2019

# Neighborhood scouting for real estate

Elie MAZE

July 25, 2019

# Table of Contents

# Data

Humans cannot explore manually a dozen variables for hundreds of neighborhoods in order to find the best one.

The approach here is to transcribe the customer's criteria as features for all neighborhoods and to clusterize the neighborhoods.

The second step consists in comparing the clusters and identify the group of neighborhoods that best fits the needs of the customer then we should have a smaller amount of neighborhoods to analyze.

## Sources and collecting data

As we do not have access to ready-to-use data, we need to grap data from different sources.

As some of the criteria are related to the proximity or the number of some categories of venues, we will use the Foursquare API to identify venues within a given radius from the centroid of each neighborhood.

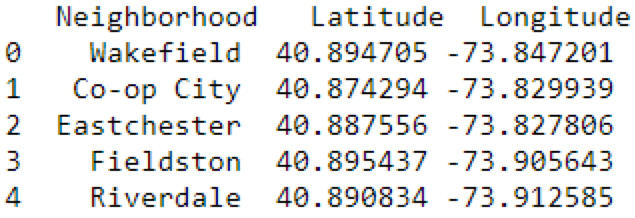
So first of all we need to collect all the neighborhoods of the 4 cities and the geographical locations.

Then we will collect the crimes that were reported in each city in 2018. As the crimes are not always grouped by neighborhood, we will count the number of crimes occurring within a given radius from the centroid of each neighborhood (very time consuming). And fortunately the latitude/longitude are available for each dataset.

### New-York City

The list of NYC neighborhoods and theirs locations is stored in a JSON file (newyork_data.json) coming from the last notebook : Coursera – Applied data-science capstone, week #3: Lab - Segmenting and Clustering Neighborhoods in New York City. The file contains a list of the neighborhoods plus the latitude and longitude.

Dataframe preview:



```
  Neighborhood   Latitude  Longitude
0    Wakefield  40.894705 -73.847201
1   Co-op City  40.874294 -73.829939
2  Eastchester  40.887556 -73.827806
3    Fieldston  40.895437 -73.905643
4    Riverdale  40.890834 -73.912585
```

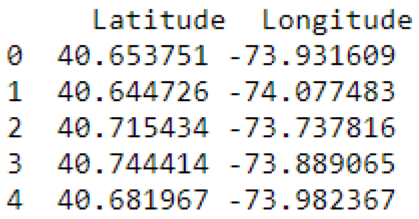
*Figure 1: NYC dataframe containing neighborhoods and their location.*

The crimes were found on the Kaggle web site:

https://www.kaggle.com/mihalw28/nyc-crimes-2018-data-cleaning-part-i

Data were downloaded and saved as ny_crimes.csv.

Dataframe preview:



```
    Latitude  Longitude
0  40.653751 -73.931609
1  40.644726 -74.077483
2  40.715434 -73.737816
3  40.744414 -73.889065
4  40.681967 -73.982367
```

*Figure 2: NYC dataframe containing all the crimes reported  in 2018 (only their positions).*

