Comparing the Neighborhoods of Toronto and New York Cities using Clustering

Praveen Kumar Pitta

15 May 2020

Part I

Introduction and Problem Statement

In this project, we will study, analyze, cluster, and compare the neighborhoods of two important cities in the world: Toronto which is in Canada and New York City which is in United States of America. We will investigate on what kinds of businesses are common in both cities, what kinds of businesses are more common in one of the two cities than the other city, and what kinds of businesses are not common in both cities.

Doing this project will enable us to get a better understanding of similarities and differences between the two cities which will make it known to business people what types of businesses are more likely to thrive in both cities, what are the neighborhoods that are suitable for each type of business, and what types of businesses are not very desirable in each city. This allows businesspeople to take better and more effective decisions regarding where to open their businesses.

Toronto is the most populous city in Canada. It’s recognized as one of the most multicultural and cosmopolitan cities in the world. Toronto also is a very diverse city: over 160 languages are spoken in it. On the economic side, Toronto is an international Centre for business and finance, and it is considered the financial capital of Canada.

Figure 1: Left: A picture of Toronto. Right: A picture of New York City

The second city of interest in this project is New York City (NYC). It is one of the most populous cities in the United States of America. Also, NYC is the most linguistically diverse city in the world: as many as 800 languages are spoken in it. Moreover, NYC plays an essential role in the economics of USA: if New York City were a sovereign state, it would have the 12th highest GDP in the world. New York City consists of five boroughs: Brooklyn, Queens, Manhattan, The Bronx, and Staten Island.

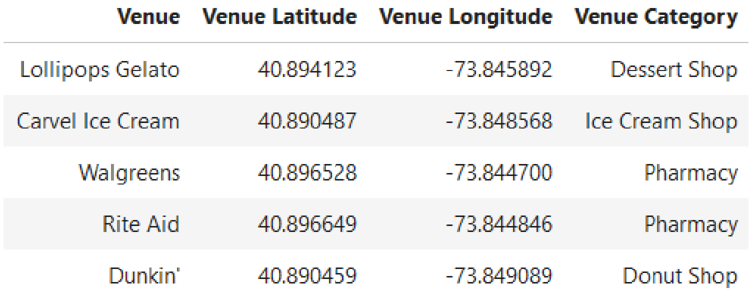
Part II

Data Acquisition and Preparation

In this section, the processes of acquiring, cleaning, and preparing each dataset used in this project for next stages will be specified. To be able to do this project, two types of data areneeded:

- **Neighborhood Data**: datasets that lists the names of the neighborhoods of Toronto and NYC and their latitude and longitude coordinates. We have some of this data provided by the instructors of "IBM Data Science Professional Certificate" and we also need to scrape some data from the internet.

- **Venues data**: data that describes the top 100 venues (restaurants, cafes, parks, museums, etc.) in each neighborhood of the two cities. The data should list the venues of each neighborhood with their categories. For example:



Venue	Venue Latitude	Venue Longitude	Venue Category
Lollipops Gelato	40.894123	-73.845892	Dessert Shop
Carvel Ice Cream	40.890487	-73.848568	Ice Cream Shop
Walgreens	40.896528	-73.844700	Pharmacy
Rite Aid	40.896649	-73.844846	Pharmacy
Dunkin'	40.890459	-73.849089	Donut Shop

Figure 2: Example of the venues data

This data will be retrieved from Foursquare which is one of the world largest sources of location and venue data. Foursquare API will be utilized to get and download the data.

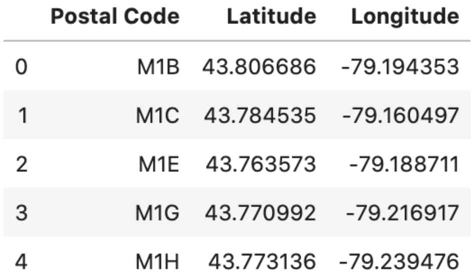
1 Neighborhood Data

For each city, data that describes the names of its neighborhoods and their coordinates is needed.

1.1 Toronto

For Toronto, there is no dataset that contains all needed neighborhood data; only a dataset that maps Toronto postal codes to latitude and longitude coordinates was provided by the organizers of “Applied Data Science Capstone” course mentioned above;

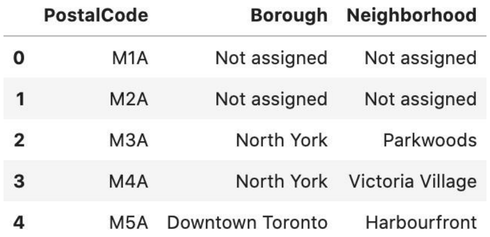
Figure 3 shows few rows of this dataset. Another dataset that lists the neighborhoods and their postal codes should be used so the combination of the two datasets produces the desired results.



