	0	0	0	0	0	0	0	1	0	0
1	Marble Hill	0	0	0	0	0	0	0	0	0	1
2	Marble Hill	1	0	0	0	0	0	0	0	0	0
3	Marble Hill	0	0	0	0	0	0	0	0	1	0
4	Marble Hill	1	0	0	0	0	0	0	0	0	0

Notice that in this dataframe we supress venue's name as we are not going to use this information down the processing flow.

Then we group the dataframe by neighborhood in order to find the number of each category places.

(Figure 3 – Number of venueses per category)

	Neighborhood	Athletics & Sports	Bus Stop	Drugstore	Fruit & Vegetable Store	Italian Restaurant	Laundry Service	Market	Metro Station	Pharmacy	Pizza Place
0	Battery Park City	14	0	0	0	4	0	1	18	7	2
1	Carnegie Hill	17	0	0	0	4	0	0	6	14	7
2	Central Harlem	15	3	0	0	3	4	1	8	6	7
3	Chelsea	20	0	0	0	5	0	1	13	4	4
4	Chinatown	12	0	0	0	9	0	0	12	8	5

As we gonna apply the weights for each category, we need to normalize the data to avoid distortions on the results. This can be done by dividing each cell by the maximum value found on it. At the same time, we apply the weight specified by the user for each category. We just apply the formula:

*cell value = (cell value * weight of its category) / maximum value of the cell column*

The data in each cell will be between 0 and 3 after this transformation, as the higher priority is 3.

We create another column with the result of the sum of the ratings for each line (neighborhood), 'Total Rating', that will be used for further analysis. Values for this new column range from 0 to (3 x number of categories).

A category with a great number of venues but with a lower priority may be less important as a category with lower number of venues but with a higer priority. The number of venues is important, but we need to take it associated with the priority of its category.

Important to notice that changing priorities, boroughs or categories may lead to a change in the clusters and groups, reflecting the individual needs.

The result dataframe looks like that:

(Figure 4 – Rating of each neighborhood x category)

	Neighborhood	Athletics & Sports	Bus Stop	Drugstore	Fruit & Vegetable Store	Italian Restaurant	Laundry Service	Market	Metro Station	Pharmacy	Pizza Place	Total Rating
0	Battery Park City	1.12	0.0	0.0	0.0	0.727273	0.0	1.5	2.70	0.500000	0.307692	6.854965
1	Carnegie Hill	1.36	0.0	0.0	0.0	0.727273	0.0	0.0	0.90	1.000000	1.076923	5.064196
2	Central Harlem	1.20	1.5	0.0	0.0	0.545455	0.8	1.5	1.20	0.428571	1.076923	8.250949
3	Chelsea	1.60	0.0	0.0	0.0	0.909091	0.0	1.5	1.95	0.285714	0.615385	6.860190
4	Chinatown	0.96	0.0	0.0	0.0	1.636364	0.0	0.0	1.80	0.571429	0.769231	5.737023

As we can see in this example, the number of stores of "Fruit & Vegetable", "Drugstore" and "Market" are not that many. This may suggest that the borough chosen by the user lacks some amenities. However, the borough seems to be plenty of gymns, metro stations and italian/pizza restaurants.

3.1 Clustering

In order to analyse Kmeans clustering we transform our dataframe into another one. In this new dataframe we have just sorted categories by the number of ocurrences of venues for each neighborhood, organizing the columns to show the most common venue first to the less common venue last. The algorithm is expected to group neighborhoods based on this dataframe, that ignores the total number of venues in each neighborhood. The new dataframe looks like this one.

