Data Science Presentation

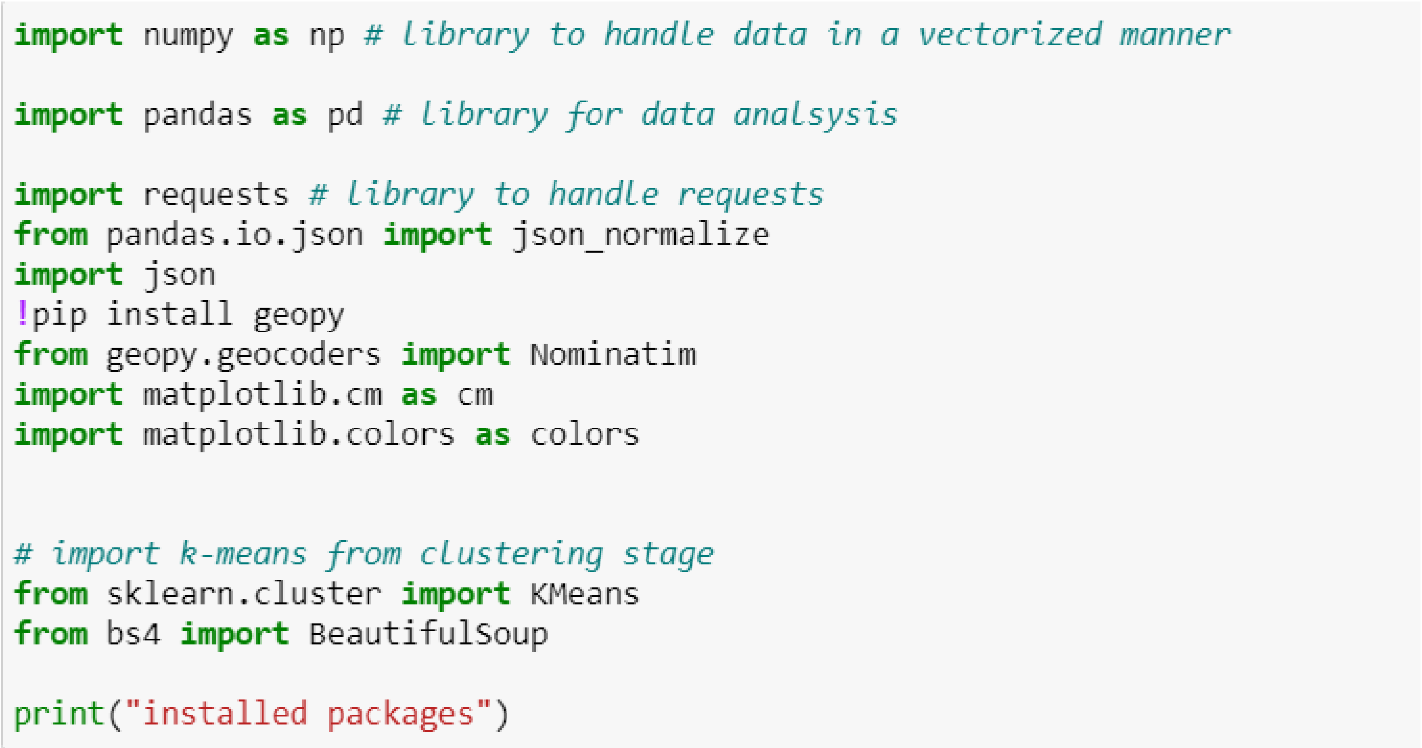
Introduction

- Toronto, Canada there is a ever expanding demand for new restaurants as new tech companies roll in and residents demands new food options
- Every famous chain from Canada usually has some start in Toronto, but from all the 82 chains started in Canada there are only a handful of Pizzerias
- We must capitalize on the markets lack of exposure to Pizza before anyone else does to claim the spot as the top Pizza restaurant in Canada.

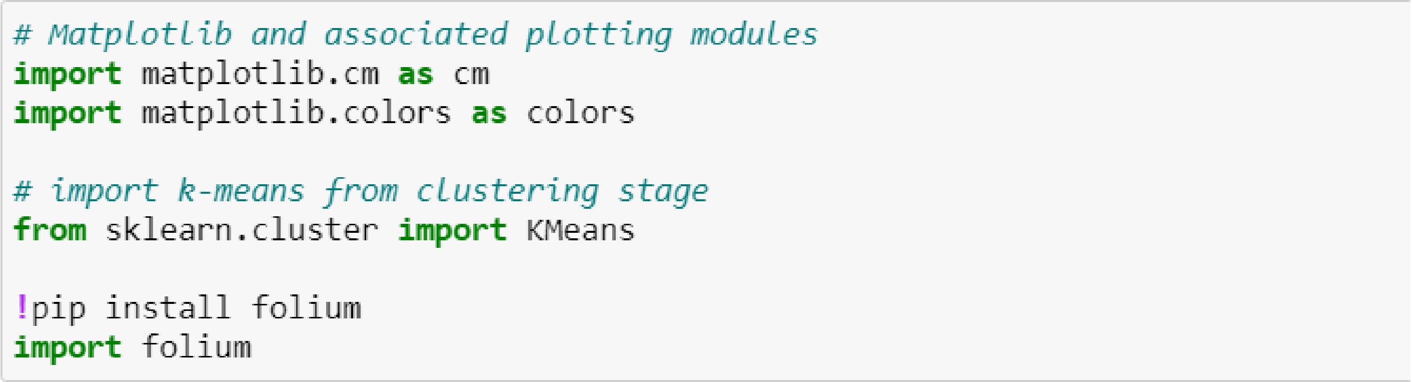
Business Problem

- Location, location, location. The new restaurant must not be too far from the food scene to be a burden for customers.
- The best way to figure that out is to see how many amenities there are nearby to attract local and visiting Toronto tourists.
- With an estimated 1.8 Million new jobs opening within the restaurant industry it's easy to see why now the perfect time to open a new restaurant.

Libraries installed



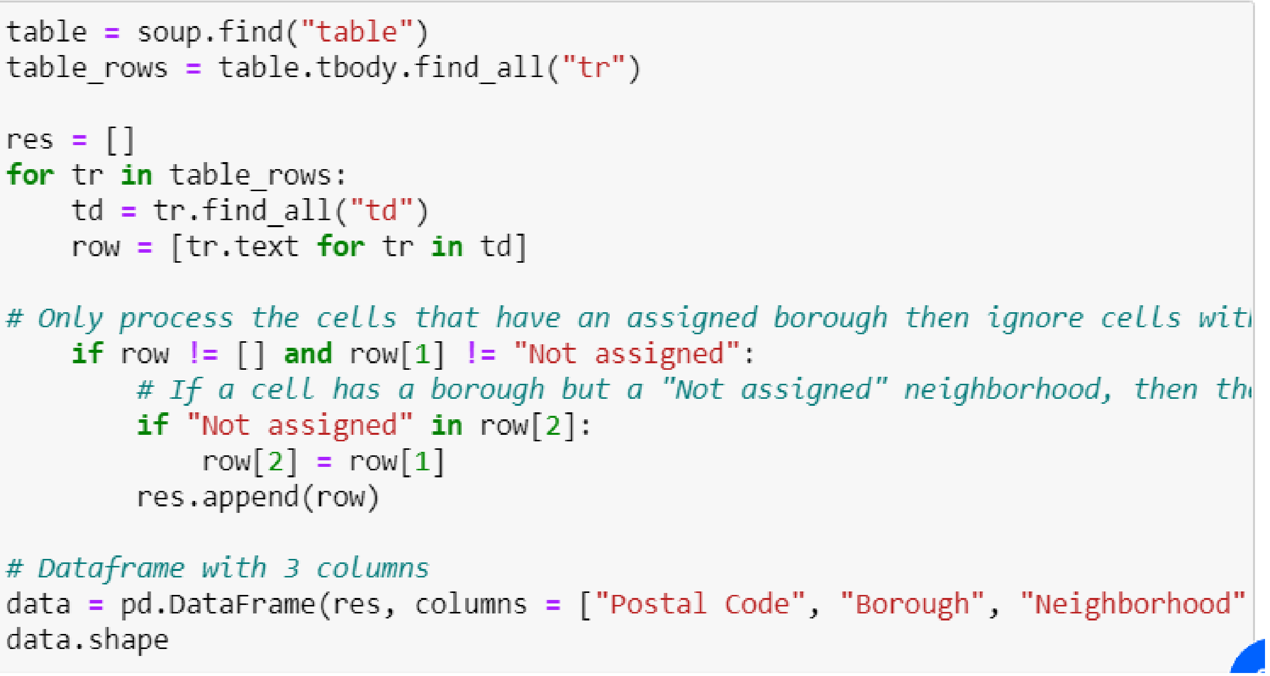
```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis

import requests # library to handle requests
from pandas.io.json import json_normalize
import json
!pip install geopy
from geopy.geocoders import Nominatim
import matplotlib.cm as cm
import matplotlib.colors as colors


# import k-means from clustering stage
from sklearn.cluster import KMeans
from bs4 import BeautifulSoup

print("installed packages")
```



```
# 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

!pip install folium
import folium
```

