Project Capstone: Battle of Neighbourhoods - Week 1_Part2

1. Introduction/Business Problem

Toronto is the capital city of the Canadian province of Ontario. With a recorded population of 2,731,571 in 2016, It is the most populous city in Canada and the fourth most populous city in North America. The city is the anchor of the Golden Horseshoe, an urban agglomeration of 9,245,438 people (as of 2016) surrounding the western end of Lake Ontario, while the Greater Toronto Area (GTA) proper had a 2016 population of 6,417,516. Toronto is an international centre of business, finance, arts, and culture, and is recognized as one of the most multicultural and cosmopolitan cities in the world the diverse population of Toronto reflects its current and historical role as an important destination for immigrants to Canada. More than 50 percent of residents belong to a visible minority population group, and over 200 distinct ethnic origins are represented among its inhabitants. While the majority of Torontonians speak English as their primary language, over 160 languages are spoken in the city

2. Business Problems

At Toronto there no information relate Neighbourhoods is there any a hotel/restaurant near this area within 1000 M. what kind of most top ten "Common Venues" popular entertainment on those area as a venues categories such as: Fast food Restaurant, Bar, coffe shop, etc.

3. Expected / Interested Audience

By giving the solution of information from the above business problems such as explore the maps & display list the most top ten "Common Venues" around those area.

4. Data Collection

To complete this case, we require data set as follows:

Part_1:

- Venue data relate to Hotel & Restaurant. This will help us find the neighborhoods that are more suitable to find the best hotel & Restaurant

Part_2:

- List of neighborhoods in Toronto, Ontario Canada
- Latitute and Longiture of these neigborhoods

5. Extracting The Data

Scrapping of Toronto neighborhoods via Wikipedia Getting Latitute and Longitude data of these neighborhoods via Geocoder packages

Importing Library

Importing Library

download the library that will be use for this projects as follows: 1). Folium 2). Beautifoulsoap 3). geopy

In [87]:
```
# Import libraries
import numpy as np # data in a vectorized manner manipulation
import pandas as pd # data analsysis

from pandas.io.json import json_normalize # tranform JSON file into a pandas data
import json # JSON files manipulation

from sklearn.cluster import KMeans # clustering algorithm

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

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

#!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim

#!conda install -c conda-forge folium=0.5.0 --yes
import folium # map rendering library
import requests # HTTP library
from bs4 import BeautifulSoup # scraping library

print("@@Libraries Imported@@")
```

```
@@Libraries Imported@@
```

Data Collection

- List of neighborhoods in Toronto, Ontario Canada
- Latitute and Longiture of these neigborhoods
- Venue data relate to Hotel & Restaurant. This will help us find the neighborhoods that are more suitable to find the best hotel & Restaurant

Website scraping with BeautifulSoup

Use the Notebook to build the code to scrape the following Wikipedia page, https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M (https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M), in order to obtain the data that is in the table of postal codes and to transform the data into a pandas dataframe like the one shown below:

In [88]:
```
url = "https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M"

resp = requests.get(url)
toronto_html = BeautifulSoup(resp.content)
```

In [89]:
```
soup                = BeautifulSoup(str(toronto_html))
neighborhood_table = soup.find('table')
table_str           = str(neighborhood_table.extract())
toronto_df          = pd.read_html(table_str)[0]

toronto_df.head()
```

Out[89]:

	Postal Code	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	Regent Park, Harbourfront

In [90]:
```
toronto_df.shape
```

Out[90]: (180, 3)

Display All data & Cleansing at column Borough "Not assigned"

In [91]:
```
toronto_df.Borough.value_counts()
```

Out[91]:
```
Not assigned        77
North York          24
Downtown Toronto    19
Scarborough         17
Etobicoke           12
Central Toronto      9
West Toronto         6
East Toronto         5
York                 5
East York            5
Mississauga          1
Name: Borough, dtype: int64
```

In [92]:
```
Borough_va = toronto_df[toronto_df.Borough == 'Not assigned']
toronto_df.drop(Borough_va.index, inplace=True)
```

