Presentation

Jeronimo needs to open his Burger Joint in some Locality (District) of Bogotá, Colombia. Furthermore, he is asking our assistance in finding the best Locality. Business Problem So as to open Jeronimo's burger joint, we will want to utilize some data about the Localities in Bogotá. We should be certain that where the burger joint is opened will have enough clients and furthermore that there aren't an enormous ammount of burer joints in the spot.

Information

To help Jeronimo in his hunt we should get to following information:

- The Localities of Bogotá, Colombia from Wikipedia: https://es.wikipedia.org/wiki/Anexo:Localidades_de_Bogot%C3%A1

- The directions (scope, longitude) ot these Localities of Bogotá from Open Street Map APIs

- From Foursquare we will require following scenes information:

- the burger joint settings of the Localities

- the workplaces settings of the Localities

- the secondary schools settings of the Localities

- the colleges settings of the Localities

We will at that point influence the information so as to figure out which area is the most proper so as to find the burger joint.

Methodology

- For each locality, all office, school, university and burger joints venues data have been collected from Foursquare.

- Then for each locality, the sums of the office, school, university and burger joints were computed.

- For each of this 4 categories, a weight (or penalty) has been defined according to what Jeronimo considers the most important.
- Burger Joints have been weighted with -1, since Paolo wants to avoid concurrence.

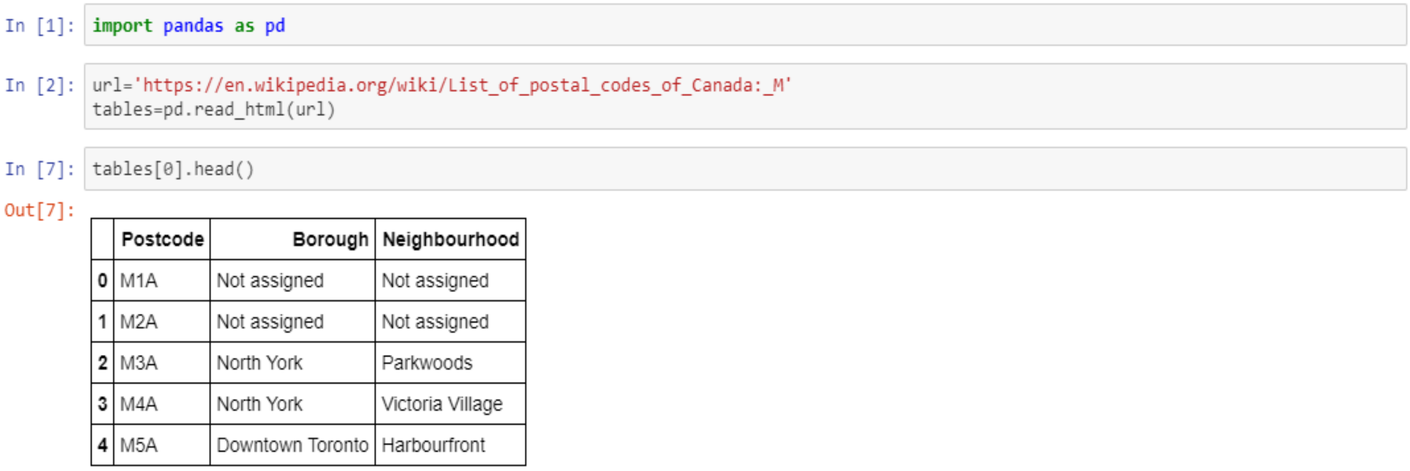
- Schools have been weighted with 1, since student are good customers.

- Universities have been weighted with 1.5, since students are good customers.

- Offices have been weighted with 2, since employees are even better customers.
- Note that the weights can be modified according to the importance of each category.