Data Collection

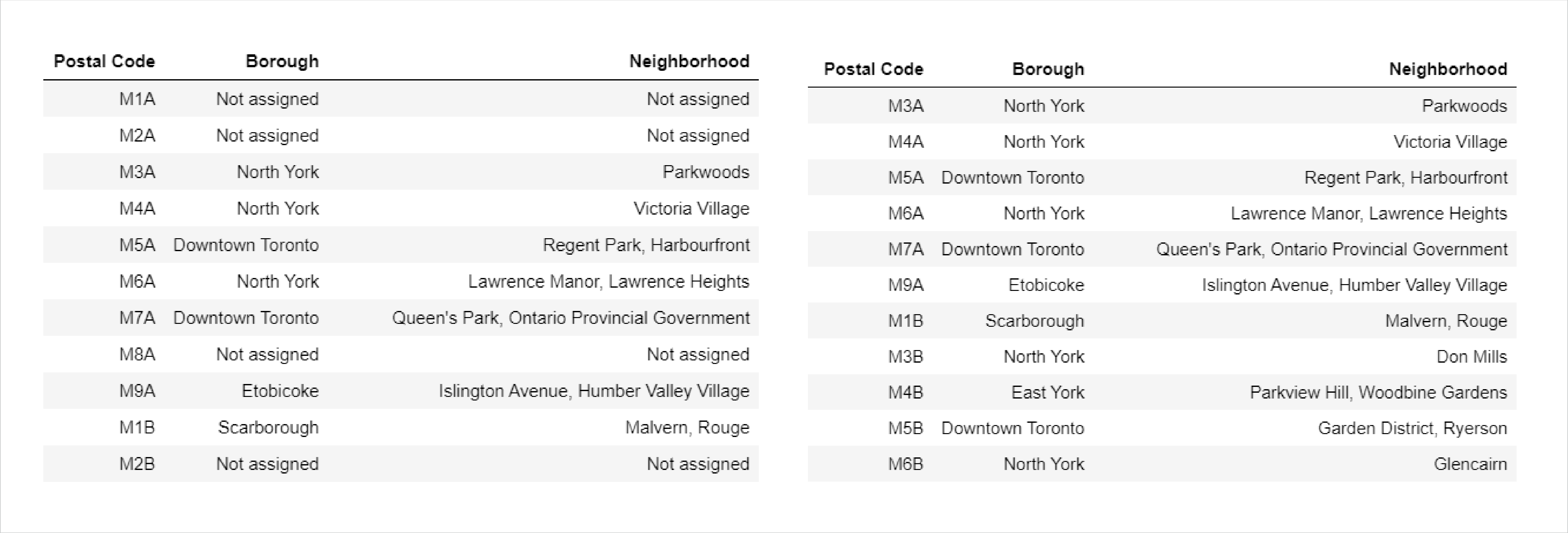
- data extracted from Wikipedia we will use BeautifulSoup to extract and eventually place all information



```
table = soup.find("table")
table_rows = table.tbody.find_all("tr")

res = []
for tr in table_rows:
    td = tr.find_all("td")
    row = [tr.text for tr in td]

# Only process the cells that have an assigned borough then ignore cells wit
    if row != [] and row[1] != "Not assigned":
        # If a cell has a borough but a "Not assigned" neighborhood, then th
        if "Not assigned" in row[2]:
            row[2] = row[1]
        res.append(row)

# Dataframe with 3 columns
data = pd.DataFrame(res, columns = ["Postal Code", "Borough", "Neighborhood"
data.shape
```

Data Sorting

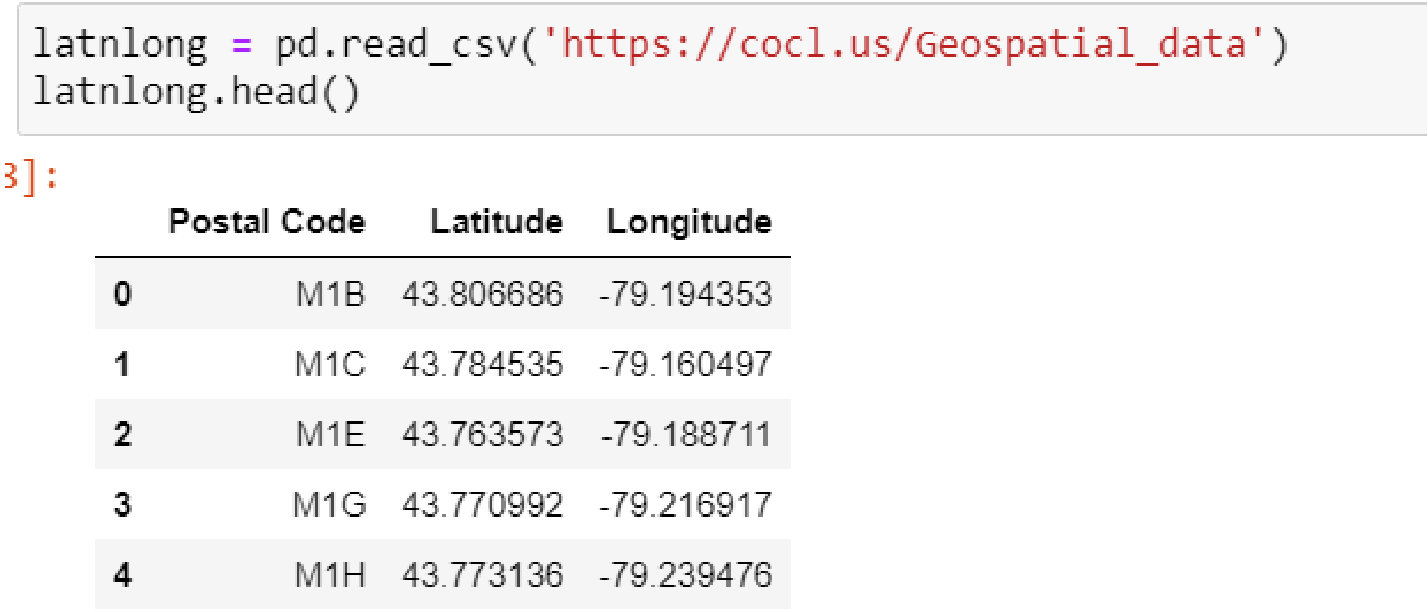
Turning a data frame into something more ready to be used



