Introduction

Background:
A large technology company based in New York City is relocating its office and all of its employees to Toronto, Canada.

Problem:
All of the employees need to find a new place to live within 30 days. They don't have the time to talk to locals, research on their own, or tour with realtors before moving day. They need help to make a confident, data-driven decision that takes their current neighborhood, lifestyle, and interests into account.

Target Audience:
Individual employees at the relocating company or anyone moving from NYC to Toronto. Individuals care about this comparative analysis because it will save them time and make their new neighborhood feel more like home.

Why Stakeholders Care:
The stakeholders at the large technology company care because their profitability (and annual bonus) is greatly impacted by a high employee turnover rate. If employees were to quickly move into a neighborhood that wasn't a good fit, they may not stay in Toronto very long, and ultimately impact their bottom line. They also care about cultivating a positive corporate culture that will attract new talent, so it's critical that their employees are happy and well-adjusted to the new city.

Data

Summary:
To solve this problem, I will primarily use Foursquare venue data, lists of neighborhoods within Toronto/NYC, and the associated latitude/longitude coordinates of those neighborhoods.

After cleaning and wrangling the data into a two separate pandas dataframes for NYC and Toronto, I can use one-hot encoding and list the frequency of venue types within each neighborhood. Then I will use a similar approach to a recommender system to match up the neighborhoods in NYC that most closely resemble the neighborhoods in Toronto.

The results will inform the individual employees which neighborhoods in Toronto that are most likely to feel like home.
.

Data Sources with Examples:

1. List of neighborhoods in Toronto. For Toronto, the neighborhood name data will be scraped from a Wikipedia page here using the pd.read_csv function.: https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M .

   Example of the Toronto wikipedia data before data cleanup (see #1 above)

	Postal Code	Borough	Neighborhood
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

2. List of Toronto postal codes and associated latitude/longitude coordinates. This data is available from a hosted .csv file located here: http://cocl.us/Geospatial_data
   Example of Toronto coordinate data.

	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

3. Merged Toronto dataframe. For comparison to the NYC datatable, the two Toronto datatables (shown in #1 and #2 above) must be merged into one datatable like the example shown below. .

	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

4. List of neighborhoods and associated latitude/longitude coordinates in NYC is available from a large json file located here: https://geo.nyu.edu/catalog/nyu_2451_34572

```
newyork_data
```


```
{'type': 'FeatureCollection',
 'totalFeatures': 306,
 'features': [{'type': 'Feature',
   'id': 'nyu_2451_34572.1',
   'geometry': {'type': 'Point',
    'coordinates': [-73.84720052054902, 40.89470517661]},
   'geometry_name': 'geom',
   'properties': {'name': 'Wakefield',
    'stacked': 1,
    'annoline1': 'Wakefield',
    'annoline2': None,
    'annoline3': None,
    'annoangle': 0.0,
    'borough': 'Bronx',
    'bbox': [-73.84720052054902,
     40.89470517661,
     -73.84720052054902,
     40.89470517661]}},
  {'type': 'Feature',
   'id': 'nyu_2451_34572.2',
   'geometry': {'type': 'Point',
    'coordinates': [-73.82993910812398, 40.87429419303012]},
   'geometry_name': 'geom',
   'properties': {'name': 'Co-op City',
    'stacked': 2,
    'annoline1': 'Co-op',
    'annoline2': 'City',
    'annoline3': None,
    'annoangle': 0.0,
    'borough': 'Bronx',
    'bbox': [-73.82993910812398,
     40.87429419303012,
     -73.82993910812398,
     40.87429419303012]}},
```

The json file is basically a list of nested Python dictionaries. Before using this data to compare with the Toronto data, we'll need to convert it to a Pandas dataframe. We can use a for loop and the append method to fill an empty dataframe with the proper headings. The resulting dataframe will look like this:

```
neighborhoods.head()
```

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

5. Venue data from Foursquare Developer API explore calls.

Foursquare data explanation:
Foursquare is a social media platform that provides highly accurate data about 105MM places within 90 countries. The Foursquare Developer API (https://developer.foursquare.com/docs/places-api/) allows anyone with a free account to make calls to their global places database. My analysis will focus on the explore function to extract venue location and venue type. There are approximately 234 different venue categories such as "Yoga Studio," "Wine Bar," and "American Restaurant."

The Foursquare API "explore" call will retrieve all the venue information requested around a specific point.

```
LIMIT = 100

radius = 500

url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    neighborhood_latitude,
    neighborhood_longitude,
    radius,
    LIMIT)
url
```

From there I will use one-hot encoding in order to compare the relative frequency of venue types within each neighborhood. Then we will be able to find and compare similar neighborhoods between the two city's dataframes.
Example of the venue-type frequency by Neighborhood output:

```
----Berczy Park----
          venue  freq
0   Coffee Shop  0.09
1  Cocktail Bar  0.05
2    Restaurant  0.03
3          Café  0.03
4   Cheese Shop  0.03


----Brockton, Parkdale Village, Exhibition Place----
                  venue  freq
0                 Café  0.12
1  Performing Arts Venue  0.08
2           Coffee Shop  0.08
3         Breakfast Spot  0.08
4            Yoga Studio  0.04
```

In summary, the final report will show which Toronto neighborhoods are most similar to NYC neighborhoods by comparing their frequency of venue types. For example, a neighborhood in NYC with a large percentage of nightclubs and bars would likely have a similar feel to a Toronto neighborhood with a similar percentage of nightclubs and bars.

This analysis will be important to the individual employees because they will learn exactly which neighborhood is most likely to fit their lifestyle. This, in turn, will help the company stakeholders keep turnover low and stay profitable during the big move.