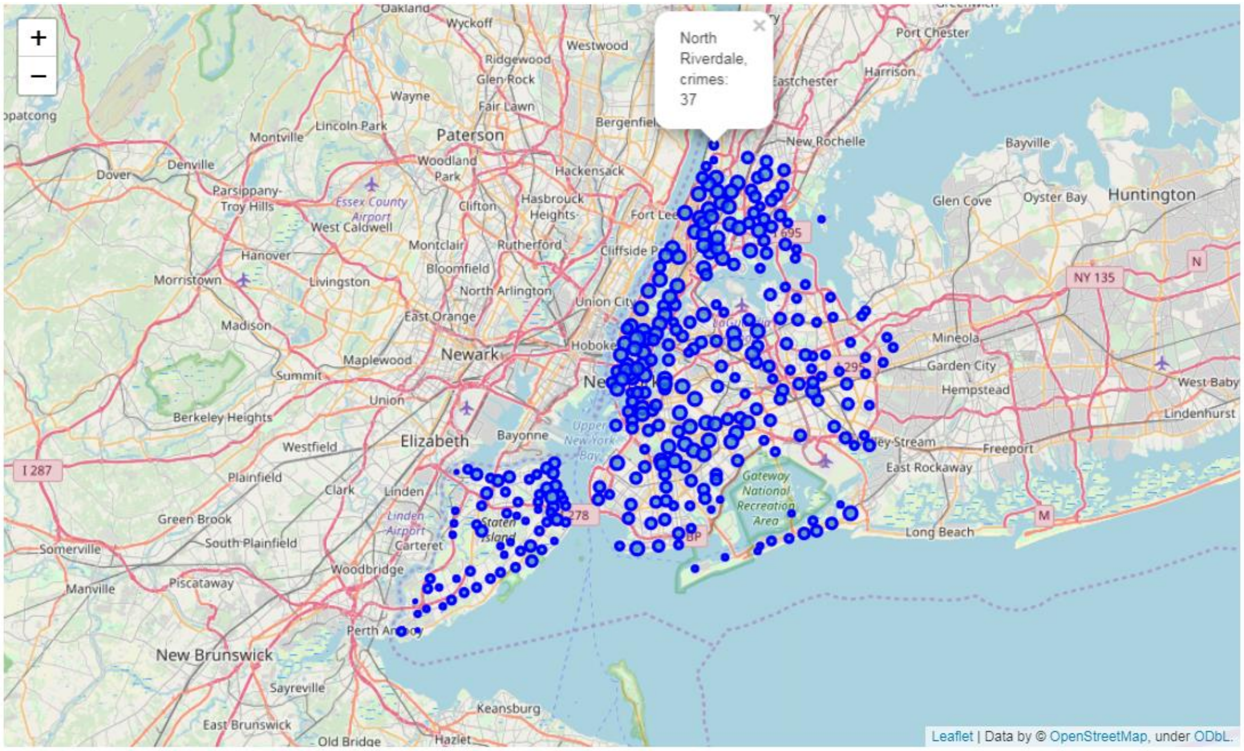


*Figure 3: Location of NYC neighborhoods, the radius of the markers is related to the number of crimes.*

## Toronto

We do the same workflow as the one described in Coursera – Applied data-science capstone, week #3: Segmenting and Clustering Neighborhoods in the city of Toronto, Canada. The list of neighborhoods and zip codes is available in Wikipedia:

https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M

We use then the module geopy to fetch the latitude and longitude of each neighborhood given its address that includes the zip code.

The crimes data are downloaded as toronto_crimes_2014_to_2018.csv from the web site:

https://data.torontopolice.on.ca/datasets/98f7dde610b54b9081dfca80be453ac9_0/data?geometry=-334.512%2C-52.268%2C334.512%2C52.268