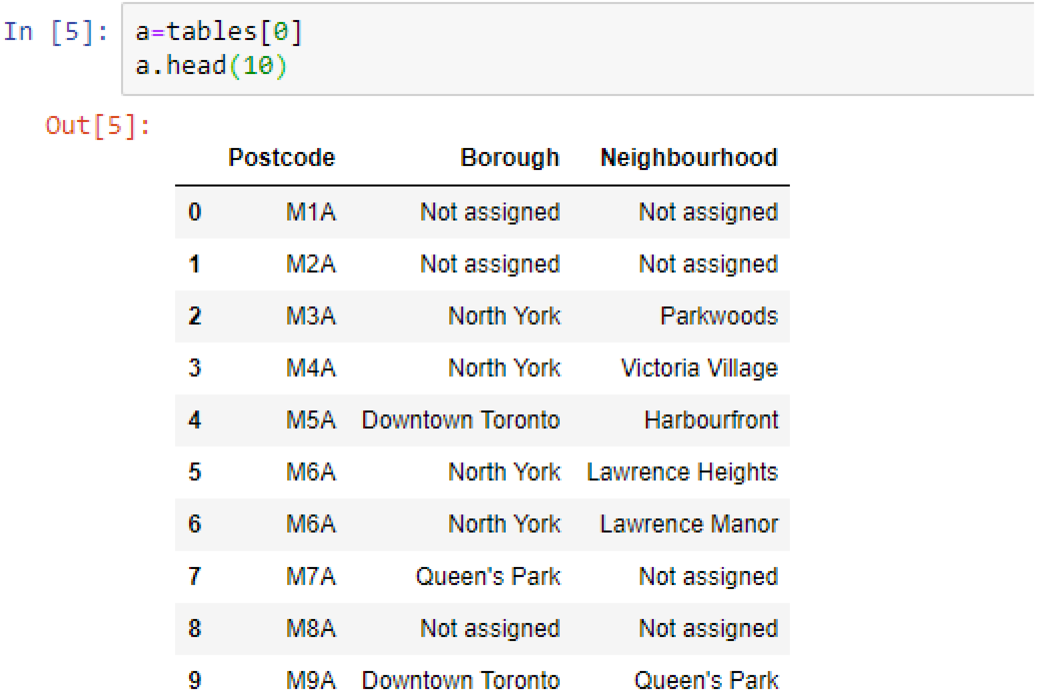
- Lastly, a score was computed for each locality as the weighted sum of the number of venues in each of the 4 categories (school, university, office, burger joints).

Coding



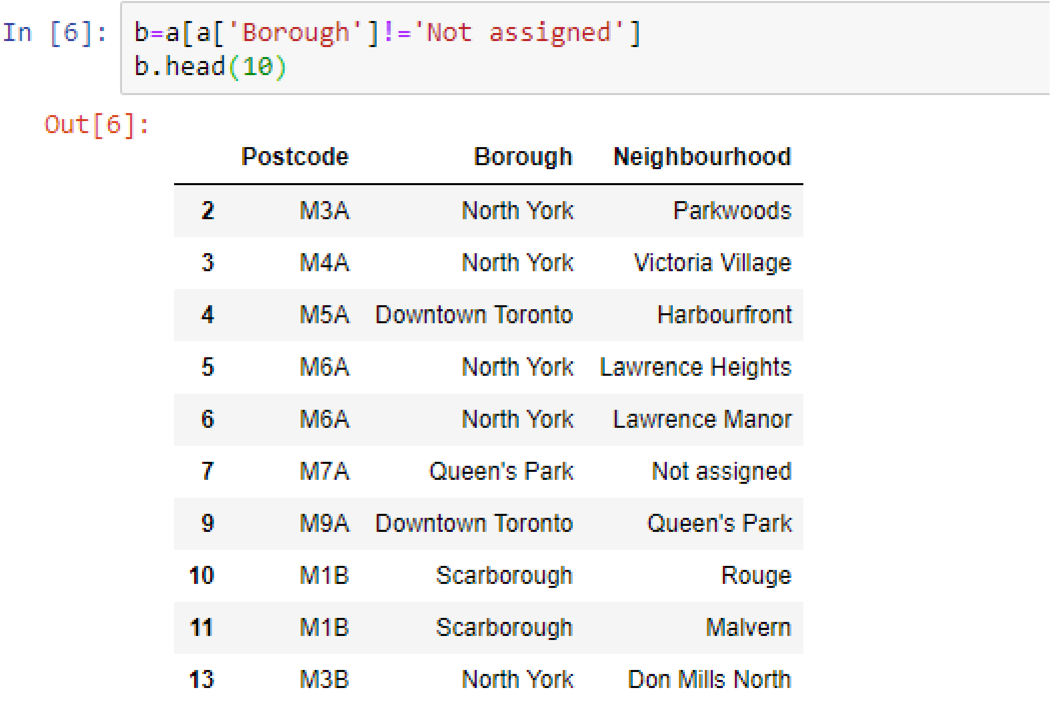
```
In [1]: import pandas as pd

In [2]: url='https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M'
        tables=pd.read_html(url)

In [7]: tables[0].head()

Out[7]:
```

	Postcode	Borough	Neighbourhood
0	M1A	Not assigned	Not assigned
1	M2A	Not assigned	Not assigned
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Harbourfront



```
In [5]: a=tables[0]
        a.head(10)

Out[5]:
```

	Postcode	Borough	Neighbourhood
0	M1A	Not assigned	Not assigned
1	M2A	Not assigned	Not assigned
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Harbourfront
5	M6A	North York	Lawrence Heights
6	M6A	North York	Lawrence Manor
7	M7A	Queen's Park	Not assigned
8	M8A	Not assigned	Not assigned
9	M9A	Downtown Toronto	Queen's Park

