Finding similar neighborhoods in New York and Toronto cities

Nikhil Thomas
Aug 6, 2021

Table of Contents

Introduction

As people move from New York City to Toronto and vice versa, many are worried about the new changes they would bring in their lives. They are concerned about their neighborhoods and wish to move to a similar area they live in. It would be beneficial if they could know the neighborhood that matches their current one. This would ease their nervousness and help them properly focus on the actual neighborhood of their liking rather than regretting the choice after the move. Therefore, we would try to identify and compare each area in New York and Toronto and create a similar grouping to help people identify their similar neighborhoods in the other city.

We would plan to capture all the neighborhoods in New York and Toronto cities. Using the Foursquare integration, we could then find all types of places surrounding these neighborhoods. Break them into religious, restaurants, schools/colleges, parks, movie theaters, etc. Based on this information, we could then utilize K-means clustering to similar group neighborhoods together.

Data

For this project, we have used data from public sources and Foursquare data through its API interface.

Public Data:

The New York data about their neighborhoods would be downloaded from the NYC Planning website. The data would list all the neighborhoods and the geographic data.

Link: https://www1.nyc.gov/site/planning/data-maps/open-data.page

The Toronto zip codes, along with their neighborhood names, can be downloaded from the wiki page.

Link: https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M

Foursquare API:

Foursquare provides valuable and publicly accessible location information like the amenities in nearby locations. We use their developer tools to access the required information about the neighborhoods in a city. Using this accessed information, we then rank the neighborhoods based on the amenities they have. This service is free of charge.

Methodology

The First Step of the data analysis is to retrieve information about all the boroughs and their neighborhoods for both cities (New York and Toronto). This information is captured from public data sources. Then we would pass the coordinates of neighborhoods in New York and Toronto to the Foursquare and retrieve the lead all the venues in about the 500-mile radius. The venue information would include its name, location coordinates, and the category of the venue.

Once we have all the information, we would merge the data into a single data frame. We still would retain all the information about the location's parents. We would identify all the top 10 venues in each neighborhood based on each venue's frequency. i.e., by placing the frequency of venue category in each location. This would help us identify typical venues like parks, types of restaurants, types of business, etc. This would allow us to understand what the people in each location like and provide a location's character. So, if we see any areas having many parks, we could assume that the people in this location like to go to parks and hence would tend to be more connected with nature.

Once we have identified the top venues in each location, we will use the K-Means Clustering algorithm to group venues in neighborhoods. Then we can reduce the number of individual venue comparisons to be done against each neighborhood. We will make these comparisons against the types of venue, individually, collectively, or all together. The reason why er choose the K-means clustering is that it's easier to define and use.

We can examine each cluster and determine the distinguishing characteristics on venue categories that distinguish each cluster. This will help us provide an alternate if anyone wants to find a similar neighborhood in another city. This should also help if anyone is planning to move within the city.

Results

Understanding the New York Layout

Exploring the New York Data

```
with open('newyork_data.json') as json_data:
    newyork_data = json.load(json_data)

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]}},
  {'type': 'Feature',
   'id': 'nyu_2451_34572.3',
   'geometry': {'type': 'Point',
    'coordinates': [-73.82780644716412, 40.887555677350775]},
   'geometry_name': 'geom',
   'properties': {'name': 'Eastchester',
    'stacked': 1,
    'annoline1': 'Eastchester',
    'annoline2': None,
    'annoline3': None,
    'annoangle': 0.0,
    'borough': 'Bronx',
    'bbox': [-73.82780644716412,
     40.887555677350775,
     -73.82780644716412,
     40.887555677350775]}},
```

Transform the data into a dataframe

```
# Extracting features from the data downloaded
neighborhoods_data = newyork_data['features']

# define the dataframe columns
column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']

# instantiate the dataframe
neighborhoods = pd.DataFrame(columns=column_names)

for data in neighborhoods_data:
    borough = neighborhood_name = data['properties']['borough']
    neighborhood_name = data['properties']['name']

    neighborhood_latlon = data['geometry']['coordinates']
    neighborhood_lat = neighborhood_latlon[1]
    neighborhood_lon = neighborhood_latlon[0]

    neighborhoods = neighborhoods.append({'Borough': borough,
                                          'Neighborhood': neighborhood_name,
                                          'Latitude': neighborhood_lat,
                                          'Longitude': neighborhood_lon}, ignore_index=True)
neighborhoods.head()
```