Postal Code	Borough	Neighborhood
M1A	Not assigned	Not assigned
M2A	Not assigned	Not assigned
M3A	North York	Parkwoods
M4A	North York	Victoria Village
M5A	Downtown Toronto	Regent Park, Harbourfront
M6A	North York	Lawrence Manor, Lawrence Heights
M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government
M8A	Not assigned	Not assigned
M9A	Etobicoke	Islington Avenue, Humber Valley Village
M1B	Scarborough	Malvern, Rouge
M2B	Not assigned	Not assigned

Postal Code	Borough	Neighborhood
M3A	North York	Parkwoods
M4A	North York	Victoria Village
M5A	Downtown Toronto	Regent Park, Harbourfront
M6A	North York	Lawrence Manor, Lawrence Heights
M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government
M9A	Etobicoke	Islington Avenue, Humber Valley Village
M1B	Scarborough	Malvern, Rouge
M3B	North York	Don Mills
M4B	East York	Parkview Hill, Woodbine Gardens
M5B	Downtown Toronto	Garden District, Ryerson
M6B	North York	Glencairn

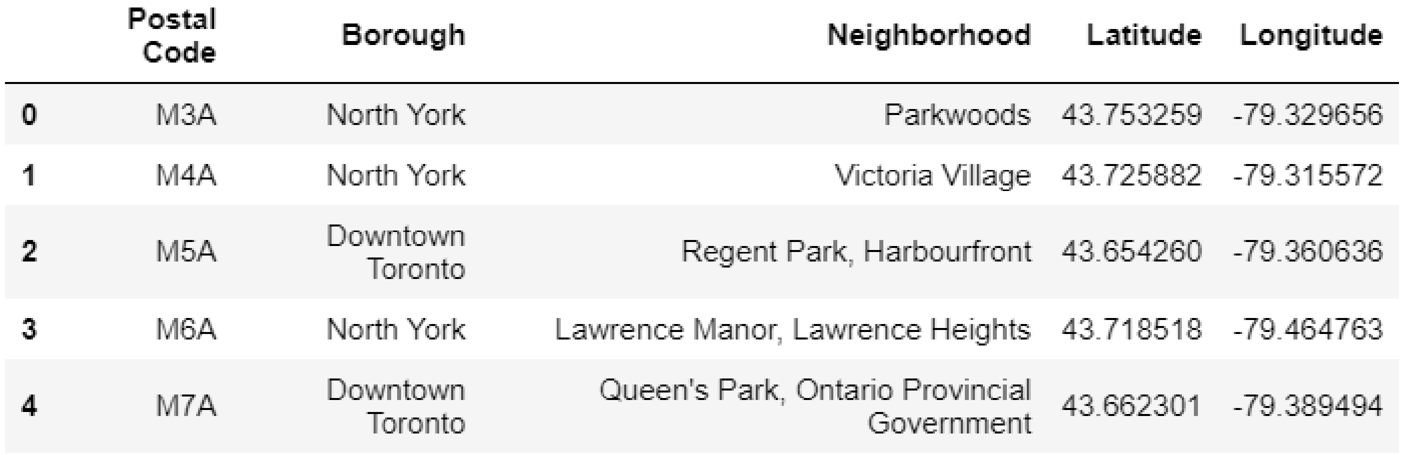
Adding the Locations to the neighborhoods



```
latnlong = pd.read_csv('https://cocl.us/Geospatial_data')
latnlong.head()
```

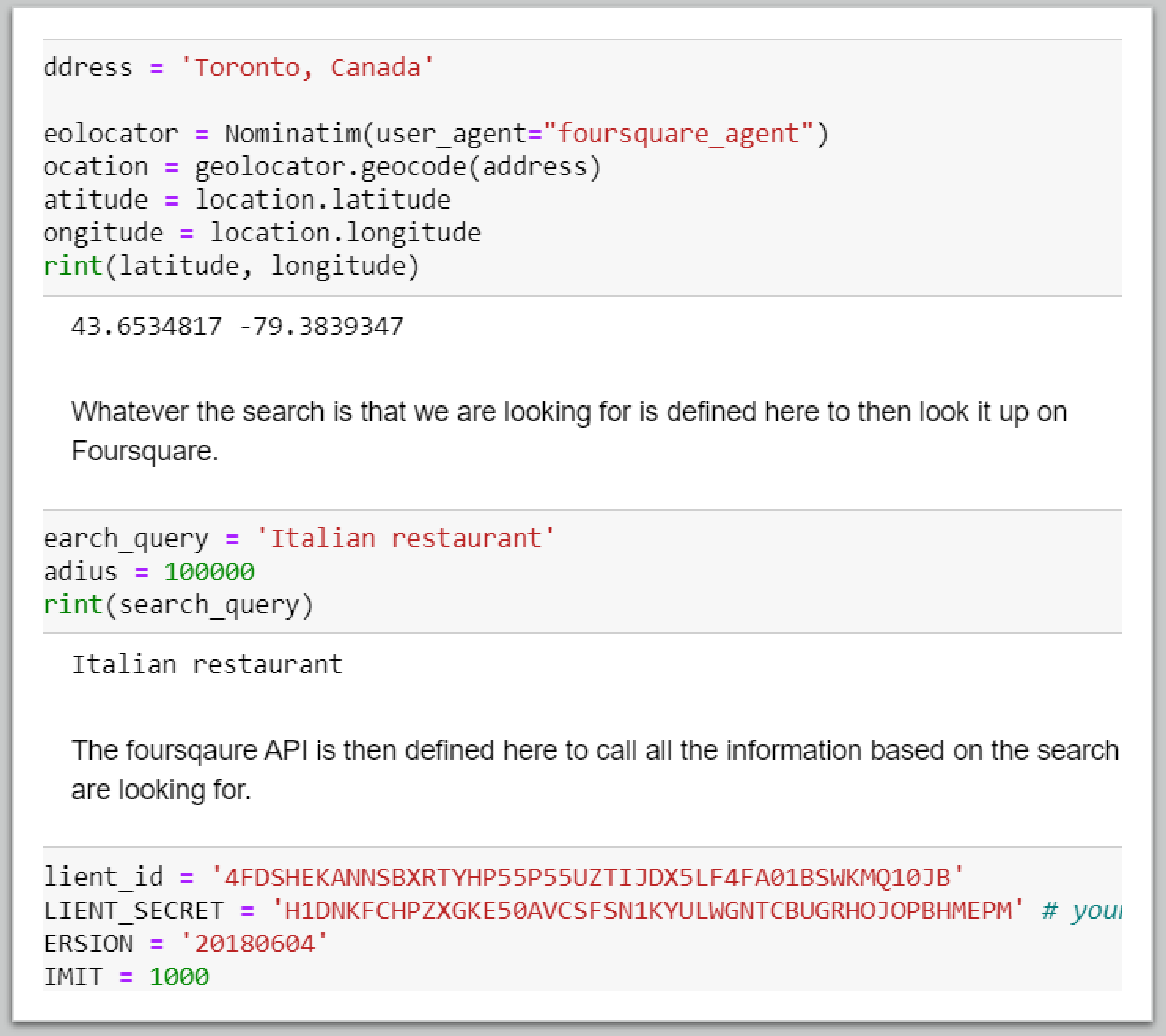
3]:

	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



	Postal Code	Borough	Neighborhood	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	Regent Park, Harbourfront	43.654260	-79.360636
3	M6A	North York	Lawrence Manor, Lawrence Heights	43.718518	-79.464763
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494

Now lets connect to the Foursquare API



```
ddress = 'Toronto, Canada'

eolocator = Nominatim(user_agent="foursquare_agent")
ocation = geolocator.geocode(address)
atitude = location.latitude
ongitude = location.longitude
rint(latitude, longitude)
```

```
43.6534817 -79.3839347
```

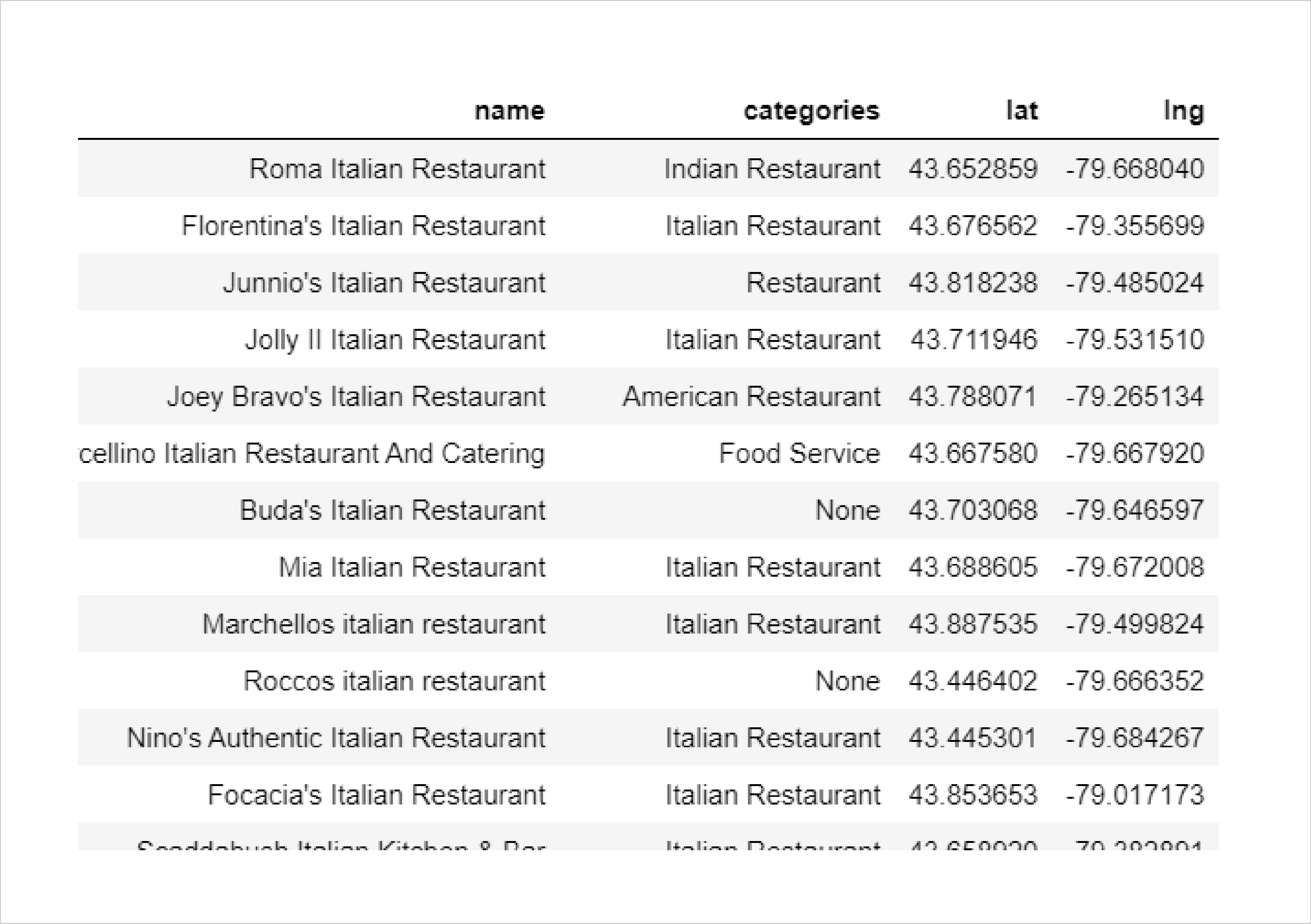
Whatever the search is that we are looking for is defined here to then look it up on Foursquare.

```
earch_query = 'Italian restaurant'
adius = 100000
rint(search_query)
```

```
Italian restaurant
```

The foursqaure API is then defined here to call all the information based on the search are looking for.

```
lient_id = '4FDSHEKANNSBXRTYHP55P55UZTIJDX5LF4FA01BSWKMQ10JB'
LIENT_SECRET = 'H1DNKFCHPZXGKE50AVCSFSN1KYULWGNTCBUGRHOJOPBHMEPM' # you
ERSION = '20180604'
IMIT = 1000
```