	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

Figure 3: Toronto’s postal codes with their coordinates

There is a Wikipedia page titled “List of postal codes of Canada: M”. This page lists the postal codes in Canada that start with the letter M which are the postal codes of Toronto city; it lists the postal codes with the neighborhood and borough name associated with each postal code. To download this web page and extract the relevant data from it, Pandas read_html() functions can be used. It reads HTML tables on a web page in a list of dataframes. Figure 4 shows the first few rows of the dataframe extracted from that web page.



	PostalCode	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	Harbourfront

Figure 4: Toronto’s postal codes with neighborhood and borough names

In that dataframe, there are 77 records out of 288 where the “Borough” variable has the value “Not assigned”; for these 77 records, the “Neighborhood” variable also has the value “Not as- signed”; Figure 7 shows some examples of these records. Thus, these records will be deleted because they don’t carry meaningful information regarding Toronto neighborhoods.

Figure 5 shows a map of Toronto city and its neighborhoods; each blue circle represents the location of one neighborhood or a group of neighborhoods that share the same coordinates.

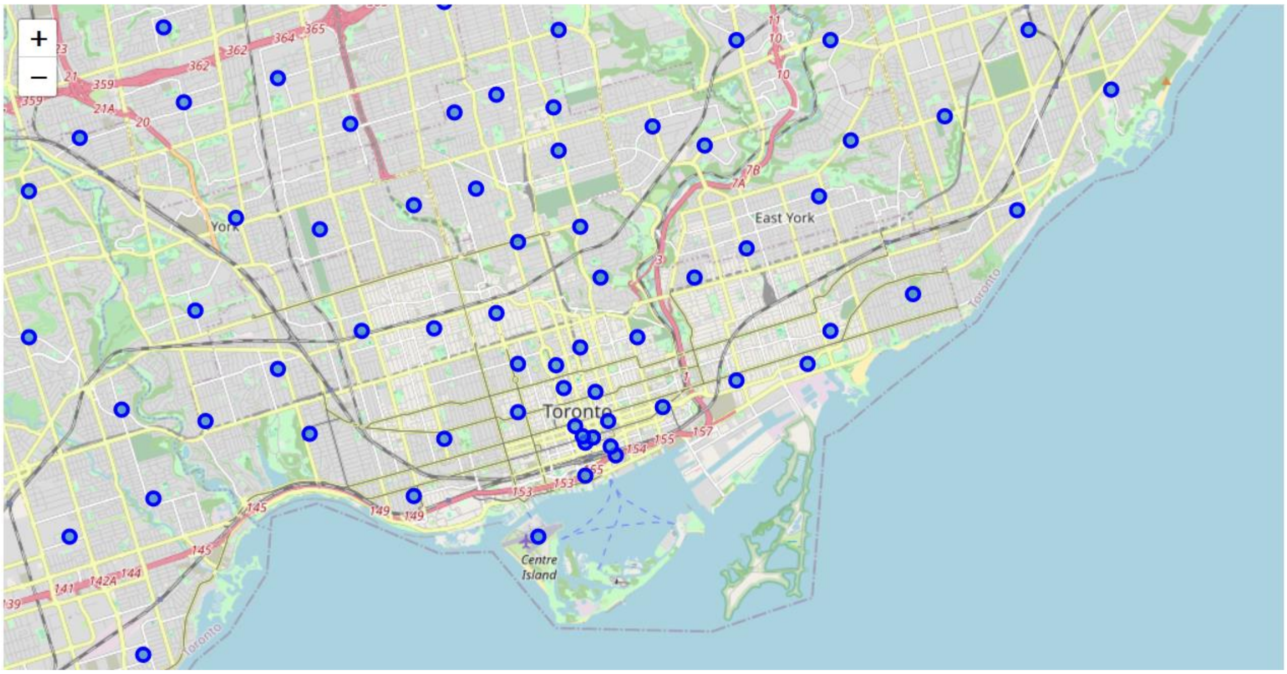
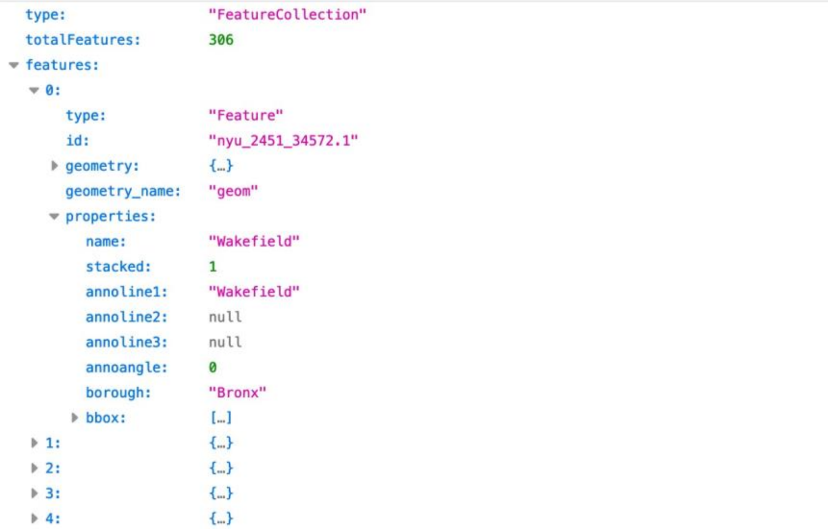