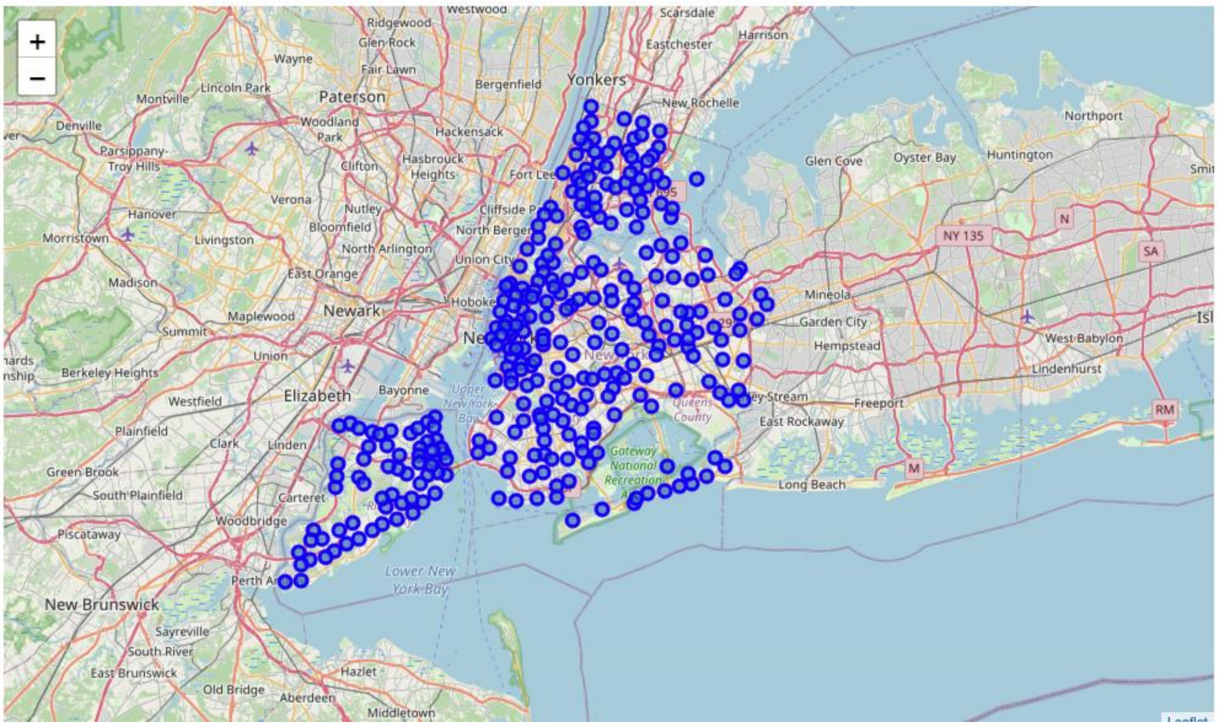
	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806
3	Bronx	Fieldston	40.895437	-73.905643
4	Bronx	Riverdale	40.890834	-73.912585

```python
# Creating a Map of New York for identifying the neighbourhoods

# create map of New York using latitude and longitude values
map_newyork = folium.Map(location=[ny_latitude, ny_longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(neighborhoods['Latitude'], neighborhoods['Longitude'], neighborhoods['Borou
gh'], neighborhoods['Neighborhood']):
    label = '{}, {}'.format(neighborhood, borough)
    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_newyork)

map_newyork
```



Getting the venue locations from Foursquare for New York

```
ny_venues = getNearbyVenues(names=neighborhoods['Neighborhood'],
                                   borough=neighborhoods['Borough'],
                                   latitudes=neighborhoods['Latitude'],
                                   longitudes=neighborhoods['Longitude']
                                  )
```