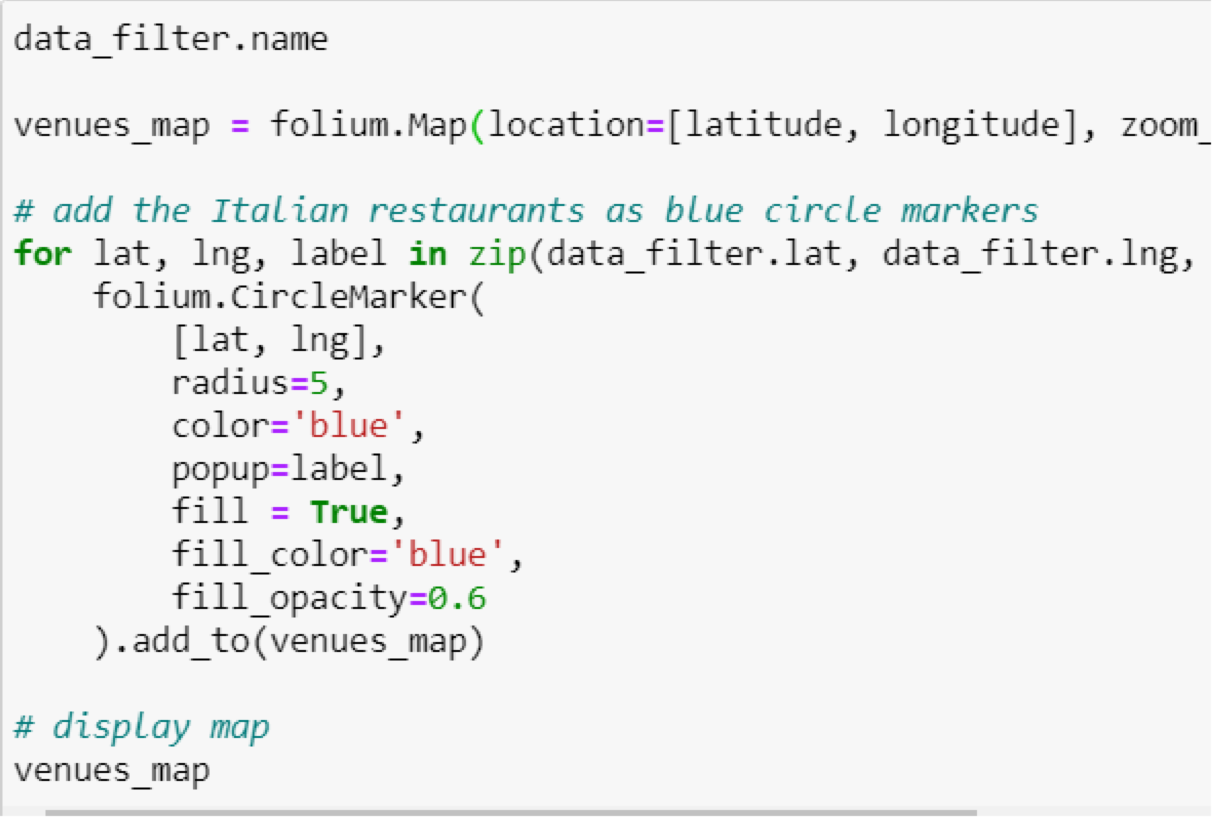


name	categories	lat	lng
Roma Italian Restaurant	Indian Restaurant	43.652859	-79.668040
Florentina's Italian Restaurant	Italian Restaurant	43.676562	-79.355699
Junnio's Italian Restaurant	Restaurant	43.818238	-79.485024
Jolly II Italian Restaurant	Italian Restaurant	43.711946	-79.531510
Joey Bravo's Italian Restaurant	American Restaurant	43.788071	-79.265134
cellino Italian Restaurant And Catering	Food Service	43.667580	-79.667920
Buda's Italian Restaurant	None	43.703068	-79.646597
Mia Italian Restaurant	Italian Restaurant	43.688605	-79.672008
Marchellos italian restaurant	Italian Restaurant	43.887535	-79.499824
Roccos italian restaurant	None	43.446402	-79.666352
Nino's Authentic Italian Restaurant	Italian Restaurant	43.445301	-79.684267
Focacia's Italian Restaurant	Italian Restaurant	43.853653	-79.017173
Scaddabush Italian Kitchen & Bar	Italian Restaurant	43.658020	-79.382801

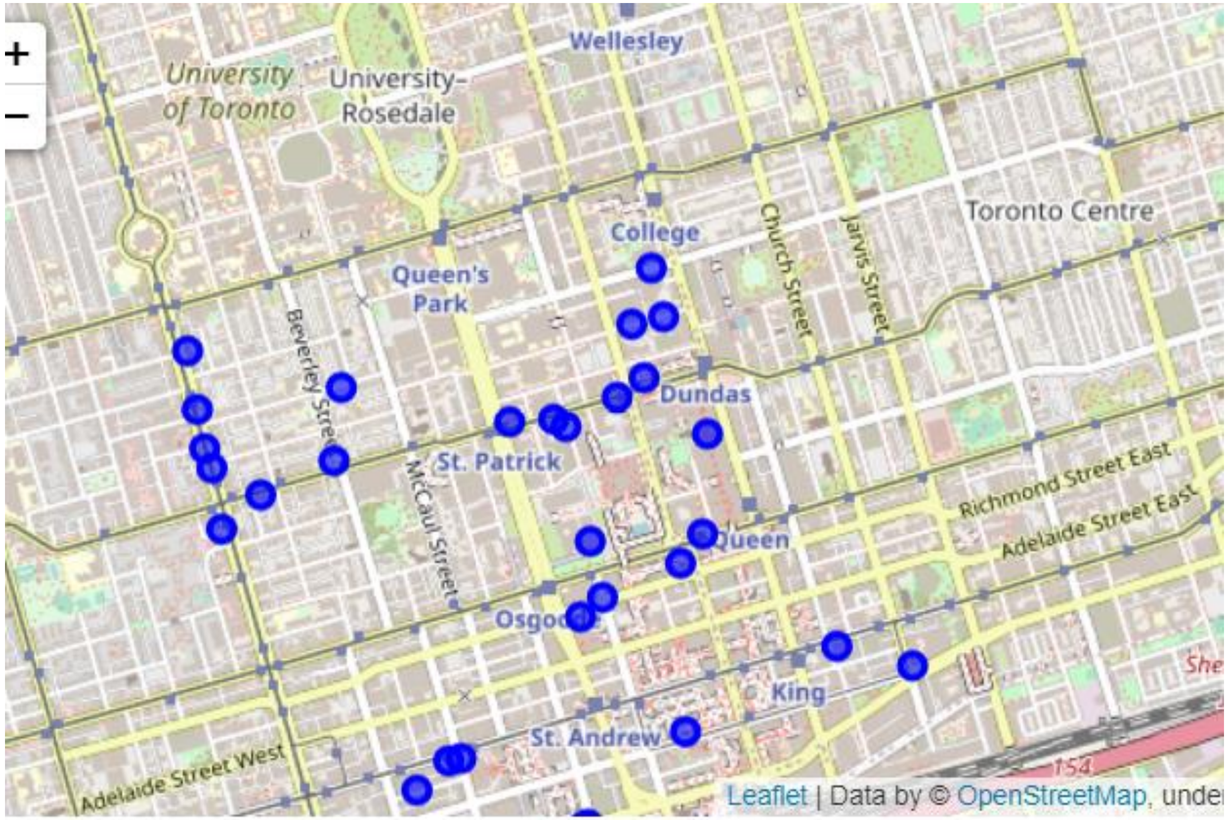
Using if else statements we can filter out our data frame and clean it up

Data Exploration

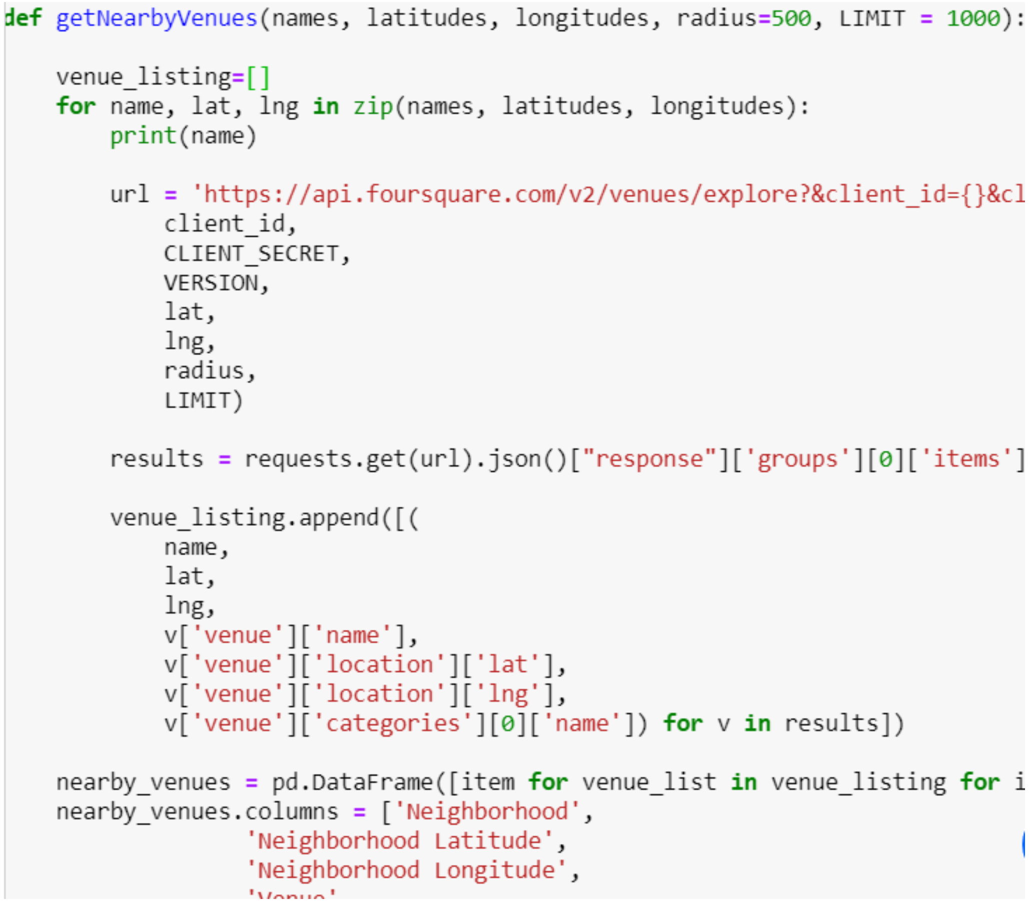
Let's visualize our competing Italian restaurants