In [93]:
```
toronto_df.shape
```

Out[93]: (103, 3)

In [94]:
```
toronto_df.head()
```

Out[94]:

	Postal Code	Borough	Neighbourhood
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Regent Park, Harbourfront
5	M6A	North York	Lawrence Manor, Lawrence Heights
6	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government

Group by Postal Code & Borough

In [95]:
```
toronto_df = toronto_df.groupby(['Postal Code', 'Borough'])['Neighbourhood'].appl
```

In [96]:
```
toronto_df.reset_index(inplace=True)
toronto_df.head()
```

Out[96]:

	Postal Code	Borough	Neighbourhood
0	M1B	Scarborough	Malvern, Rouge
1	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek
2	M1E	Scarborough	Guildwood, Morningside, West Hill
3	M1G	Scarborough	Woburn
4	M1H	Scarborough	Cedarbrae

Check the column Neighbourhood after cleansing with the value of "Not assigned"

In [97]:
```
toronto_df[toronto_df.Neighbourhood == "Not assigned"]
```

Out[97]:

	Postal Code	Borough	Neighbourhood

In [98]:
```
toronto_df.head()
```

Out[98]:

	Postal Code	Borough	Neighbourhood
0	M1B	Scarborough	Malvern, Rouge
1	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek
2	M1E	Scarborough	Guildwood, Morningside, West Hill
3	M1G	Scarborough	Woburn
4	M1H	Scarborough	Cedarbrae

In [99]:
```
toronto_df.shape
```

Out[99]: (103, 3)

Built a dataframe of the postal code of each neighborhood along with the borough name and neighborhood name, in order to utilize the Foursquare location data, we need to get the latitude and the longitude coordinates of each neighborhood.

In [100]:
```
geo_url = "https://cocl.us/Geospatial_data"
geocode_df = pd.read_csv(geo_url)

geocode_df.head()
```

Out[100]:

	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

Merge toronto_df with goecode_df into one dataframes

In [101]:
```
toronto_df = pd.merge(toronto_df, geocode_df, on='Postal Code')
toronto_df.head()
```

Out[101]:

	Postal Code	Borough	Neighbourhood	Latitude	Longitude
0	M1B	Scarborough	Malvern, Rouge	43.806686	-79.194353
1	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek	43.784535	-79.160497
2	M1E	Scarborough	Guildwood, Morningside, West Hill	43.763573	-79.188711
3	M1G	Scarborough	Woburn	43.770992	-79.216917
4	M1H	Scarborough	Cedarbrae	43.773136	-79.239476

Extracting The Data

- Scrapping of Toronto neighborhoods via Wikipedia
- Getting Latitute and Longitude data of these neighborhoods via Geocoder packages

In [102]:
```
address = 'Toronto, Ontario'

geolocator = Nominatim(user_agent="toronto_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Toronto are {}, {}.'.format(latitude, longit
```