**Checking the size of the data returned

```
print(ny_venues.shape)

ny_venues['City'] ='New York'

ny_venues.head()
```

```
(10199, 8)
```

	Neighborhood	Neighborhood Borough	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	City
0	Wakefield	Bronx	40.894705	-73.847201	Lollipops Gelato	40.894123	-73.845892	Dessert Shop	New York
1	Wakefield	Bronx	40.894705	-73.847201	Rite Aid	40.896649	-73.844846	Pharmacy	New York
2	Wakefield	Bronx	40.894705	-73.847201	Carvel Ice Cream	40.890487	-73.848568	Ice Cream Shop	New York
3	Wakefield	Bronx	40.894705	-73.847201	Walgreens	40.896528	-73.844700	Pharmacy	New York
4	Wakefield	Bronx	40.894705	-73.847201	Dunkin'	40.890459	-73.849089	Donut Shop	New York

Understanding the Toronto Layout

Understanding the Toronto Data

```
wiki = requests.get("https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M").text

soup = BeautifulSoup(wiki,"lxml")

table_contents=[]
table=soup.find('table')
for row in table.findAll('td'):
    cell = {}
    if row.span.text=='Not assigned':
        pass
    else:
        cell['PostalCode'] = row.p.text[:3]
        cell['Borough'] = (row.span.text).split('(')[0]
        cell['Neighborhood'] = (((((row.span.text).split('(')[1]).strip(')')).replace(' /',',')).replace(')',' ')).str
ip(' ')
        table_contents.append(cell)

# print(table_contents)
df_Tor=pd.DataFrame(table_contents)
df_Tor['Borough']=df_Tor['Borough'].replace({'Downtown TorontoStn A PO Boxes25 The Esplanade':'Downtown Toronto Stn
A',
                                             'East TorontoBusiness reply mail Processing Centre969 Eastern':'East Toro
nto Business',
                                             'EtobicokeNorthwest':'Etobicoke Northwest','East YorkEast Toronto':'East
York/East Toronto',
                                             'MississaugaCanada Post Gateway Processing Centre':'Mississauga'})

df_Tor.head()
```

	PostalCode	Borough	Neighborhood
0	M3A	North York	Parkwoods
1	M4A	North York	Victoria Village
2	M5A	Downtown Toronto	Regent Park, Harbourfront
3	M6A	North York	Lawrence Manor, Lawrence Heights
4	M7A	Queen's Park	Ontario Provincial Government

Visualizing Toronto and its neighbourhoods

```
## Retriving the geocordinates of Toronto City

address = 'Toronto, ON'

geolocator = Nominatim(user_agent="explorer")
location = geolocator.geocode(address)
tor_latitude = location.latitude
tor_longitude = location.longitude
print('The geographical coordinate of Toronto City are {}, {}.'.format(tor_latitude, tor_longitude))
```