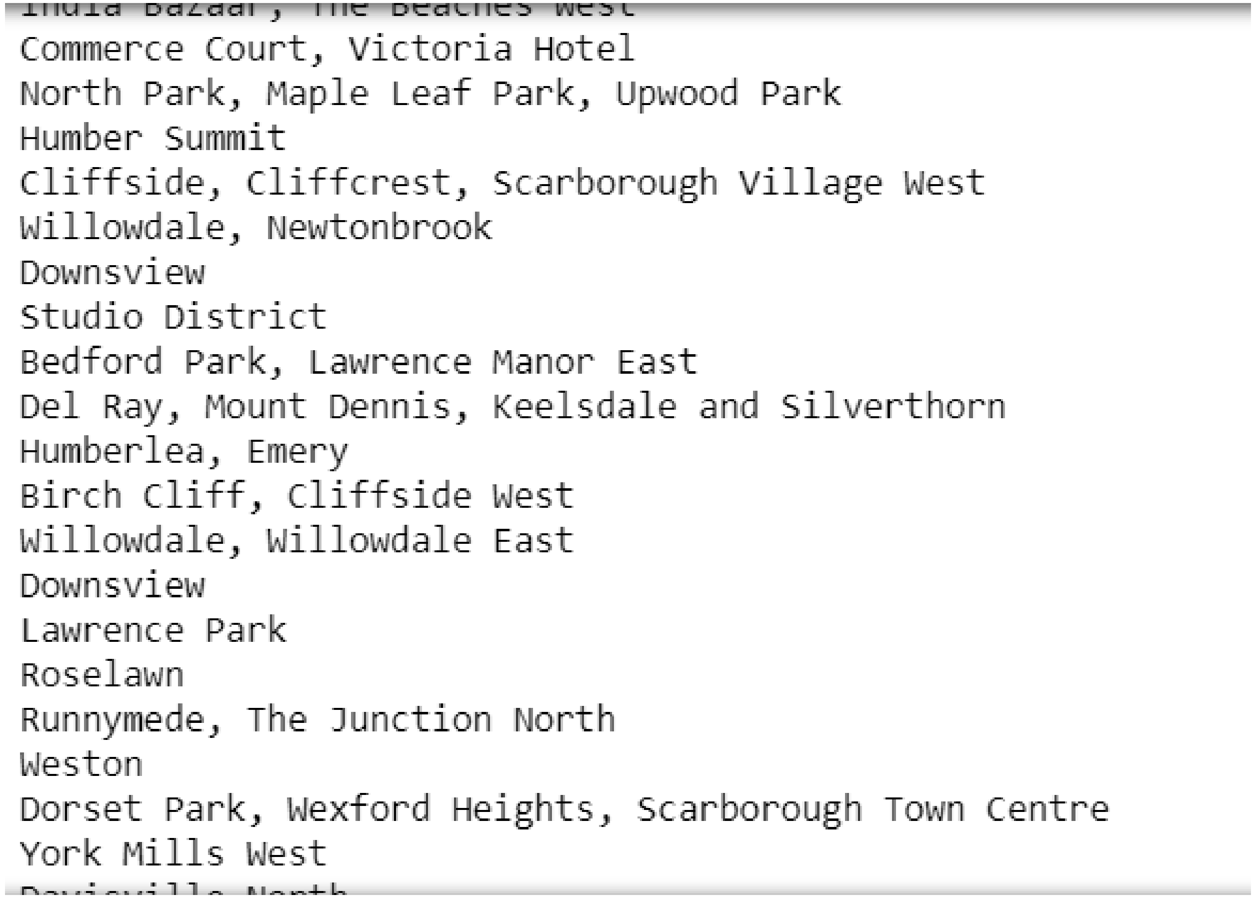
```
data_filter.name

venues_map = folium.Map(location=[latitude, longitude], zoom_

# add the Italian restaurants as blue circle markers
for lat, lng, label in zip(data_filter.lat, data_filter.lng,
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        color='blue',
        popup=label,
        fill = True,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(venues_map)

# display map
venues_map
```



To better understand the neighborhoods we will gather all surrounding venues



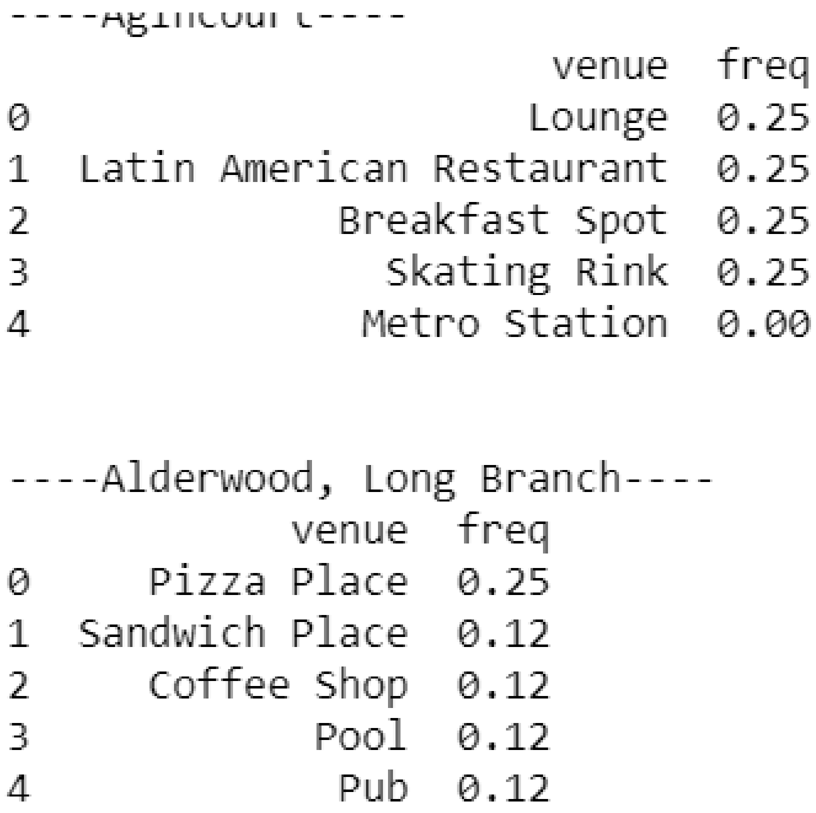
```
def getNearbyVenues(names, latitudes, longitudes, radius=500, LIMIT = 1000):

    venue_listing=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&cl
            client_id,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        results = requests.get(url).json()["response"]['groups'][0]['items']

        venue_listing.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venue_listing for i
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
```



```
India Bazaar, The Beaches West
Commerce Court, Victoria Hotel
North Park, Maple Leaf Park, Upwood Park
Humber Summit
Cliffside, Cliffcrest, Scarborough Village West
Willowdale, Newtonbrook
Downsview
Studio District
Bedford Park, Lawrence Manor East
Del Ray, Mount Dennis, Keelsdale and Silverthorn
Humberlea, Emery
Birch Cliff, Cliffside West
Willowdale, Willowdale East
Downsview
Lawrence Park
Roselawn
Runnymede, The Junction North
Weston
Dorset Park, Wexford Heights, Scarborough Town Centre
York Mills West
Davisville North
```