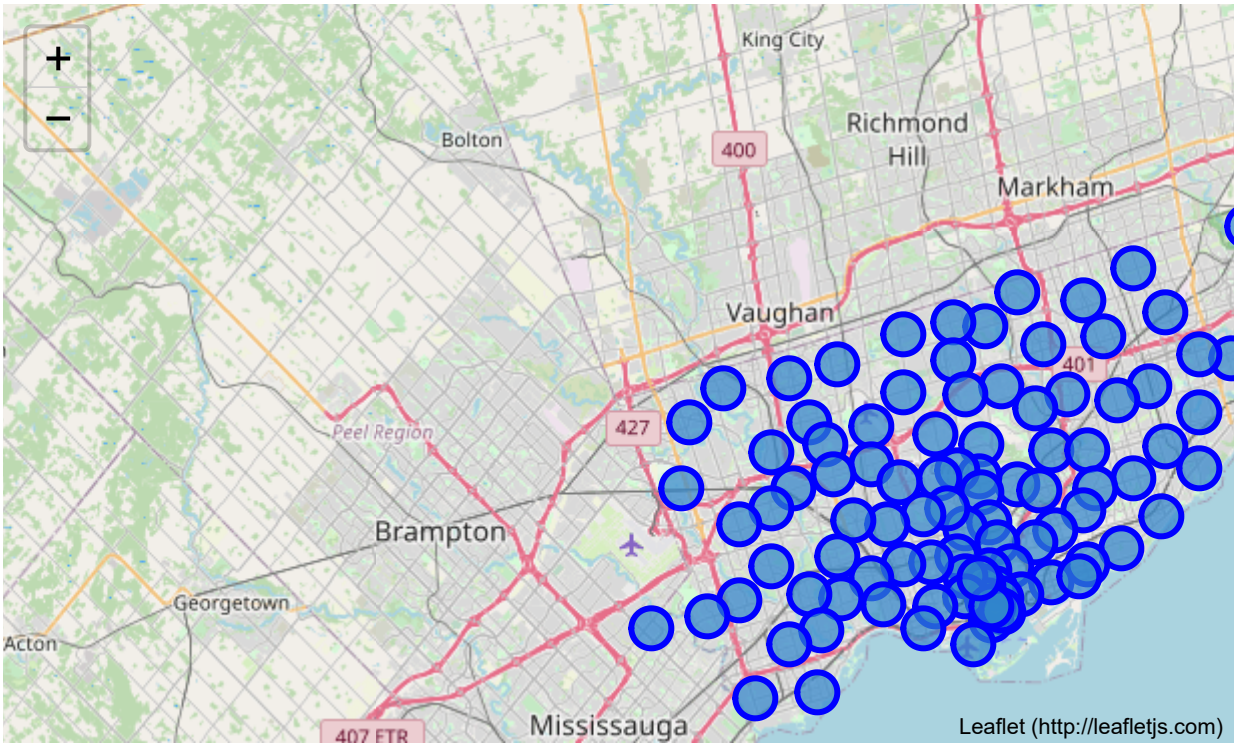
```
The geograpical coordinate of Toronto are 43.6534817, -79.3839347.
```

In [103]:
```
# create map of Toronto using latitude and longitude values
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, postalcode, borough, Neighbourhood in zip(toronto_df['Latitude'], t
    label = '{}, {}, {}'.format(postalcode, borough, Neighbourhood)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=10,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_toronto)

map_toronto
```

Out[103]:



Define Foursquare Credentials and Version

In [104]:
```
CLIENT_ID = 'Y4AOTAYDY23ILH3DR4XZZDYEZDHNECVYXORDA4N4WWLLJ3NB' # your Foursquare 
CLIENT_SECRET = 'XSI4HLFPMXC1PFUWGYR1U35DVKVOO32KWR3FQGGIJINNPEG2' # your Foursqu
VERSION = '20200804' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: Y4AOTAYDY23ILH3DR4XZZDYEZDHNECVYXORDA4N4WWLLJ3NB
CLIENT_SECRET:XSI4HLFPMXC1PFUWGYR1U35DVKVOO32KWR3FQGGIJINNPEG2
```

In [105]:
```
toronto_df.loc[0, 'Neighbourhood']
```

Out[105]:
```
'Malvern, Rouge'
```

In [106]:
```
neighborhood_latitude  = toronto_df.loc[0, 'Latitude'] # neighborhood latitude va
neighborhood_longitude = toronto_df.loc[0, 'Longitude'] # neighborhood longitude

neighborhood_name = toronto_df.loc[0, 'Neighbourhood'] # neighborhood name

print('Latitude and longitude values of {} are {}, {}.'.format(neighborhood_name,
                                                               neighborhood_latit
                                                               neighborhood_longi
```

```
Latitude and longitude values of Malvern, Rouge are 43.806686299999996, -79.194
35340000001.
```

Get the top 100 venues that are in Malvern, Rouge within a radius of 1000 meters. Create the GET request URL. Name your URL url.