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

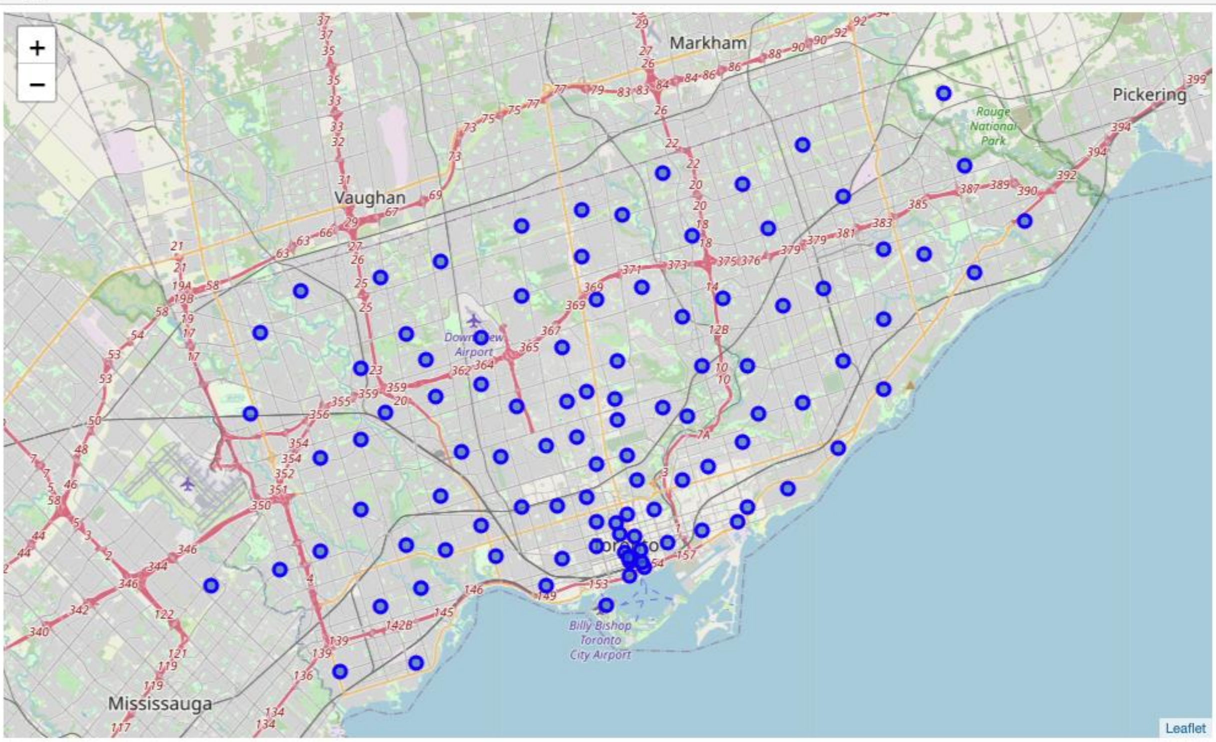
**Plotting the cordinates into the map

```
# Creating a Map of Toronto for identifying the neighbourhoods

# create map of Toronto using latitude and longitude values
map_Toronto = folium.Map(location=[tor_latitude, tor_longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(df_Tor['Latitude'], df_Tor['Longitude'], df_Tor['Borough'], df_Tor['Neighbo
rhood']):
    label = '{}, {}'.format(neighborhood, borough)
    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
```



Getting the venue locations from Foursquare for Toronto

Retrieve the venues from Foursquare for Toronto

```
tor_venues = getNearbyVenues(names=df_Tor['Neighborhood'],
                                   borough=df_Tor['Borough'],
                                   latitudes=df_Tor['Latitude'],
                                   longitudes=df_Tor['Longitude']
                                  )
tor_venues.shape
```

```
(2149, 8)
```

**Adding City to identify the city in the merged data.

```
print(tor_venues.shape)

tor_venues['City'] ='Toronto'

tor_venues.head(20)
```

```
(2149, 8)
```