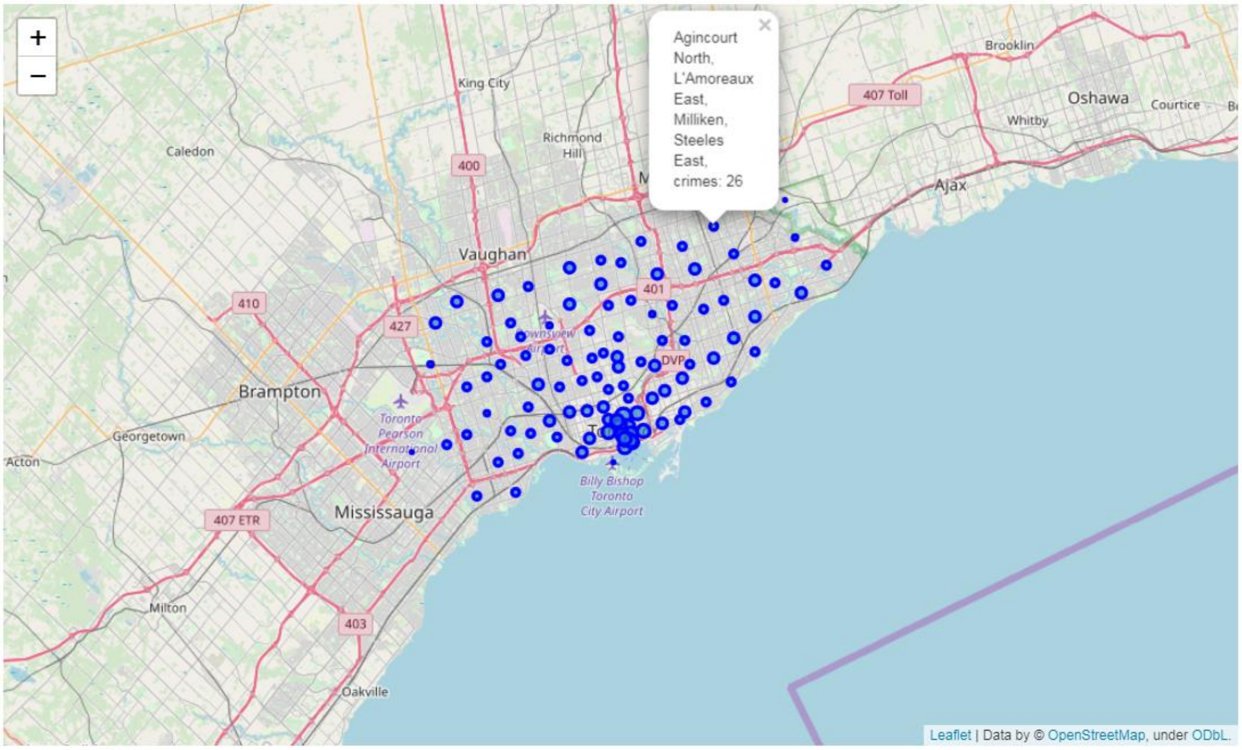

+
−
Agincourt North, L'Amoreaux East, Milliken, Steeles East, crimes: 26
King City
Richmond Hill
Caledon
Brooklin
407 Toll
Oshawa
Courtice
Whitby
Ajax
400
Vaughan
401
410
427
DVP
Brampton
Toronto Pearson International Airport
Georgetown
Acton
Billy Bishop Toronto City Airport
407 ETR
Mississauga
Milton
403
Oakville
Leaflet | Data by © OpenStreetMap, under ODbL.

*Figure 4:Location of Toronto neighborhoods. The radius of the markers is related to the number of crimes.*

## Chicago

The neighborhoods plus the locations were found in Wikipedia and saved as chicago_neighborhoods.csv. It's may refer more to the community areas than to the neighborhoods but it is sufficient for NAPE.

Crimes were saved as Chicago_Crimes_2018.csv and were downloading from Kaggle:

https://www.kaggle.com/spirospolitis/chicago-crimes-20012018-november

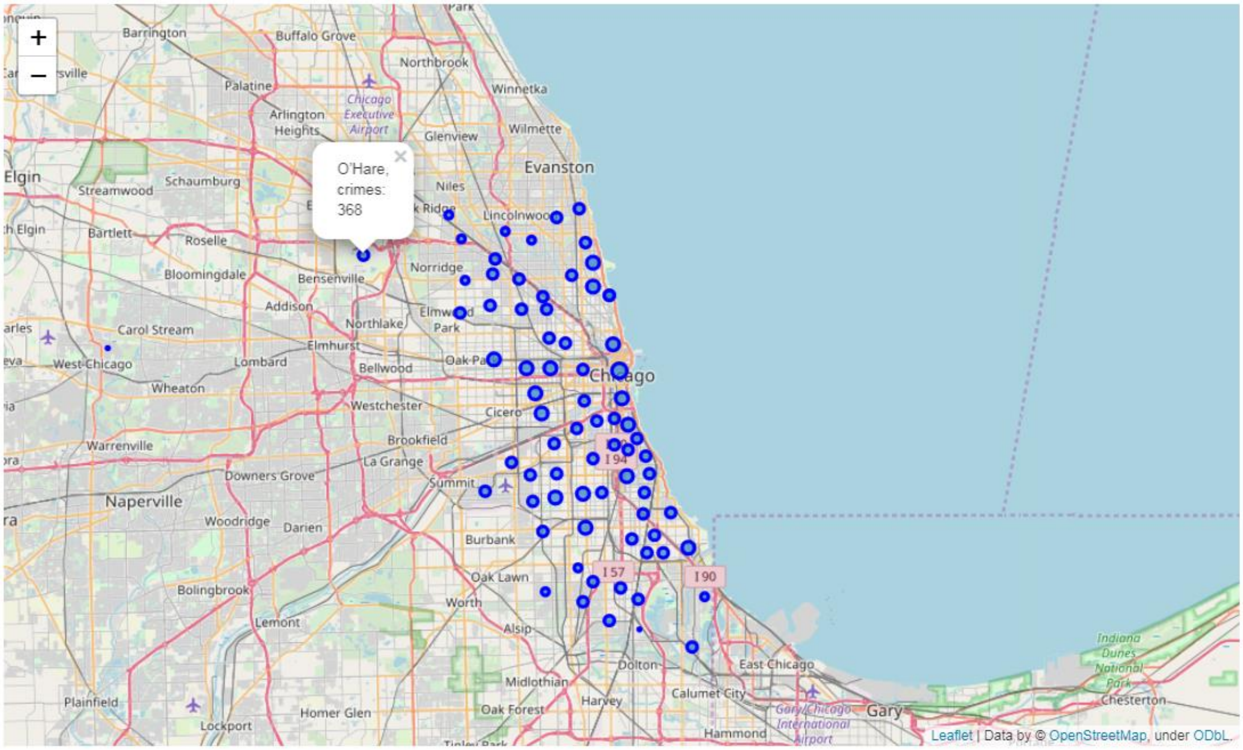


*Figure 5: Location of Chicago neighborhoods. The radius of the markers is related to the number of crimes.*

## San-Francisco

Neighborhoods, latitudes and longitudes were found in Wikipedia and saved as sf_neighborhood.csv.

The crimes data were fetched also on Kaggle and saved as SF_crimes.csv:

https://www.kaggle.com/psmavi104/san-francisco-crime-data

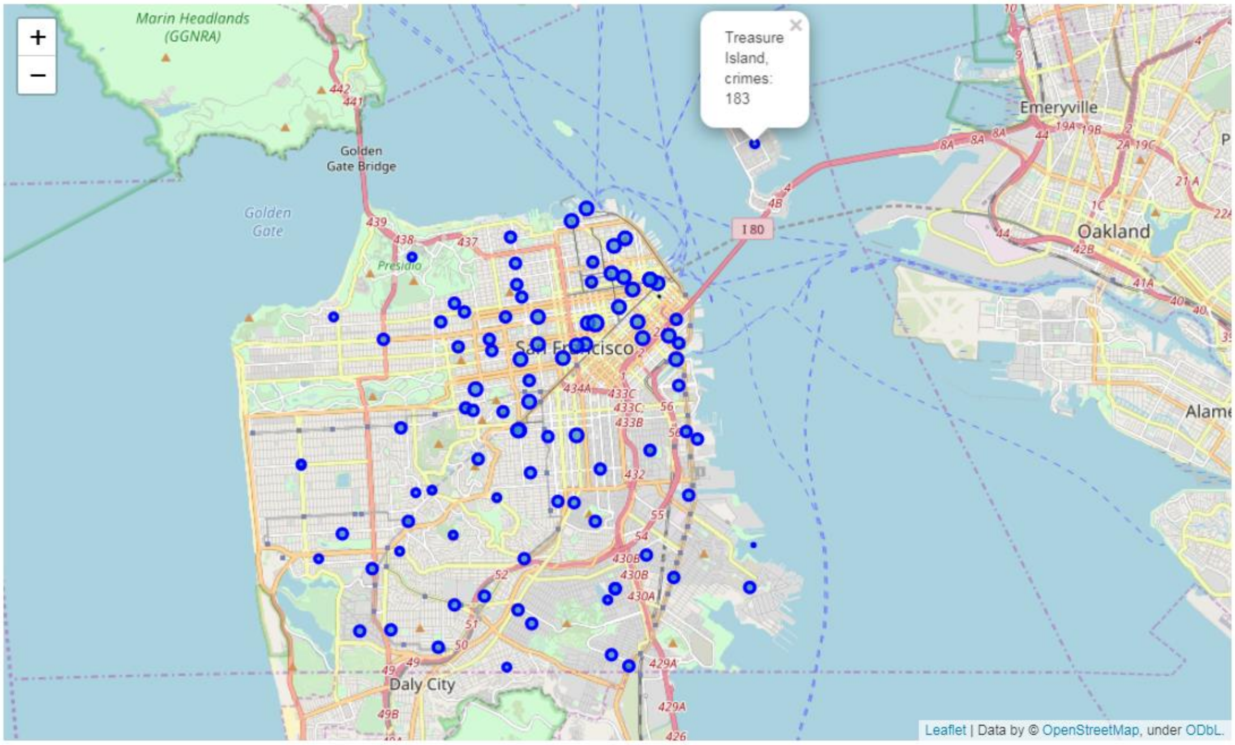


*Figure 6: Location of San-Francisco neighborhoods. The radius of the markers is related to the number of crimes.*

## Data preparation

1. We will stored the features that are related to the customer's criteria as columns and the neighborhoods as rows. The objective is to generate a single dataframe containing:
   - The name of the neighborhood,
   - The number of crimes which occurred at a maximum distance of 500 meters from the neighborhood centroid,
   - The number of distinct type of restaurants,
   - The number of distinct type of entertainment venues for the night,
   - The average distance between the neighborhood location and the locations of the venues related to the category "Transport",
   - The average distance between the neighborhood location and the locations of the services in the city,
   - The average distance between the neighborhood location and the locations of the medical venues in the city,
   - The average distance between the neighborhood location and the locations of the cultural venues in the city,
   - The average distance between the neighborhood location and the locations of the stadiums and sport complexes in the city,
   - The average distance between the neighborhood location and the locations of the green areas (for running) or the swimming pools in the city.

2. Given these features, we will clusterize the neighborhoods using the K-means algorithm. Then will compare the average characteristics of each cluster to determine the best group.

3. After that, we will visualize the characteristics of each neighborhood of the best cluster to identify which neighborhood represents the best compromise to meet the needs of the customer.

## Collecting and cleaning

After collecting the neighborhoods and locations, missing values are dropped. Missing values are:

- neighborhood names containing None, NaN or the mention "unassigned",
- no match found for the neighborhood address using geopy so the latitude and longitude are NaN.

The dataframes containing the crimes data have also missing values. Sometimes the date of occurrence is missing or inconsistent (example : 03-18-1019 instead of 03-18-2019 ?). Rows with such anomalies are removed from the dataframe.

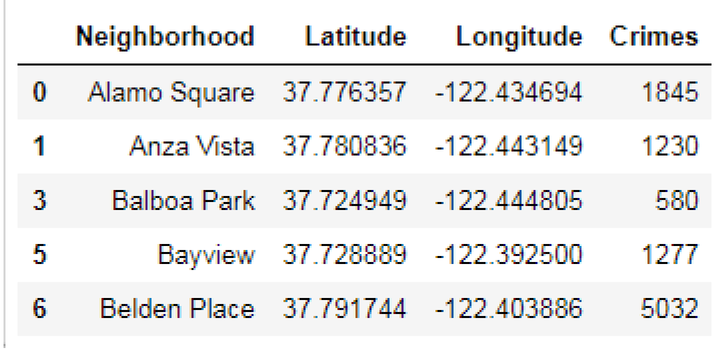
Finally the crimes data are filtered, we keep only crimes that occurred in 2018.

## Data processing

Neighborhoods and their latitudes and longitudes as stored in dataframe. There is one dataframe per city so we have 4 dataframes.