In [107]:
```
LIMIT = 100 # max venues to retrieve
radius = 1000 # radius from Malvern, Rouge coordinates

# create URL and name ir "url" using the method "explore"
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    neighborhood_latitude,
    neighborhood_longitude,
    radius,
    LIMIT)
#url
url # display URL
```

Out[107]:
```
'https://api.foursquare.com/v2/venues/explore?&client_id=Y4AOTAYDY23ILH3DR4XZZD
YEZDHNECVYXORDA4N4WWLLJ3NB&client_secret=XSI4HLFPMXC1PFUWGYR1U35DVKVOO32KWR3FQG
GIJINNPEG2&v=20200804&ll=43.806686299999996,-79.19435340000001&radius=1000&limi
t=100'
```

In [108]:
```
results = requests.get(url).json()
```

From the Foursquare lab in the previous module, we know that all the information is in the items key. Before we proceed, let's borrow the get_category_type function from the Foursquare lab.

In [109]:
```
# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

Now we are ready to clean the json and structure it into a pandas dataframe.

In [110]:
```
venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venu
nearby_venues =nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues.head()
```

Out[110]:

	name	categories	lat	lng
0	Images Salon & Spa	Spa	43.802283	-79.198565
1	Harvey's	Restaurant	43.800020	-79.198307
2	Wendy's	Fast Food Restaurant	43.802008	-79.198080
3	Staples Morningside	Paper / Office Supplies Store	43.800285	-79.196607
4	Wendy's	Fast Food Restaurant	43.807448	-79.199056

In [111]:
```
print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))
```

```
19 venues were returned by Foursquare.
```

Explore Neighborhoods in Toronto

Let's create a function to repeat the same process to all the neighborhoods in Toronto¶

In [112]:
```
def getNearbyVenues(names, latitudes, longitudes, radius=500):

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

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.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 venues_list for item in
    nearby_venues.columns = ['Neighbourhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

Now write the code to run the above function on each neighborhood and create a new dataframe called manhattan_venues.

In [113]:
```
toronto_venues = getNearbyVenues(names=toronto_df['Neighbourhood'],
                                 latitudes=toronto_df['Latitude'],
                                 longitudes=toronto_df['Longitude']
                                )
```

```
Malvern, Rouge
Rouge Hill, Port Union, Highland Creek
Guildwood, Morningside, West Hill
Woburn
Cedarbrae
Scarborough Village
Kennedy Park, Ionview, East Birchmount Park
Golden Mile, Clairlea, Oakridge
Cliffside, Cliffcrest, Scarborough Village West
Birch Cliff, Cliffside West
Dorset Park, Wexford Heights, Scarborough Town Centre
Wexford, Maryvale
Agincourt
Clarks Corners, Tam O'Shanter, Sullivan
Milliken, Agincourt North, Steeles East, L'Amoreaux East
Steeles West, L'Amoreaux West
Upper Rouge
Hillcrest Village
Fairview, Henry Farm, Oriole
Bayview Village
York Mills, Silver Hills
Willowdale, Newtonbrook
Willowdale, Willowdale East
York Mills West
Willowdale, Willowdale West
Parkwoods
Don Mills
Don Mills
Bathurst Manor, Wilson Heights, Downsview North
Northwood Park, York University
Downsview
Downsview
Downsview
Downsview
Victoria Village
Parkview Hill, Woodbine Gardens
Woodbine Heights
The Beaches
Leaside
Thorncliffe Park
East Toronto, Broadview North (Old East York)
The Danforth West, Riverdale
India Bazaar, The Beaches West
Studio District
Lawrence Park
Davisville North
North Toronto West, Lawrence Park
Davisville
Moore Park, Summerhill East
Summerhill West, Rathnelly, South Hill, Forest Hill SE, Deer Park
Rosedale
```

```
St. James Town, Cabbagetown
Church and Wellesley
Regent Park, Harbourfront
Garden District, Ryerson
St. James Town
Berczy Park
Central Bay Street
Richmond, Adelaide, King
Harbourfront East, Union Station, Toronto Islands
Toronto Dominion Centre, Design Exchange
Commerce Court, Victoria Hotel
Bedford Park, Lawrence Manor East
Roselawn
Forest Hill North & West, Forest Hill Road Park
The Annex, North Midtown, Yorkville
University of Toronto, Harbord
Kensington Market, Chinatown, Grange Park
CN Tower, King and Spadina, Railway Lands, Harbourfront West, Bathurst Quay,
South Niagara, Island airport
Stn A PO Boxes
First Canadian Place, Underground city
Lawrence Manor, Lawrence Heights
Glencairn
Humewood-Cedarvale
Caledonia-Fairbanks
Christie
Dufferin, Dovercourt Village
Little Portugal, Trinity
Brockton, Parkdale Village, Exhibition Place
North Park, Maple Leaf Park, Upwood Park
Del Ray, Mount Dennis, Keelsdale and Silverthorn
Runnymede, The Junction North
High Park, The Junction South
Parkdale, Roncesvalles
Runnymede, Swansea
Queen's Park, Ontario Provincial Government
Canada Post Gateway Processing Centre
Business reply mail Processing Centre, South Central Letter Processing Plant
Toronto
New Toronto, Mimico South, Humber Bay Shores
Alderwood, Long Branch
The Kingsway, Montgomery Road, Old Mill North
Old Mill South, King's Mill Park, Sunnylea, Humber Bay, Mimico NE, The Queen
sway East, Royal York South East, Kingsway Park South East
Mimico NW, The Queensway West, South of Bloor, Kingsway Park South West, Roy
al York South West
Islington Avenue, Humber Valley Village
West Deane Park, Princess Gardens, Martin Grove, Islington, Cloverdale
Eringate, Bloordale Gardens, Old Burnhamthorpe, Markland Wood
Humber Summit
Humberlea, Emery
Weston
Westmount
Kingsview Village, St. Phillips, Martin Grove Gardens, Richview Gardens
South Steeles, Silverstone, Humbergate, Jamestown, Mount Olive, Beaumond Hei
ghts, Thistletown, Albion Gardens
Northwest, West Humber - Clairville
```