	Neighborhood	Neighborhood Borough	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	City
0	Parkwoods	North York	43.753259	-79.329656	Brookbanks Park	43.751976	-79.332140	Park	Toronto
1	Parkwoods	North York	43.753259	-79.329656	KFC	43.754387	-79.333021	Fast Food Restaurant	Toronto
2	Parkwoods	North York	43.753259	-79.329656	Variety Store	43.751974	-79.333114	Food & Drink Shop	Toronto
3	Victoria Village	North York	43.725882	-79.315572	Victoria Village Arena	43.723481	-79.315635	Hockey Arena	Toronto
4	Victoria Village	North York	43.725882	-79.315572	Tim Hortons	43.725517	-79.313103	Coffee Shop	Toronto
5	Victoria Village	North York	43.725882	-79.315572	Portugril	43.725819	-79.312785	Portuguese Restaurant	Toronto
6	Victoria Village	North York	43.725882	-79.315572	Eglinton Ave E & Sloane Ave/Bermondsey Rd	43.726086	-79.313620	Intersection	Toronto
7	Victoria Village	North York	43.725882	-79.315572	Pizza Nova	43.725824	-79.312860	Pizza Place	Toronto
8	Regent Park, Harbourfront	Downtown Toronto	43.654260	-79.360636	Roselle Desserts	43.653447	-79.362017	Bakery	Toronto
9	Regent Park, Harbourfront	Downtown Toronto	43.654260	-79.360636	Tandem Coffee	43.653559	-79.361809	Coffee Shop	Toronto
10	Regent Park, Harbourfront	Downtown Toronto	43.654260	-79.360636	Cooper Koo Family YMCA	43.653249	-79.358008	Distribution Center	Toronto
11	Regent Park, Harbourfront	Downtown Toronto	43.654260	-79.360636	Impact Kitchen	43.656369	-79.356980	Restaurant	Toronto
12	Regent Park, Harbourfront	Downtown Toronto	43.654260	-79.360636	Body Blitz Spa East	43.654735	-79.359874	Spa	Toronto
13	Regent Park, Harbourfront	Downtown Toronto	43.654260	-79.360636	Corktown Common	43.655618	-79.356211	Park	Toronto
14	Regent Park, Harbourfront	Downtown Toronto	43.654260	-79.360636	The Extension Room	43.653313	-79.359725	Gym / Fitness Center	Toronto
15	Regent Park, Harbourfront	Downtown Toronto	43.654260	-79.360636	Dominion Pub and Kitchen	43.656919	-79.358967	Pub	Toronto
16	Regent Park, Harbourfront	Downtown Toronto	43.654260	-79.360636	The Distillery Historic District	43.650244	-79.359323	Historic Site	Toronto
17	Regent Park, Harbourfront	Downtown Toronto	43.654260	-79.360636	SOMA chocolatemaker	43.650622	-79.358127	Chocolate Shop	Toronto
18	Regent Park, Harbourfront	Downtown Toronto	43.654260	-79.360636	Sumach Espresso	43.658135	-79.359515	Coffee Shop	Toronto
19	Regent Park, Harbourfront	Downtown Toronto	43.654260	-79.360636	Distillery Sunday Market	43.650075	-79.361832	Farmers Market	Toronto

Merging the New York and Toronto Data

Merging the data from New York and Toronto

```
merge_data = pd.concat([ny_venues,tor_venues])
merge_data.head()
```

	Neighborhood	Neighborhood Borough	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	City
0	Wakefield	Bronx	40.894705	-73.847201	Lollipops Gelato	40.894123	-73.845892	Dessert Shop	New York
1	Wakefield	Bronx	40.894705	-73.847201	Rite Aid	40.896649	-73.844846	Pharmacy	New York
2	Wakefield	Bronx	40.894705	-73.847201	Carvel Ice Cream	40.890487	-73.848568	Ice Cream Shop	New York
3	Wakefield	Bronx	40.894705	-73.847201	Walgreens	40.896528	-73.844700	Pharmacy	New York
4	Wakefield	Bronx	40.894705	-73.847201	Dunkin'	40.890459	-73.849089	Donut Shop	New York

```
merge_data.groupby(['Neighborhood','Neighborhood Borough','City']).count()
```

			Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood	Neighborhood Borough	City						
Agincourt	Scarborough	Toronto	4	4	4	4	4	4
Alderwood, Long Branch	Etobicoke	Toronto	8	8	8	8	8	8
Allerton	Bronx	New York	27	27	27	27	27	27
Annadale	Staten Island	New York	10	10	10	10	10	10
Arden Heights	Staten Island	New York	4	4	4	4	4	4
Arlington	Staten Island	New York	9	9	9	9	9	9
Arrochar	Staten Island	New York	19	19	19	19	19	19
Arverne	Queens	New York	21	21	21	21	21	21
Astoria	Queens	New York	100	100	100	100	100	100
Astoria Heights	Queens	New York	15	15	15	15	15	15
Auburndale	Queens	New York	19	19	19	19	19	19
Bath Beach	Brooklyn	New York	50	50	50	50	50	50
Bathurst Manor, Wilson Heights, Downsview North	North York	Toronto	23	23	23	23	23	23
Battery Park City	Manhattan	New York	92	92	92	92	92	92

Analyzing each neighborhood

Analyze Each Neighborhood

```
# one hot encoding
venues_onehot = pd.get_dummies(merge_data[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
venues_onehot['Neighborhood'] = merge_data['Neighborhood']

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

venues_onehot.head()
```