(Figure 5 – Dataframe for Kmeans analysis)

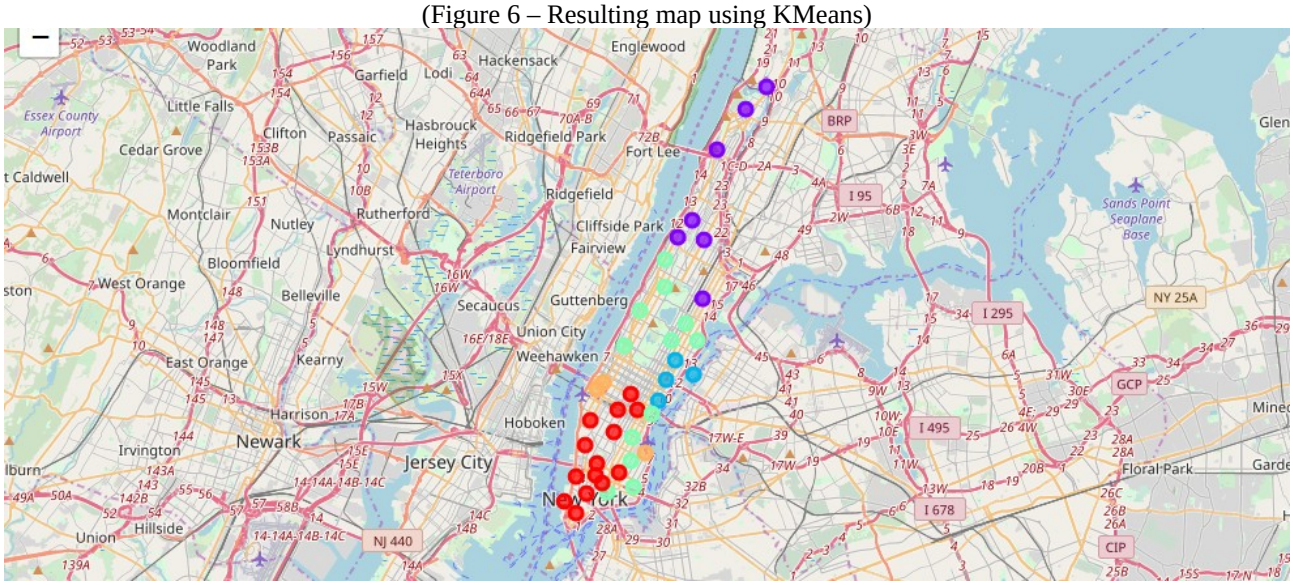
	Neighborhood	Total Rating	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Battery Park City	6.854965	Metro Station	Market	Athletics & Sports	Italian Restaurant	Pharmacy	Pizza Place	Laundry Service	Fruit & Vegetable Store	Drugstore	Bus Stop
1	Carnegie Hill	5.064196	Athletics & Sports	Pizza Place	Pharmacy	Metro Station	Italian Restaurant	Market	Laundry Service	Fruit & Vegetable Store	Drugstore	Bus Stop
2	Central Harlem	8.250949	Market	Bus Stop	Metro Station	Athletics & Sports	Pizza Place	Laundry Service	Italian Restaurant	Pharmacy	Fruit & Vegetable Store	Drugstore
3	Chelsea	6.860190	Metro Station	Athletics & Sports	Market	Italian Restaurant	Pizza Place	Pharmacy	Laundry Service	Fruit & Vegetable Store	Drugstore	Bus Stop
4	Chinatown	5.737023	Metro Station	Italian Restaurant	Athletics & Sports	Pizza Place	Pharmacy	Market	Laundry Service	Fruit & Vegetable Store	Drugstore	Bus Stop

Notice that the number of venues in each neighborhood and priorities for each category are not used for this analysis.

Then we fit our Kmeans model with this dataframe and the cluster labels are inserted into the dataframe before we plot the map.

In the map below we can see the clusters in different colors.

(Figure 6 – Resulting map using KMeans)