Let's check the size of the resulting dataframe

In [114]:
```
print(toronto_venues.shape)
toronto_venues.head()
```

```
(2131, 7)
```

Out[114]:

	Neighbourhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Malvern, Rouge	43.806686	-79.194353	Wendy's	43.807448	-79.199056	Fast Food Restaurant
1	Rouge Hill, Port Union, Highland Creek	43.784535	-79.160497	Royal Canadian Legion	43.782533	-79.163085	Bar
2	Rouge Hill, Port Union, Highland Creek	43.784535	-79.160497	Affordable Toronto Movers	43.787919	-79.162977	Moving Target
3	Guildwood, Morningside, West Hill	43.763573	-79.188711	RBC Royal Bank	43.766790	-79.191151	Bank
4	Guildwood, Morningside, West Hill	43.763573	-79.188711	G & G Electronics	43.765309	-79.191537	Electronics Store

In [115]:
```
toronto_venues.groupby('Neighbourhood').count()
```

New Toronto, Mimico South, Humber Bay Shores	13	13	13	13	13	13
North Park, Maple Leaf Park, Upwood Park	5	5	5	5	5	5
North Toronto West, Lawrence Park	19	19	19	19	19	19
Northwest, West Humber - Clairville	4	4	4	4	4	4
Northwood Park, York University	7	7	7	7	7	7
Old Mill South, King's Mill Park, Sunnylea, Humber Bay, Mimico NE, The Queensway East, Royal	1	1	1	1	1	1

Let's find out how many unique categories can be curated from all the returned venues

In [116]:
```
print('There are {} uniques categories.'.format(len(toronto_venues['Venue Categor
```

```
There are 269 uniques categories.
```

Analyze Each Neighborhood

In [117]:
```
# Drop rows where 'Venue Category' == 'Neighborhood'
indexNamesVC = toronto_venues[toronto_venues['Venue Category'] == 'Neighbourhood'
toronto_venues.drop(indexNamesVC, inplace=True)
toronto_venues = toronto_venues.reset_index()
del toronto_venues['index']
```