	Yoga Studio	ATM	Accessories Store	Acupuncturist	Adult Boutique	Afghan Restaurant	African Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	An
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```
venues_onehot.shape
```

```
(12348, 473)
```

```
venues_grouped = venues_onehot.groupby('Neighborhood').mean().reset_index()
venues_grouped.head()
```

	Neighborhood	Yoga Studio	ATM	Accessories Store	Acupuncturist	Adult Boutique	Afghan Restaurant	African Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal
0	Agincourt	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	Alderwood, Long Branch	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	Allerton	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	Annadale	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	Arden Heights	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Identifying the top 10 venues in each neighborhood

Identify the top 10 venues of the neighborhoods

```
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Neighborhood']
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['Neighborhood'] = venues_grouped['Neighborhood']

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

neighborhoods_venues_sorted.head()
```

	Neighborhood	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	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Agincourt	Lounge	Latin American Restaurant	Skating Rink	Breakfast Spot	Cultural Center	Cupcake Shop	Empanada Restaurant	English Restaurant	Entertainment Service	Escape Room
1	Alderwood, Long Branch	Pizza Place	Pub	Playground	Sandwich Place	Pharmacy	Coffee Shop	Gym	Filipino Restaurant	Field	Dumpling Restaurant
2	Allerton	Pizza Place	Deli / Bodega	Cosmetics Shop	Chinese Restaurant	Discount Store	Supermarket	Breakfast Spot	Gas Station	Pharmacy	Bus Station
3	Annadale	Cosmetics Shop	Pizza Place	Diner	Pharmacy	Restaurant	American Restaurant	Train Station	Sushi Restaurant	Bar	Liquor Store
4	Arden Heights	Bus Stop	Pharmacy	Coffee Shop	Pizza Place	Falafel Restaurant	Electronics Store	Empanada Restaurant	English Restaurant	Entertainment Service	Escape Room

```
neighborhoods_venues_sorted.shape
```

```
(398, 11)
```

Clustering neighborhoods to find the similarities

```
# set number of clusters
kclusters = 5

venues_grouped_clustering = venues_grouped.drop('Neighborhood', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(venues_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

```
array([4, 0, 0, 4, 2, 2, 0, 4, 4, 0], dtype=int32)
```

Creating a dataframe with complete view based on our analysis

```
# Preping up the data for complete the missing information
final_df = merge_data.drop(['Venue','Venue Latitude','Venue Longitude','Venue Category'], axis=1)
final_df = final_df.drop_duplicates().reset_index(drop=True)
final_df.head()
```

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

Merging the information with complete information to create a

```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

# merge final_df with grouped data to add latitude/longitude for each neighborhood
final_df = final_df.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

final_df.head(10)
```

	Neighborhood	Neighborhood Borough	Neighborhood Latitude	Neighborhood Longitude	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Mo Commo Venu
0	Wakefield	Bronx	40.894705	-73.847201	New York	0	Pharmacy	Sandwich Place	Caribbean Restaurant	Laundromat	Ice Cream Shop	Donut Sho
1	Co-op City	Bronx	40.874294	-73.829939	New York	0	Bus Station	Post Office	Restaurant	Fast Food Restaurant	Market	Bar
2	Eastchester	Bronx	40.887556	-73.827806	New York	0	Caribbean Restaurant	Bus Station	Diner	Deli / Bodega	Seafood Restaurant	Pizza Plac
3	Fieldston	Bronx	40.895437	-73.905643	New York	0	River	Bus Station	Athletics & Sports	Plaza	Yemeni Restaurant	Event Service
4	Riverdale	Bronx	40.890834	-73.912585	New York	3	Park	Bus Station	Medical Supply Store	Gym	Bank	Baseball Field
5	Kingsbridge	Bronx	40.881687	-73.902818	New York	0	Pizza Place	Bar	Sandwich Place	Bakery	Latin American Restaurant	Mexican Restauran
6	Marble Hill	Manhattan	40.876551	-73.910660	New York	4	Sandwich Place	Gym	Coffee Shop	Department Store	Steakhouse	Suppleme Shop
7	Woodlawn	Bronx	40.898273	-73.867315	New York	0	Deli / Bodega	Pub	Pizza Place	Playground	Food Truck	Auto Workshop
8	Norwood	Bronx	40.877224	-73.879391	New York	0	Pizza Place	Park	Chinese Restaurant	Bank	Pharmacy	Restauran
9	Williamsbridge	Bronx	40.881039	-73.857446	New York	4	Soup Place	Nightclub	Bar	Spa	Falafel Restaurant	Electronics Store