### *Crimes*

For each neighborhood, we compute the distance between the location of the neighborhood and the locations of all the crimes that occurred in the city in 2018. Then we filter the list to keep only distances less than or equal to 500 meters. Eventually we count the number of crimes. Finally, we get a single dataframe per city that contains the name of the neighborhood, its latitude and longitude and the number of crimes that occurred at a maximum distance of 500 meters from its position (centroid).



| | Neighborhood | Latitude | Longitude | Crimes |
|---|---|---|---|---|
| 0 | Alamo Square | 37.776357 | -122.434694 | 1845 |
| 1 | Anza Vista | 37.780836 | -122.443149 | 1230 |
| 3 | Balboa Park | 37.724949 | -122.444805 | 580 |
| 5 | Bayview | 37.728889 | -122.392500 | 1277 |
| 6 | Belden Place | 37.791744 | -122.403886 | 5032 |

*Figure 7: Preview of the San-Francisco dataframe containing the neighborhood names, locations and the number of crimes.*

### *Venues*

We use the Foursquare API to get the top 100 common venues within a radius of 500 meters from the neighborhoods locations. We obtain a dataframe with the name and the category of the venues found for each neighborhood.

As criteria are mainly based on some specific categories of venues, we need to explore all the categories and try to build a mapping in order to gather the categories. We extract the venue categories from the dataframe of each city and remove the duplicates. Then we manually analyze them to normalize the categories we are interested in.

We found more than 500 unique venue categories that we will group into 8 final categories. For example Indian restaurant, French restaurant, Sandwich Place will correspond to the category "Food". The final categories are:

| Final category | Description |
|---|---|
| **Medical Care** | All venue categories related to health (medical center, hospital, drugstore, doctor's office…). |
| **Transport** | All the transport venues: train station, bus, metro… We only take into account the intra-city transport, so boat/ferry and planes are ignored. |
| **Food** | The venues were the customer can eat (various kinds of restaurants, BBQ places, Hot dog/Hamburger places, etc.). |
| **Nature-Sport** | This category gather all the venues related to green areas (outdoors, parks…) or the swimming pools. |
| **Culture** | The cultural venues, it could be monuments, museums, comedy clubs, Opera… |
| **Entertainment** | The venues where the customer will attend sport games (stadiums, …). |
| **Services** | This category gather the most common services like the post office, bank or insurance. |
| **Other** | All other venue categories. |

Once the new column (final_category) is created, we will:

1. Group the records of the dataframe by neighborhood and:
   - Count the distinct venue categories with final category "Food",
   - Count the distinct venue categories having the final category "Night-Entertainment".

2. Compute the average distance between each neighborhood location and the venues locations belonging to a specific final category to compute the following features:
   - Transport_distance,
   - Service_distance,
   - Stadium_distance,
   - Medical_distance,
   - Culture_distance,
   - Nature-Sport_dist.

### *End of processing*

So we have 4 dataframes with the following columns:

| Feature name | Description |
|---|---|
| **Neighborhood** | Name of the neighborhood. |
| **Crimes** | Number of crimes occurring within a radius of 500m from the neighborhood centroid. |
| **Food** | The number of distinct kind of places to eat. |
| **Night-Entertainment** | The number of distinct kind of places to have fun during the night |
| **Transport_distance** | The average distance between the neighborhood centroid and all the venues related to the intra-city transport. |
| **Service_distance** | The average distance between the neighborhood centroid and all the services in the city. |
| **Stadium_distance** | The average distance between the neighborhood centroid and all the services in the city. |
| **Medical_distance** | The average distance between the neighborhood centroid and all the services in the city. |
| **Culture_distance** | The average distance between the neighborhood centroid and all the services in the city. |
| **Nature-Sport_dist** | The average distance between the neighborhood centroid and all the services in the city. |

The 4 dataframes are merged into a single one.

As we want to compare all criteria independently and give them the same importance, we scaled all the features between 0.0 and 1.0 so that they are in the same range.