In [118]:
```
# one hot encoding
toronto_onehot = pd.get_dummies(toronto_venues[['Venue Category']], prefix="", pr

# add neighborhood column back to dataframe
toronto_onehot['Neighbourhood'] = toronto_venues['Neighbourhood']

# move neighborhood column to the first column
fixed_columns = [toronto_onehot.columns[-1]] + list(toronto_onehot.columns[:-1])
toronto_onehot = toronto_onehot[fixed_columns]

toronto_onehot.head()
```

Out[118]:

	Neighbourhood	Accessories Store	Afghan Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	R
0	Malvern, Rouge	0	0	0	0	0	0	0	0	
1	Rouge Hill, Port Union, Highland Creek	0	0	0	0	0	0	0	0	
2	Rouge Hill, Port Union, Highland Creek	0	0	0	0	0	0	0	0	
3	Guildwood, Morningside, West Hill	0	0	0	0	0	0	0	0	
4	Guildwood, Morningside, West Hill	0	0	0	0	0	0	0	0	

In [119]:
```
toronto_onehot.shape
```

Out[119]: (2131, 270)

Next, let's group rows by neighborhood and by taking the mean of the frequency of occurrence of each category

In [120]:
```
toronto_grouped = toronto_onehot.groupby('Neighbourhood').mean().reset_index()
toronto_grouped
```

3	Bayview Village	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	Bedford Park, Lawrence Manor East	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	Berczy Park	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
6	Birch Cliff, Cliffside West	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7	Brockton, Parkdale Village, Exhibition Place	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	Business reply mail Processing Centre, South C...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
9	CN Tower, King and Spadina, Railway Lands, Har	0.000000	0.000000	0.058824	0.058824	0.058824	0.117647	0.117647

In [121]:
```
toronto_grouped.shape
```

Out[121]: (96, 270)

Let's print each neighborhood along with the top 5 most common venues

In [122]:
```
num_top_venues = 5

for hood in toronto_grouped['Neighbourhood']:
    print("----"+hood+"----")
    temp = toronto_grouped[toronto_grouped['Neighbourhood'] == hood].T.reset_inde
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(n
    print('\n')
```

```
                      venue  freq
0               Coffee Shop  0.15
1                     Hotel  0.15
2              Intersection  0.08
3  Middle Eastern Restaurant  0.08
4              Burrito Place  0.08


----Cedarbrae----
                  venue  freq
0     Hakka Restaurant  0.11
1    Athletics & Sports  0.11
2  Caribbean Restaurant  0.11
3   Fried Chicken Joint  0.11
4                  Bank  0.11


----Central Bay Street----
                venue  freq
0         Coffee Shop  0.18
1                Café  0.05
```

Let's put that into a pandas dataframe

First, let's write a function to sort the venues in descending order.

In [123]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

Now let's create the new dataframe and display the top 10 venues for each neighborhood.

In [124]:
```
num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighbourhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighbourhood'] = toronto_grouped['Neighbourhood']

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

neighborhoods_venues_sorted.head()
```

Out[124]:

	Neighbourhood	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	
0	Agincourt	Lounge	Latin American Restaurant	Skating Rink	Breakfast Spot	Donut Shop	Diner	Discount Store	D
1	Alderwood, Long Branch	Pizza Place	Coffee Shop	Skating Rink	Pharmacy	Pub	Sandwich Place	Gym	
2	Bathurst Manor, Wilson Heights, Downsview North	Coffee Shop	Bank	Mobile Phone Shop	Chinese Restaurant	Bridal Shop	Sandwich Place	Diner	F
3	Bayview Village	Café	Bank	Chinese Restaurant	Japanese Restaurant	Yoga Studio	Diner	Discount Store	D
4	Bedford Park, Lawrence Manor East	Restaurant	Coffee Shop	Sandwich Place	Italian Restaurant	Juice Bar	Pharmacy	Thai Restaurant	

In [ ]:
```
```