Machine Learning

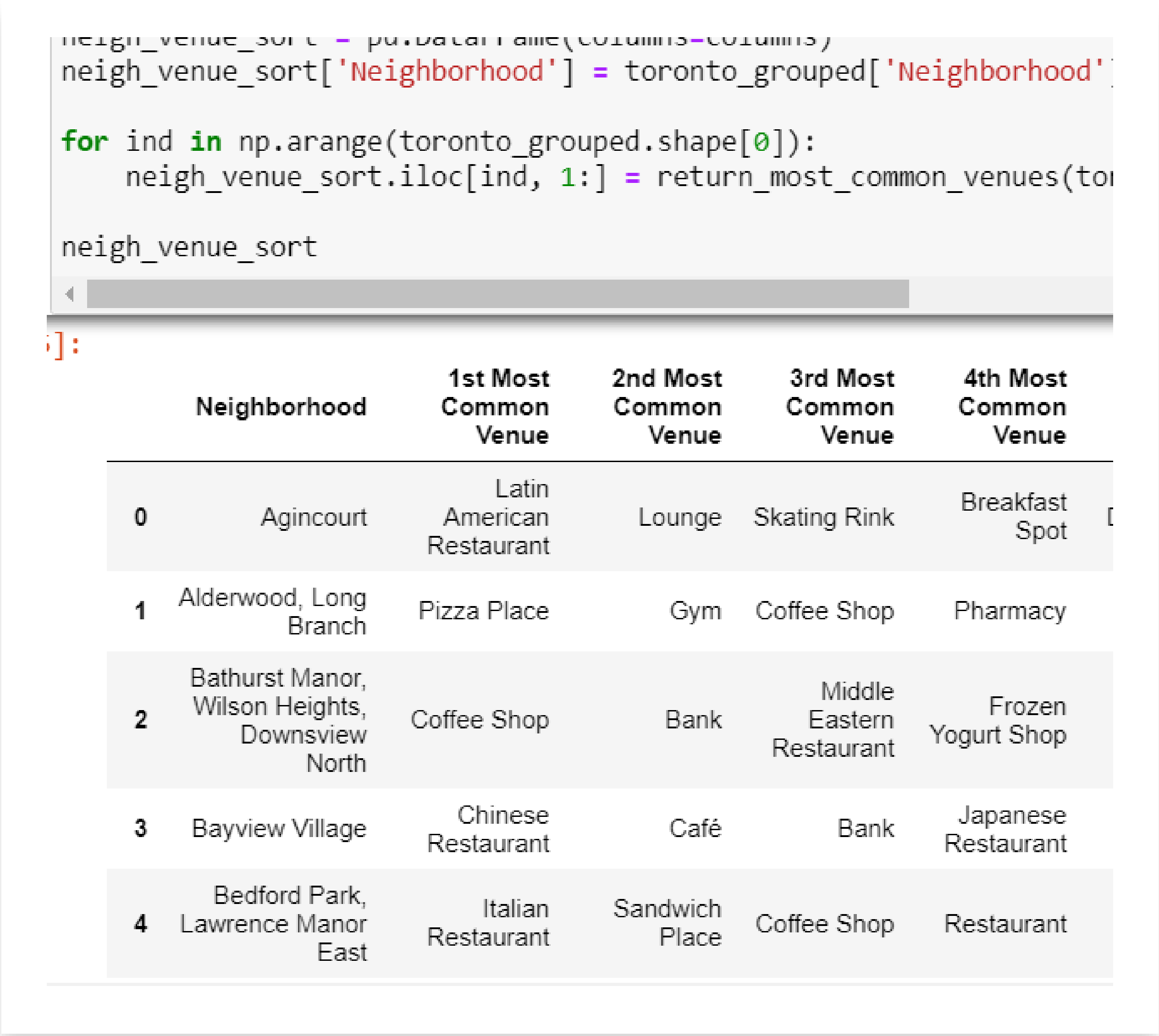
- Using One Hot encoding we can understand the frequency of certain categories of venues.



```
----Agincourt----
                        venue  freq
0                      Lounge  0.25
1  Latin American Restaurant  0.25
2              Breakfast Spot  0.25
3                Skating Rink  0.25
4               Metro Station  0.00


----Alderwood, Long Branch----
            venue  freq
0     Pizza Place  0.25
1  Sandwich Place  0.12
2     Coffee Shop  0.12
3            Pool  0.12
4             Pub  0.12
```

We can now visualize what each neighborhood has to offer in terms of venues

Data Analysis

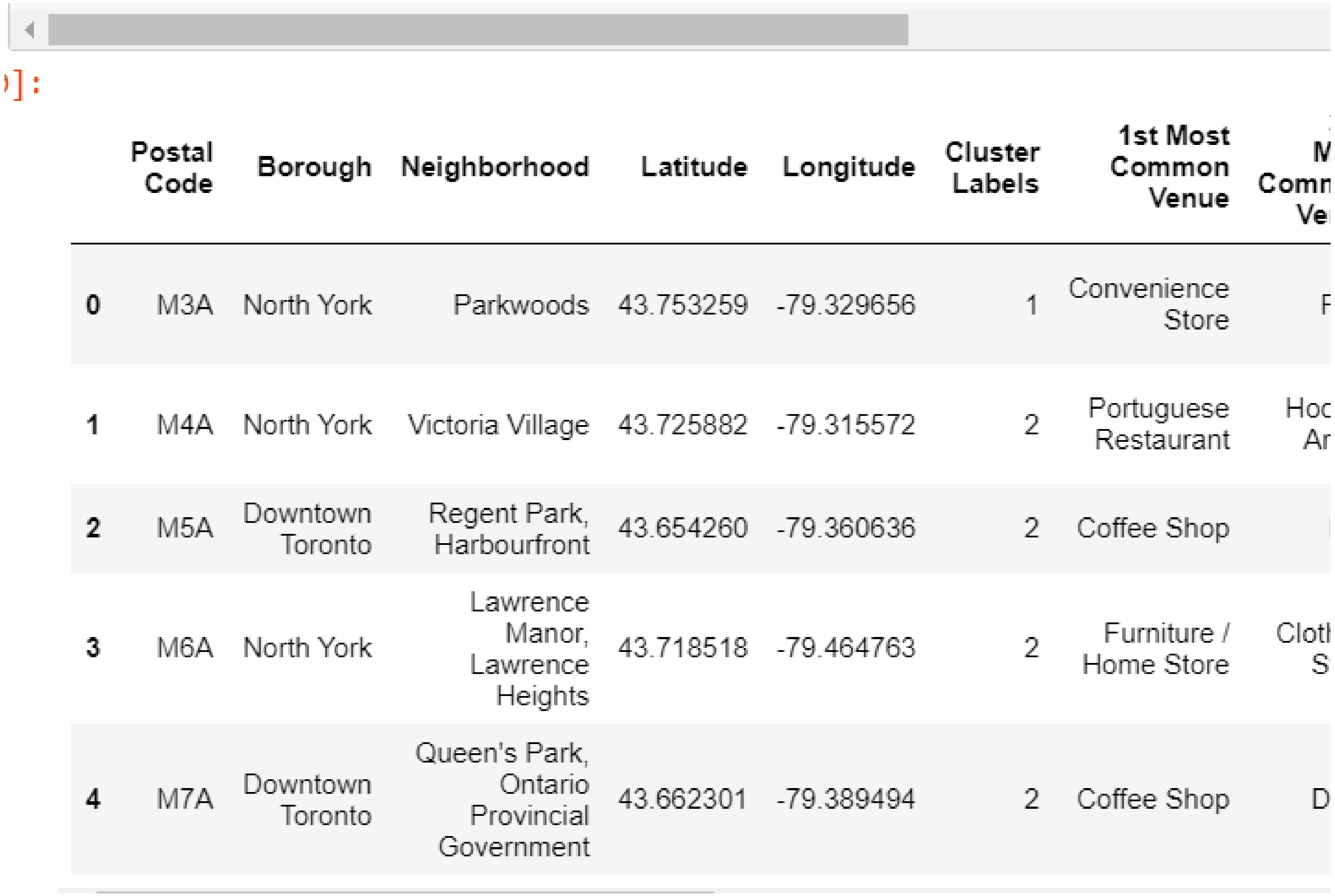


```
neigh_venue_sort = pd.DataFrame(columns=columns)
neigh_venue_sort['Neighborhood'] = toronto_grouped['Neighborhood'

for ind in np.arange(toronto_grouped.shape[0]):
    neigh_venue_sort.iloc[ind, 1:] = return_most_common_venues(tor

neigh_venue_sort
```