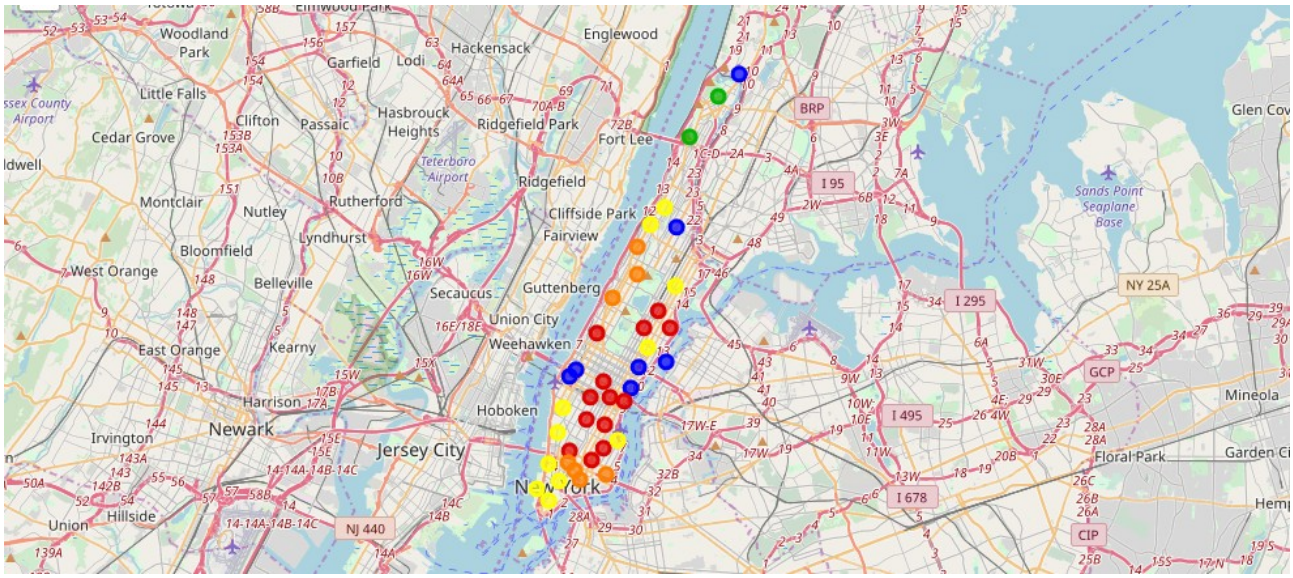
3.2 Grouping by ratings

Ratings have a more objective approach. The rational behind this technique is quite simple. We have calculated ratings for every line (neighborhood) based on the users choices. This method is supposed to bring a more practical result. Now the grounds for a decision are made based on the number of the venues and priorities of their categories, which tends to lead a closer result from the user's expectations.

Though this technique is much simpler than Kmeans, it seems to be more suitable for what user needs, since its more direct and objective.

We have created five intervals between the mininum and maximum values of the ratings in order to for the groups. The resulting map is shown below.

(Figure 7 – Resulting map using ratings)



4 Results and Discussion

One point important to mention is that this project didn’t involve any predictive technique, which makes it easier to accomplish. We didn’t have to create, test different parameters and approaches, or refine statistical models.

Depending on the requirements the user demands, any neighborhood can be of any interest. The tool we just developed is capable of bring some neighborhoods that fit the user expectations for a new home in New York City. However, some observations are due.

The tools is highly dependable on the quality of data provided by Foursquare:

- If a venue is not registered on Foursquare, it will not show up in our search;
- Some venues may have been shut down by the time we run the application, for instance;
- The tool only assess the quantitative aspect of the categories, not the qualitative one.

Each metropolitan area may have its own subdivision mechanism, so further programming needs to be made to adapt the tool for another area.

It would be very interesting to bring other types of data such as criminal statistics, demographics, cost of square meter for renting/acquisition, etc. These data will certainly add more value to the tool.

4.1 Differences between the two algorithms used

We have deployed two different algorithms:

- **KMeans**
- **Grouping based on the overall rating of the neighborhood**

The idea was to compare both mechanisms. While the former is based only on information about the most common categories in a neighborhood, without direct interference from user-given priorities, the latter is based on the most important aspects of calculating an indicator for each location, grouping neighborhoods according to its grades.

The KMeans algorithm seems not to be appropriate for the purpose of this project, as it ignores important premises given by the user.

5 Conclusions

The purpose of this project was to develop a tool to help people moving to New York City to find a neighborhood that best attends their specific needs in terms of venues and amenities.

With some additional programming it could be extended to other metropolitan areas.

It's is flexible enough to allow users to filter results by a huge number of combinations of categories and boroughs, and grouping locations according to its ratings, so user can have an idea of which neighborhoods are closer or distant from your needs.

Although the system recommends some locations, the user needs to check each of the indicated places and consider other factors not associated with the tool search such as noise levels, presence of illegal activities, property prices, proximity to work, etc.