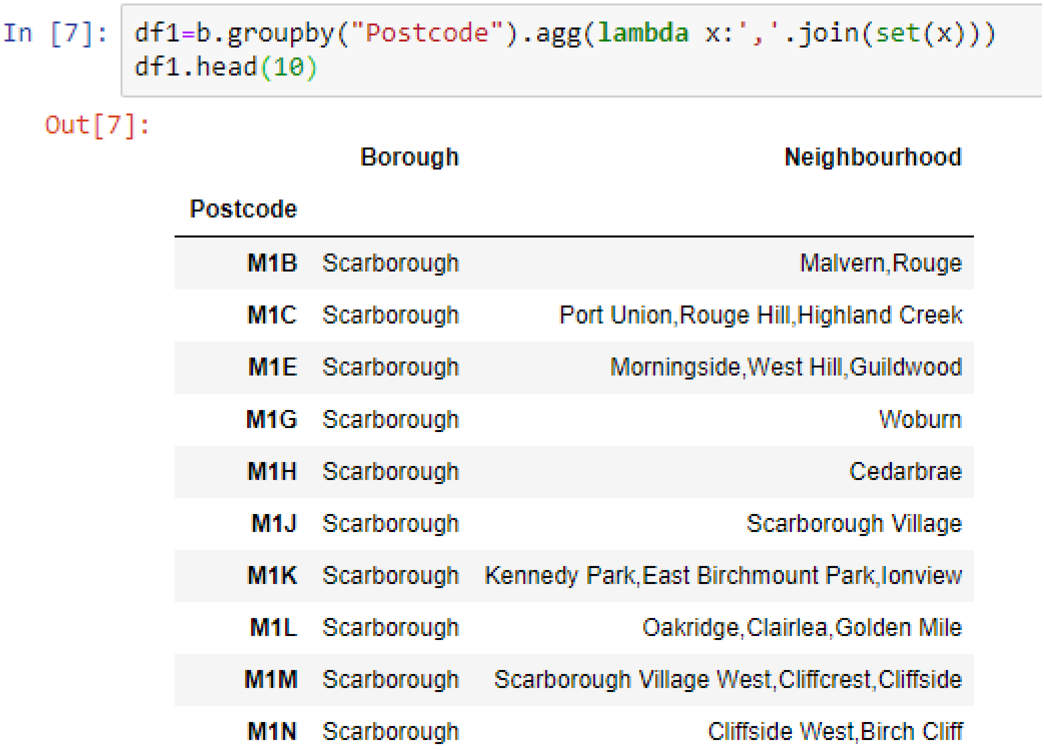


In [6]:

```
b=a[a['Borough']!='Not assigned']
b.head(10)
```

Out[6]:

	Postcode	Borough	Neighbourhood
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Harbourfront
5	M6A	North York	Lawrence Heights
6	M6A	North York	Lawrence Manor
7	M7A	Queen's Park	Not assigned
9	M9A	Downtown Toronto	Queen's Park
10	M1B	Scarborough	Rouge
11	M1B	Scarborough	Malvern
13	M3B	North York	Don Mills North