Creating an aggregate view of cluster per city and borough

Agregate view on how the clusters are layed out

```
aggr_view = pd.pivot_table(final_df,values = 'Neighborhood', index=['City','Neighborhood Borough'], columns=['Cluster Labels'],aggfunc='count')
aggr_view
```

	Cluster Labels	0	1	2	3	4
City	Neighborhood Borough					
New York	Bronx	46.0	NaN	NaN	2.0	4.0
	Brooklyn	33.0	NaN	NaN	1.0	36.0
	Manhattan	2.0	NaN	NaN	NaN	38.0
	Queens	49.0	NaN	3.0	4.0	25.0
	Staten Island	27.0	NaN	14.0	1.0	20.0
Toronto	Central Toronto	NaN	NaN	NaN	2.0	7.0
	Downtown Toronto	1.0	NaN	NaN	NaN	16.0
	Downtown Toronto Stn A	NaN	NaN	NaN	NaN	1.0
	East Toronto	NaN	NaN	NaN	NaN	4.0
	East Toronto Business	NaN	NaN	NaN	NaN	1.0
	East York	2.0	NaN	NaN	NaN	2.0
	East York/East Toronto	NaN	NaN	NaN	1.0	NaN
	Etobicoke	3.0	1.0	NaN	1.0	4.0
	Etobicoke Northwest	NaN	NaN	NaN	NaN	1.0
	Mississauga	NaN	NaN	NaN	NaN	1.0
	North York	4.0	1.0	1.0	5.0	12.0
	Queen's Park	NaN	NaN	NaN	NaN	1.0
	Scarborough	5.0	NaN	NaN	2.0	9.0
	West Toronto	NaN	NaN	NaN	NaN	6.0
	York	1.0	NaN	NaN	1.0	2.0

Analyzing each cluster

Let's look at Cluster 3 as an example

```
Cluster_3 = final_df[final_df['Cluster Labels']==3]
Cluster_3
```

