NYC and Toronto Venue Comparison

IBM Coursera Capstone Project

By: Jimmy Amundson

Introduction

Business Problem:

- What new venues in Toronto or New York City would be best to invest in?

- **Audience**: Investor/Entrepreneur

Solution:

1. Use Foursquare to locate every venue in Toronto and NYC

2. Compare the presence of venue categories in Toronto and NYC to each other.

3. Interpret the comparison data and decide what venues could be added to each city.

Collecting Venues with Foursquare

- Use the NYC and Toronto Latitude and Longitude to call all nearby venues within a 500 meter radius

```
#this call searches all the venues in the neighborhoods of New York City
Limit = 100
radius = 500

url_ny = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    ny_latitude,
    ny_longitude,
    radius,
    Limit)

url_ny
```

- The client ID and secret are unique to my foursquare account and allow the user to make calls to Foursquare.
- The url_ny gives extensive information on each venue in a json file.

JSON →Pandas Dataframe

- From the JSON file returned by the Foursquare URL, the venue categories were extracted and put into a Pandas dataframe using the code below.

```
#breaks json file down from response dictionary ->groups dictionary ->first element in list 'groups' ->items dictionary
venues_ny = results_ny['response']['groups'][0]['items']
nearby_venues_ny = json_normalize(venues_ny) # flatten JSON

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

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

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

```
toronto_venues = getNearbyVenues(names=new_cdf['Neighborhood'],
                                 latitudes=new_cdf['Latitude'],
                                 longitudes=new_cdf['Longitude']
                                )
```

The User Functions created to tabulate the venue categories into a pandas dataframe:

get_category_type

```
# 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']
```

getNearbyVenues

```
#the getNearbyVenues function collects the venues from each neighborhood in the city

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_secret={}&v={}&ll={},{}&radius={}&limit={}'.format
            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])
    #arrange venues_list into pandas dataframe
    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']
    return(nearby_venues)
```

Total Venue Comparison

Toronto

- 2147 Venues
- 272 Venue Categories

New York City

- 10047 Venues
- 430 Venue Categories

Detailed Analysis

- The venues returned for each neighborhood were mapped and analyzed using 3 different methods.
  - Top 5 venue frequency in each neighborhood
  - Neighborhood mapping with folium
  - Kmeans clustering
- This information could be useful to the investor after examining the main comparison dataframe.
- This is good for digging deeper, but not for answering the main business problem.

Merging the venue dataframes

- A counter function was imported to count the occurrences of each venue category in each city

```
toronto_venues_list = Counter(toronto_venues['Venue Category'])
tvl=list(toronto_venues_list.most_common())
tvl;
```

- The dataframes of each city were combined using the pandas attribute: .merge()

	Venue	NY_total	Toronto_total
0	Pizza Place	439.0	48.0
1	Coffee Shop	318.0	185.0
2	Italian Restaurant	311.0	41.0
3	Deli / Bodega	267.0	12.0
4	Bakery	228.0	38.0

Venue adjustment

- Because NYC is way bigger than Toronto a straight across comparison would be misleading to the investor, therefore an adjustment was made.
- Two adjustment methods were used.
  1) Population Adjustment:

$$\frac{\text{NYC annual visitors + NYC population}}{\text{TOR annual visitors + TOR population}} = 2.478$$

  2) Venues Returned Ratio:

$$\frac{\text{Total NYC venues}}{\text{Total TOR venues}} = 4.745$$

- The NY_total column was divided by these adjustment ratios for realistic comparison

Final Comparison Dataframe

```
TORONTO vs NEW YORK
     COMPARISON

  Venue Presence:       Color:
0          NYC >>>         RED
1           NYC >      ORANGE
2         Similar       WHITE
3       Toronto >  LIGHT BLUE
4     Toronto >>>        BLUE
```

	Venue	NY_total	Toronto_total	NY_adjusted
0	Pizza Place	439	48	177.2
1	Coffee Shop	318	185	128.3
2	Italian Restaurant	311	41	125.5
3	Deli / Bodega	267	12	107.7
4	Bakery	228	38	92
5	Bar	222	33	89.6
6	Chinese Restaurant	211	17	85.1
7	Grocery Store	204	26	82.3
8	Sandwich Place	184	43	74.3
9	Park	178	53	71.8
10	Mexican Restaurant	178	14	71.8
11	Pharmacy	175	23	70.6

Population Adjustment Venue Comparison Dataframe

- A user function was created to highlight the cells for easy visualization.
  - The legend describes the color coordination
- New York City is highlighted significantly more than Toronto, meaning this adjustment ratio might be an inaccurate venue comparison method.

TORONTO vs NEW YORK
COMPARISON

	Venue Presence:	Color:
0	NYC >>>	RED
1	NYC >	ORANGE
2	Similar	WHITE
3	Toronto >	LIGHT BLUE
4	Toronto >>>	BLUE

	Venue	NY_total	Toronto_total	NY_adjusted
0	Pizza Place	439	48	177.2
1	Coffee Shop	318	185	128.3
2	Italian Restaurant	311	41	125.5
3	Deli / Bodega	267	12	107.7
4	Bakery	228	38	92
5	Bar	222	33	89.6
6	Chinese Restaurant	211	17	85.1
7	Grocery Store	204	26	82.3
8	Sandwich Place	184	43	74.3
9	Park	178	53	71.8
10	Mexican Restaurant	178	14	71.8
11	Pharmacy	175	23	70.6

NYC adjustment

- The venues returned ratio equalizes the rows highlighted for Toronto and NYC, but does not factor in city population into the adjustment.
- The investor is able to compare the venues of the two cities with either adjustment ratio. Although, I believe the population adjustment ratio provides the investor with more information.
  - The entire venue comparison table can be viewed from the notebook at: https://github.com/jimboslice444/IBM_DataCourse-Capstone/blob/master/NYC%20vs.%20Toronto%20Venue%20Comparison%20analysis.ipynb

Conclusion

- Further analysis could be done in order to pinpoint exactly which venue categories should be invested in by the investor, rather than highlighting which venues types have potential in each respective city. By comparing other cities venues to NYC and Toronto, the investor could be more confident in their decision as well.
- Based on my data analysis I recommend investing in an aquarium in NYC and an international restaurant in Toronto.