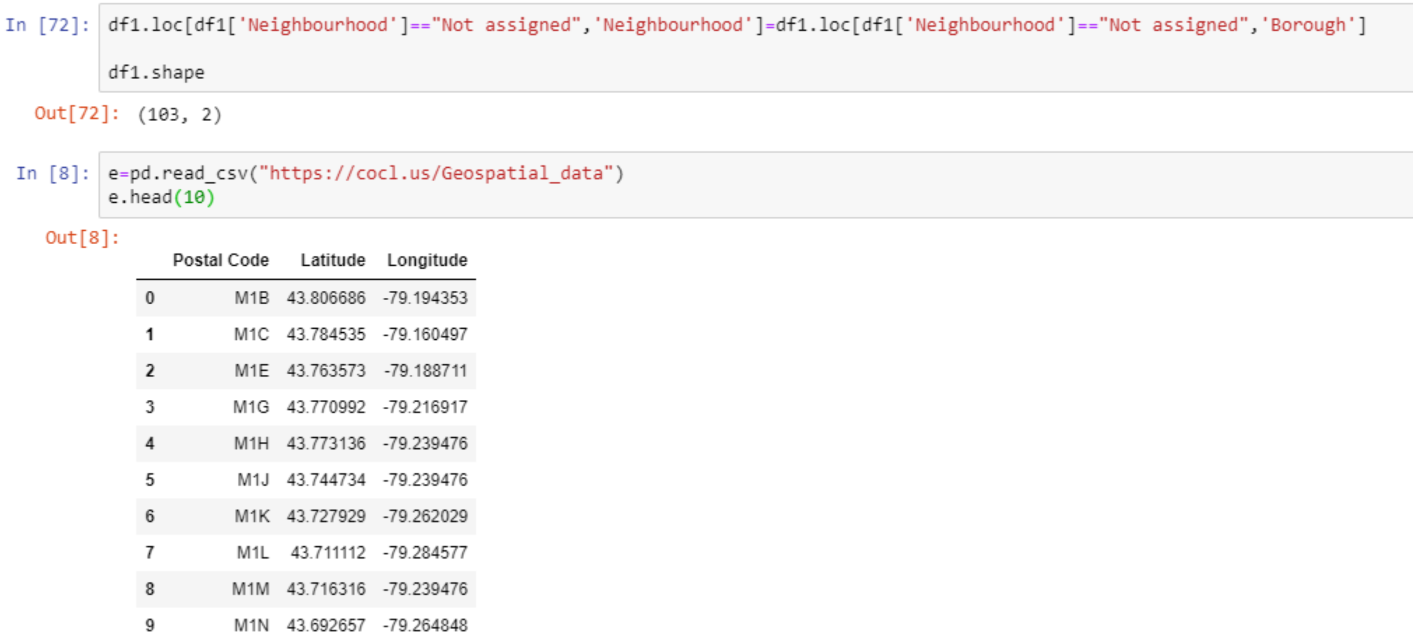


In [7]:

```
df1=b.groupby("Postcode").agg(lambda x:','.join(set(x)))
df1.head(10)
```

Out[7]:

	Borough	Neighbourhood
Postcode		
M1B	Scarborough	Malvern,Rouge
M1C	Scarborough	Port Union,Rouge Hill,Highland Creek
M1E	Scarborough	Morningside,West Hill,Guildwood
M1G	Scarborough	Woburn
M1H	Scarborough	Cedarbrae
M1J	Scarborough	Scarborough Village
M1K	Scarborough	Kennedy Park,East Birchmount Park,Ionview
M1L	Scarborough	Oakridge,Clairlea,Golden Mile
M1M	Scarborough	Scarborough Village West,Cliffcrest,Cliffside
M1N	Scarborough	Cliffside West,Birch Cliff



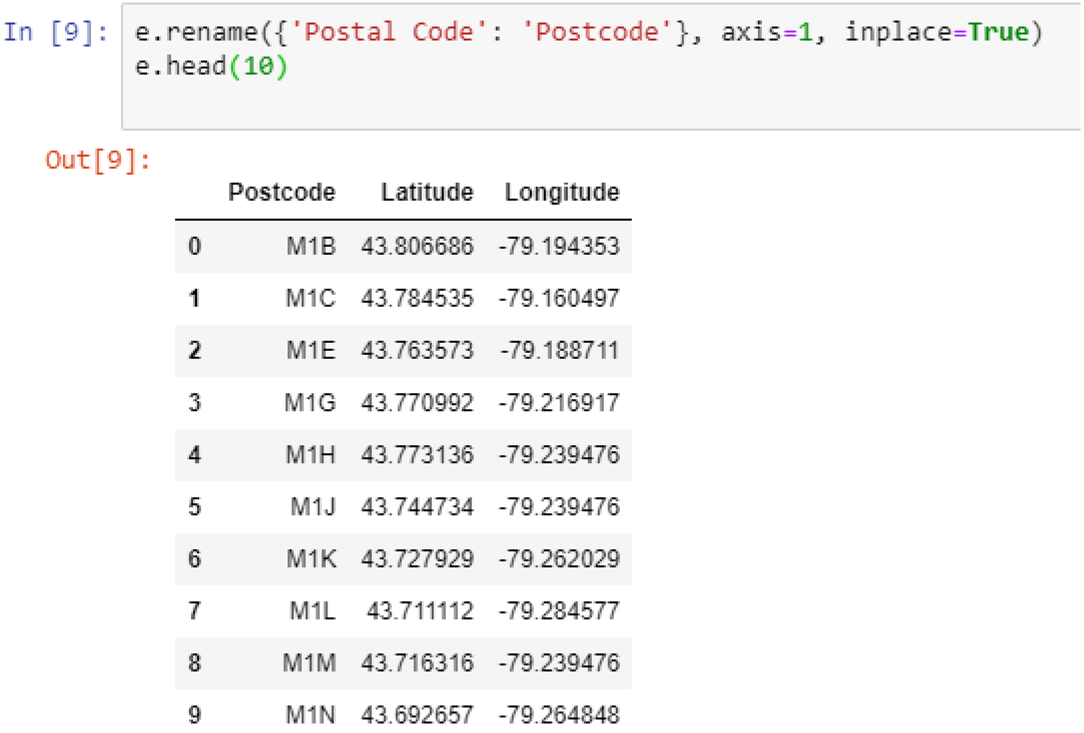
```
In [72]: df1.loc[df1['Neighbourhood']=="Not assigned",'Neighbourhood']=df1.loc[df1['Neighbourhood']=="Not assigned",'Borough']

         df1.shape
```

```
Out[72]: (103, 2)
```

```
In [8]: e=pd.read_csv("https://cocl.us/Geospatial_data")
        e.head(10)
```