After we apply the equation 1-x to:

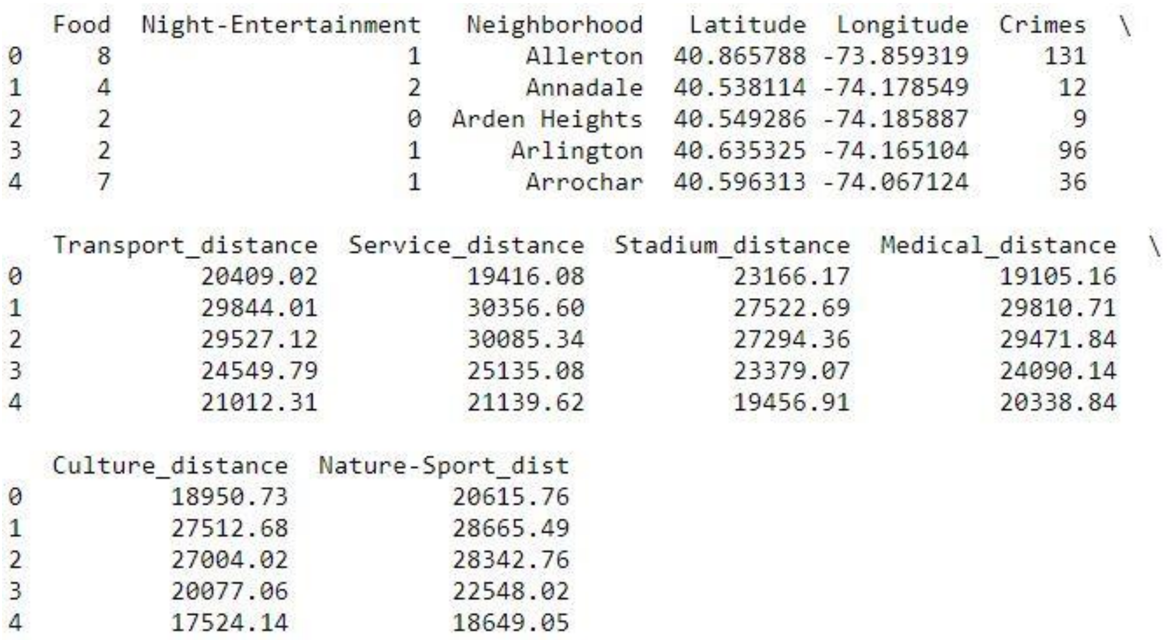
- the feature Crimes,
- all the features related to average distances.

The consequence is:

- when crimes feature is high, the number of crimes is in reality low,
- when the average distance features are high, it means than the average distance is in reality low so that in this case the venues are closed to the neighborhood centroid.

In this situation, it is easier to analyze and compare data because all we need will be to find the neighborhoods that maximize all the variables.

Now we  have a dataframe containing the neighborhood names and the features, we can start the analysis then the clustering that will be discussed in the next section coming in week #2.



```
   Food  Night-Entertainment   Neighborhood   Latitude  Longitude  Crimes  \
0     8                    1       Allerton  40.865788 -73.859319     131
1     4                    2       Annadale  40.538114 -74.178549      12
2     2                    0  Arden Heights  40.549286 -74.185887       9
3     2                    1      Arlington  40.635325 -74.165104      96
4     7                    1       Arrochar  40.596313 -74.067124      36

   Transport_distance  Service_distance  Stadium_distance  Medical_distance  \
0            20409.02          19416.08          23166.17          19105.16
1            29844.01          30356.60          27522.69          29810.71
2            29527.12          30085.34          27294.36          29471.84
3            24549.79          25135.08          23379.07          24090.14
4            21012.31          21139.62          19456.91          20338.84

   Culture_distance  Nature-Sport_dist
0          18950.73           20615.76
1          27512.68           28665.49
2          27004.02           28342.76
3          20077.06           22548.02
4          17524.14           18649.05
```

*Figure 8: The final dataframe containing the features plus the neighborhood names and the latitudes/longitudes.*