	Neighborhood	Neighborhood Borough	Neighborhood Latitude	Neighborhood Longitude	City	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
4	Riverdale	Bronx	40.890834	-73.912585	New York	3	Park	Bus Station	Medical Supply Store	Gym	Bank
27	Clason Point	Bronx	40.806551	-73.854144	New York	3	Park	South American Restaurant	Grocery Store	Convenience Store	Bus Stop
91	Bergen Beach	Brooklyn	40.615150	-73.898556	New York	3	Harbor / Marina	Park	Moving Target	Athletics & Sports	Baseball Field
148	South Ozone Park	Queens	40.668550	-73.809865	New York	3	Park	Deli / Bodega	Hotel	Bar	Donut Shop
188	Laurelton	Queens	40.667884	-73.740256	New York	3	Caribbean Restaurant	Cosmetics Shop	Park	Train Station	Duty-free Shop
192	Somerville	Queens	40.597711	-73.796648	New York	3	Auto Workshop	Nature Preserve	Park	Surf Spot	Falafel Restaurant
203	Todt Hill	Staten Island	40.597069	-74.111329	New York	3	Park	Yemeni Restaurant	Falafel Restaurant	Electronics Store	Empanada Restaurant
302	Bayswater	Queens	40.611322	-73.765968	New York	3	Playground	Yemeni Restaurant	Falafel Restaurant	Eastern European Restaurant	Electronics Store
305	Parkwoods	North York	43.753259	-79.329656	Toronto	3	Food & Drink Shop	Park	Fast Food Restaurant	Yemeni Restaurant	Factory
314	Glencairn	North York	43.709577	-79.445073	Toronto	3	Pizza Place	Park	Bakery	Japanese Restaurant	Yemeni Restaurant
324	Caledonia-Fairbanks	York	43.689026	-79.453512	Toronto	3	Park	Women's Store	Pool	Falafel Restaurant	Electronics Store
335	Scarborough Village	Scarborough	43.744734	-79.239476	Toronto	3	Cosmetics Shop	Playground	Falafel Restaurant	Eastern European Restaurant	Electronics Store
338	The Danforth East	East York/East Toronto	43.685347	-79.338106	Toronto	3	Convenience Store	Park	Yemeni Restaurant	Falafel Restaurant	Electronics Store
351	North Park, Maple Leaf Park, Upwood Park	North York	43.713756	-79.490074	Toronto	3	Park	Construction & Landscaping	Bakery	Basketball Court	Yemeni Restaurant
354	Willowdale, Newtonbrook	North York	43.789053	-79.408493	Toronto	3	Park	Yemeni Restaurant	Falafel Restaurant	Electronics Store	Empanada Restaurant
363	Lawrence Park	Central Toronto	43.728020	-79.388790	Toronto	3	Park	Swim School	Bus Line	Yemeni Restaurant	Farm
367	York Mills West	North York	43.752758	-79.400049	Toronto	3	Convenience Store	Construction & Landscaping	Park	Yemeni Restaurant	Falafel Restaurant
378	Kingsview Village, St. Phillips, Martin Grove ...	Etobicoke	43.688905	-79.554724	Toronto	3	Park	Sandwich Place	Bus Line	Mobile Phone Shop	Yemeni Restaurant
384	Moore Park, Summerhill East	Central Toronto	43.689574	-79.383160	Toronto	3	Lawyer	Park	Restaurant	Playground	Tennis Court
386	Milliken, Agincourt North, Steeles East, L'Amo...	Scarborough	43.815252	-79.284577	Toronto	3	Intersection	Park	Playground	Falafel Restaurant	Electronics Store

Discussion

Our analysis found that New York has close to 10199 venues compared to Toronto, which had 2149 venues. Therefore, we can assume that New York is going to be bigger than Toronto. Consequently, we should find alternate, very active and crowded; however, we might find it difficult to find remote and less populated spots.

New York has most of its neighborhoods spread around Cluster 0 and 4. Unfortunately, we could not define a similar location in cluster 1 for New York, and we did find that there are only two neighborhoods that qualified for Toronto.

Bronx and Brooklyn's neighborhoods dominate in Cluster 0, whereas Brooklyn and Manhattan neighborhoods dominate in Cluster 4. Most of the downtown area of Toronto belongs to Cluster 4. This illustrates that anyone moving from the downtown area in either place should find it easier and comfortable in Manhattan or vice-versa.

Cluster 2 represents the prime locations of restaurants, so it should be suitable for people who would want to be somewhere in the middle but not very close to the action of the downtown.

Cluster 0 represents neighborhoods that are a bit far away from the business neighborhoods and rely on public transport for commuting.

Cluster 3 represents people who love parks or like physical activities; this is perhaps meant for young families with kids. Therefore, they should find quite a selection in neighborhoods away from the busy and buzzing life of the city.

Cluster 4 represents restaurants and shopping neighborhoods. People who love to dine or shop would love these neighborhoods.

Conclusion

Therefore, based on our analysis, we are hopeful that this algorithm model would benefit people moving between two cities of New York and Toronto. We are optimistic that this would comfort people new to the town and find a similar environment as they were in their previous city. Furthermore, this should also be beneficial to those looking to move within the neighborhoods in the city as the model can also list similar neighborhoods in the same city.

As we control the number of clusters, we did not increase the cluster count as it increases the risk of outliers for additional clusters, like what we see in Cluster 1. However, we could expand the number to fine-tune the similarities closer to its accuracy grouping.

We were limited with the venue information provided by Foursquare, and adding additional information from other sources would help the K-means clustering model to be more accurate with new additional information. However, keeping in mind, the K-means is not great when dealing with the outliers as it would assign it to cluster based on the weakest similarities. Hence this model should be seen as helpful in shortlisting the neighborhoods. There should be additional research performed by the user to finalize and ensure the actual match.