Out[8]:

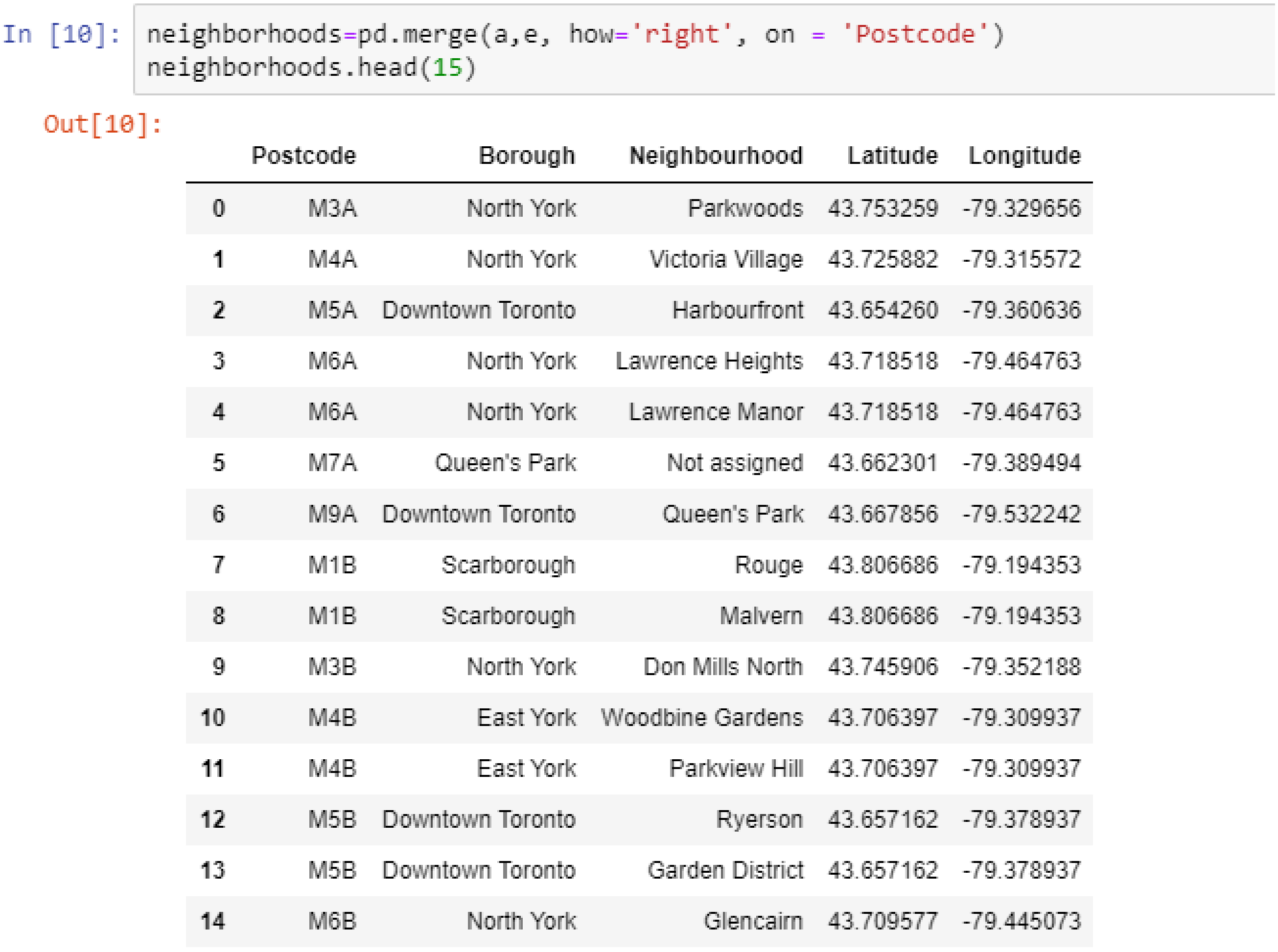
	Postal Code	Latitude	Longitude
0	M1B	43.806686	-79.194353
1	M1C	43.784535	-79.160497
2	M1E	43.763573	-79.188711
3	M1G	43.770992	-79.216917
4	M1H	43.773136	-79.239476
5	M1J	43.744734	-79.239476
6	M1K	43.727929	-79.262029
7	M1L	43.711112	-79.284577
8	M1M	43.716316	-79.239476
9	M1N	43.692657	-79.264848



```
In [9]: e.rename({'Postal Code': 'Postcode'}, axis=1, inplace=True)
        e.head(10)
```