]:

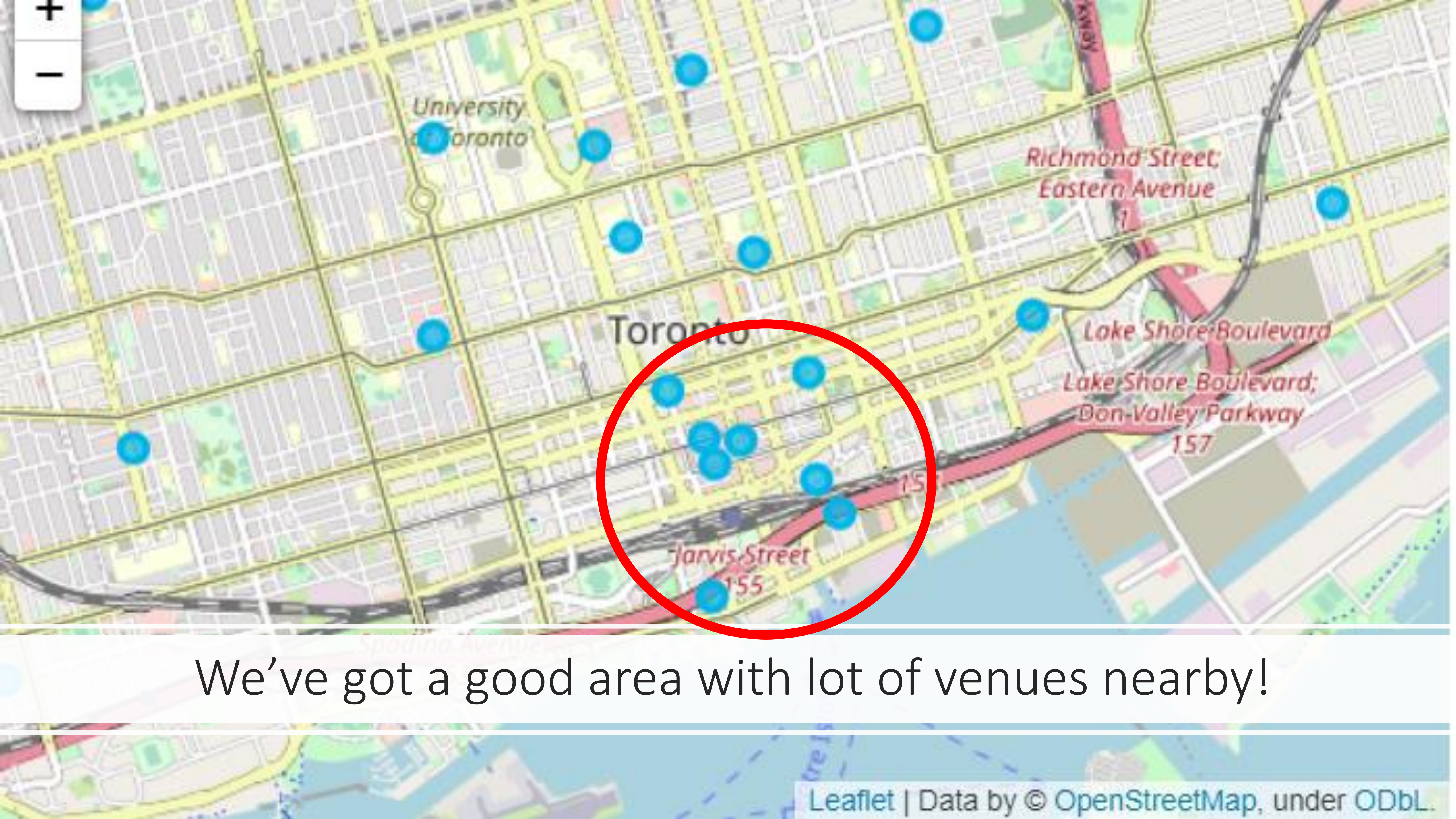
	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue
0	Agincourt	Latin American Restaurant	Lounge	Skating Rink	Breakfast Spot
1	Alderwood, Long Branch	Pizza Place	Gym	Coffee Shop	Pharmacy
2	Bathurst Manor, Wilson Heights, Downsview North	Coffee Shop	Bank	Middle Eastern Restaurant	Frozen Yogurt Shop
3	Bayview Village	Chinese Restaurant	Café	Bank	Japanese Restaurant
4	Bedford Park, Lawrence Manor East	Italian Restaurant	Sandwich Place	Coffee Shop	Restaurant

Let's go ahead and merge all the data frames we’ve gathered up to now



]:

	Postal Code	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	1st Most Common Venue	M Comn Ve
0	M3A	North York	Parkwoods	43.753259	-79.329656	1	Convenience Store	F
1	M4A	North York	Victoria Village	43.725882	-79.315572	2	Portuguese Restaurant	Hoc Ar
2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636	2	Coffee Shop	
3	M6A	North York	Lawrence Manor, Lawrence Heights	43.718518	-79.464763	2	Furniture / Home Store	Clotl S
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494	2	Coffee Shop	D


+
−
University of Toronto
Toronto
Richmond Street; Eastern Avenue
Lake Shore Boulevard
Lake Shore Boulevard; Don Valley Parkway
157
Jarvis Street
155
Leaflet | Data by © OpenStreetMap, under ODbL.

We’ve got a good area with lot of venues nearby!

Discussion/Results

- Overall, there are several tools that can be used to understand the layout of Toronto’s food and entertainment scene. The tools used for this capstone may not be the best for every scenario, but they provide users with good visualization and easy to understand steps to sort data. This showed us that extracting data from online sources is important to best understand relevant problems as we move into a more data dependent world. The results from this capstone helped a small business navigate the busy streets of Toronto without once having to step outside. This becomes more and more important as we become more globalized but require information over places we may have never been. Sources like Foursquare API allow users to gather information over locations around the world and empower use to keep discovering.

Conclusion

- In conclusion we see that downtown Toronto is the best place to put our restaurant, more specifically near the Toronto union station as a lot of venues that attract food traffic will low through that area.
- For a restaurant, having great visibility by anyone is the most important thing. In the end, this course taught us to us data to tell a story, and for this pizza shop, this story is just about to begin because of data science methods.