Out[9]:

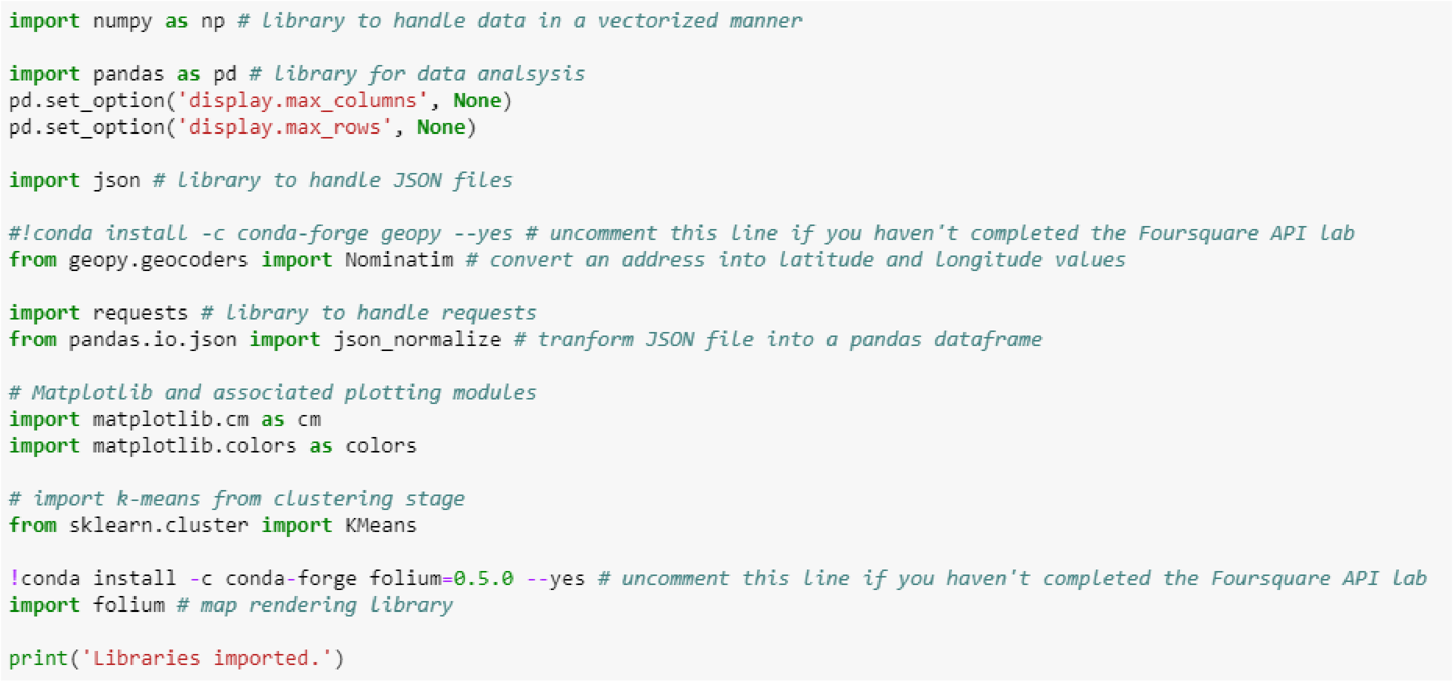
	Postcode	Latitude	Longitude
0	M1B	43.806686	-79.194353
1	M1C	43.784535	-79.160497
2	M1E	43.763573	-79.188711
3	M1G	43.770992	-79.216917
4	M1H	43.773136	-79.239476
5	M1J	43.744734	-79.239476
6	M1K	43.727929	-79.262029
7	M1L	43.711112	-79.284577
8	M1M	43.716316	-79.239476
9	M1N	43.692657	-79.264848



In [10]:
```
neighborhoods=pd.merge(a,e, how='right', on = 'Postcode')
neighborhoods.head(15)
```

Out[10]:

	Postcode	Borough	Neighbourhood	Latitude	Longitude
0	M3A	North York	Parkwoods	43.753259	-79.329656
1	M4A	North York	Victoria Village	43.725882	-79.315572
2	M5A	Downtown Toronto	Harbourfront	43.654260	-79.360636
3	M6A	North York	Lawrence Heights	43.718518	-79.464763
4	M6A	North York	Lawrence Manor	43.718518	-79.464763
5	M7A	Queen's Park	Not assigned	43.662301	-79.389494
6	M9A	Downtown Toronto	Queen's Park	43.667856	-79.532242
7	M1B	Scarborough	Rouge	43.806686	-79.194353
8	M1B	Scarborough	Malvern	43.806686	-79.194353
9	M3B	North York	Don Mills North	43.745906	-79.352188
10	M4B	East York	Woodbine Gardens	43.706397	-79.309937
11	M4B	East York	Parkview Hill	43.706397	-79.309937
12	M5B	Downtown Toronto	Ryerson	43.657162	-79.378937
13	M5B	Downtown Toronto	Garden District	43.657162	-79.378937
14	M6B	North York	Glencairn	43.709577	-79.445073



```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

#!conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

```
Solving environment: done

## Package Plan ##

  environment location: /opt/conda/envs/Python36

  added / updated specs:
    - folium=0.5.0


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    vincent-0.4.4              |             py_1          28 KB  conda-forge
    folium-0.5.0               |             py_0          45 KB  conda-forge
    altair-3.3.0               |           py36_0         747 KB  conda-forge
    branca-0.3.1               |             py_0          25 KB  conda-forge
    ca-certificates-2019.11.28 |       hecc5488_0         145 KB  conda-forge
    certifi-2019.11.28         |           py36_0         149 KB  conda-forge
    openssl-1.1.1d             |       h516909a_0         2.1 MB  conda-forge
    ------------------------------------------------------------
                                           Total:         3.2 MB

The following NEW packages will be INSTALLED:

    altair:          3.3.0-py36_0      conda-forge
    branca:          0.3.1-py_0        conda-forge
    folium:          0.5.0-py_0        conda-forge
    vincent:         0.4.4-py_1        conda-forge

The following packages will be UPDATED:

    ca-certificates: 2019.10.16-0                 --> 2019.11.28-hecc5488_0 conda-forge
    certifi:         2019.9.11-py36_0             --> 2019.11.28-py36_0     conda-forge

The following packages will be DOWNGRADED:

    openssl:         1.1.1d-h7b6447c_3            --> 1.1.1d-h516909a_0     conda-forge


Downloading and Extracting Packages
vincent-0.4.4        | 28 KB     | ##################################### | 100% 
folium-0.5.0         | 45 KB     | ##################################### | 100% 
altair-3.3.0         | 747 KB    | ##################################### | 100% 
branca-0.3.1         | 25 KB     | ##################################### | 100% 
ca-certificates-2019 | 145 KB    | ##################################### | 100% 
certifi-2019.11.28   | 149 KB    | ##################################### | 100% 
openssl-1.1.1d       | 2.1 MB    | ##################################### | 100% 
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Libraries imported.
```

In [12]:

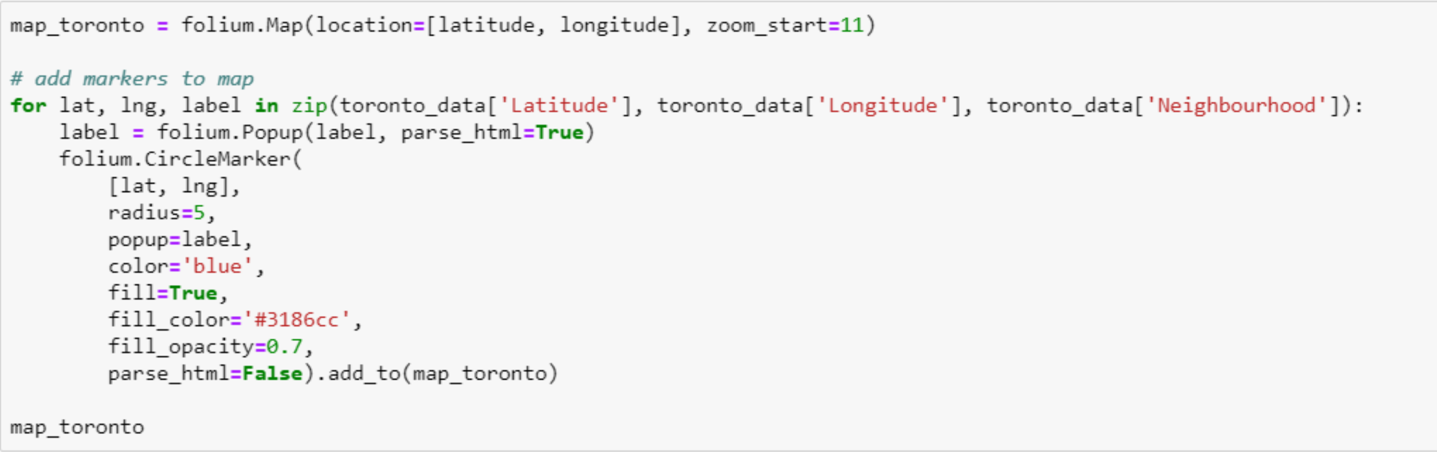
```
toronto_data= neighborhoods[neighborhoods['Borough'].str.contains('Toronto', na = False)].reset_index(drop=True)
toronto_data.head()
```

Out[12]:

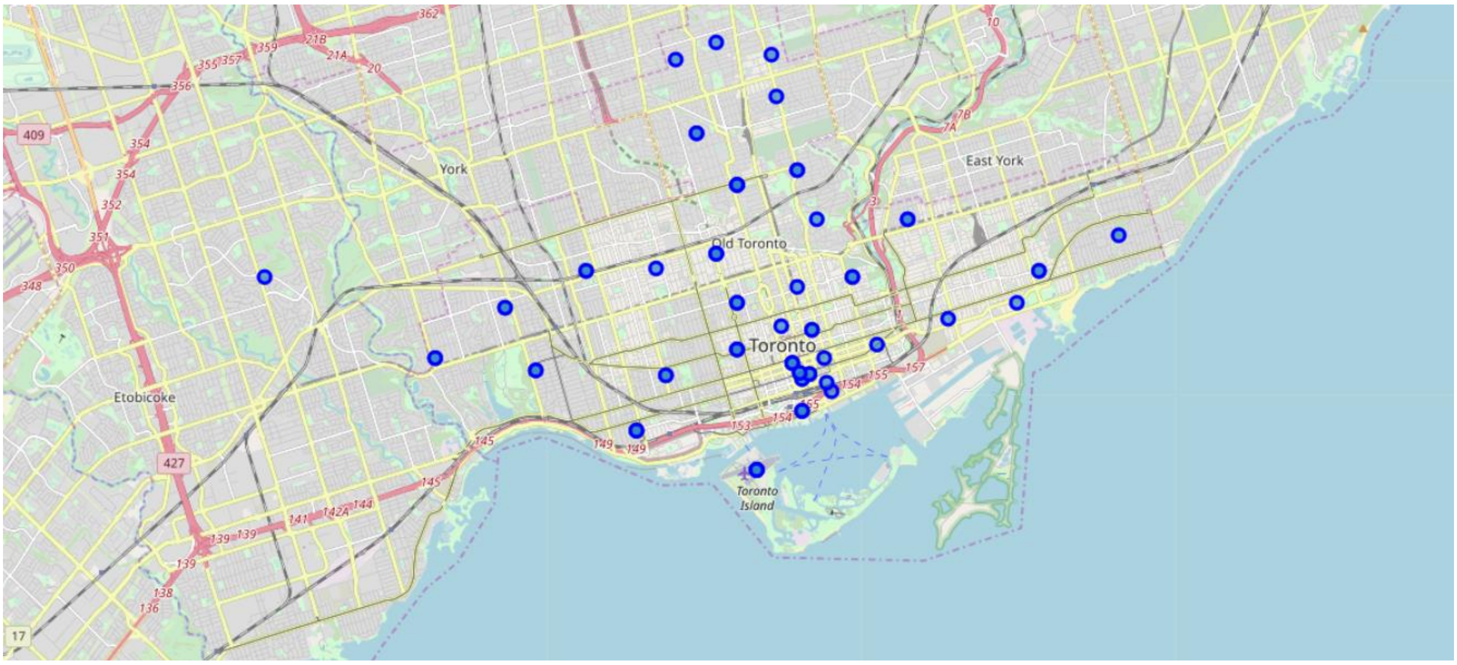
	Postcode	Borough	Neighbourhood	Latitude	Longitude
0	M5A	Downtown Toronto	Harbourfront	43.654260	-79.360636
1	M9A	Downtown Toronto	Queen's Park	43.667856	-79.532242
2	M5B	Downtown Toronto	Ryerson	43.657162	-79.378937
3	M5B	Downtown Toronto	Garden District	43.657162	-79.378937
4	M5C	Downtown Toronto	St. James Town	43.651494	-79.375418

In [13]:

```
latitude = 43.6532
longitude= -79.3832
```



```
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(toronto_data['Latitude'], toronto_data['Longitude'], toronto_data['Neighbourhood']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_toronto)

map_toronto
```



Recommendation

The following analysis can be improved with following extensions:

• Consider more categories. For example like "Night life" which is also a good source for customers. But also like "Restaurants", which even if not burger joints may be some concurrence if too many.

• In the Locality itself, it can also be computed the distance between all the venues in order to find a place with the most number of potential customers.

• Using smaller geographical areas like Neighborhoods could improve the accuracy for the scores.