Figure 5: A map of Toronto and its neighborhoods

1.2 New York City

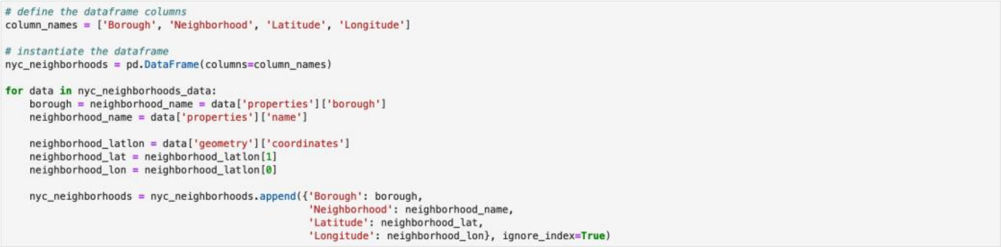
A dataset that specifies the neighborhood data for New York City was provided by the organizers of “Applied Data Science Capstone” course which is provided by IBM. The dataset is originally a JSON file that specifies the name of each neighborhood, its coordinates—latitude and longitude, its borough, and other data too. Figure 6 shows a part of this JSON file.



```
  type:                "FeatureCollection"
  totalFeatures:       306
▾ features:
  ▾ 0:
      type:            "Feature"
      id:              "nyu_2451_34572.1"
    ▸ geometry:        {…}
      geometry_name:   "geom"
    ▾ properties:
        name:          "Wakefield"
        stacked:       1
        annoline1:     "Wakefield"
        annoline2:     null
        annoline3:     null
        annoangle:     0
        borough:       "Bronx"
      ▸ bbox:          […]
  ▸ 1:                 {…}
  ▸ 2:                 {…}
  ▸ 3:                 {…}
  ▸ 4:                 {…}
```

Figure 6: A part of the JSON file that describes NYC neighborhoods

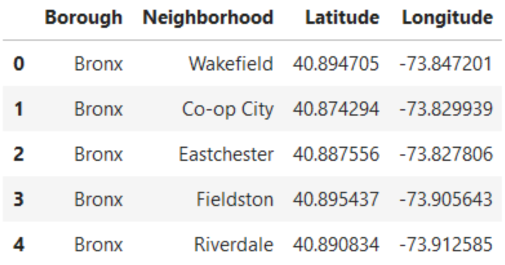
To be able to use the data of this JSON file in the later parts of this project, it should be stored in a Pandas dataframe. Figure 7 shows the Python code used to process the JSON file data and store it in a dataframe named nyc_neighborhoods. Note that in the figure, the JSON file is stored in a variable named nyc_neighborhoods_data.



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

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

for data in nyc_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]

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

Figure 7: The code used to store NYC neighborhood data into a dataframe

Figure 8 shows the resulting dataframe which contains data on **306** neighborhoods.



	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

Figure 8: The NYC neighborhood-data dataframe

Having data of the coordinates of NYC neighborhoods, it is possible to draw a map using Folium Python package of NYC and its neighborhoods. Figure 9 shows this map; each orange circle represents the location of one neighborhood.

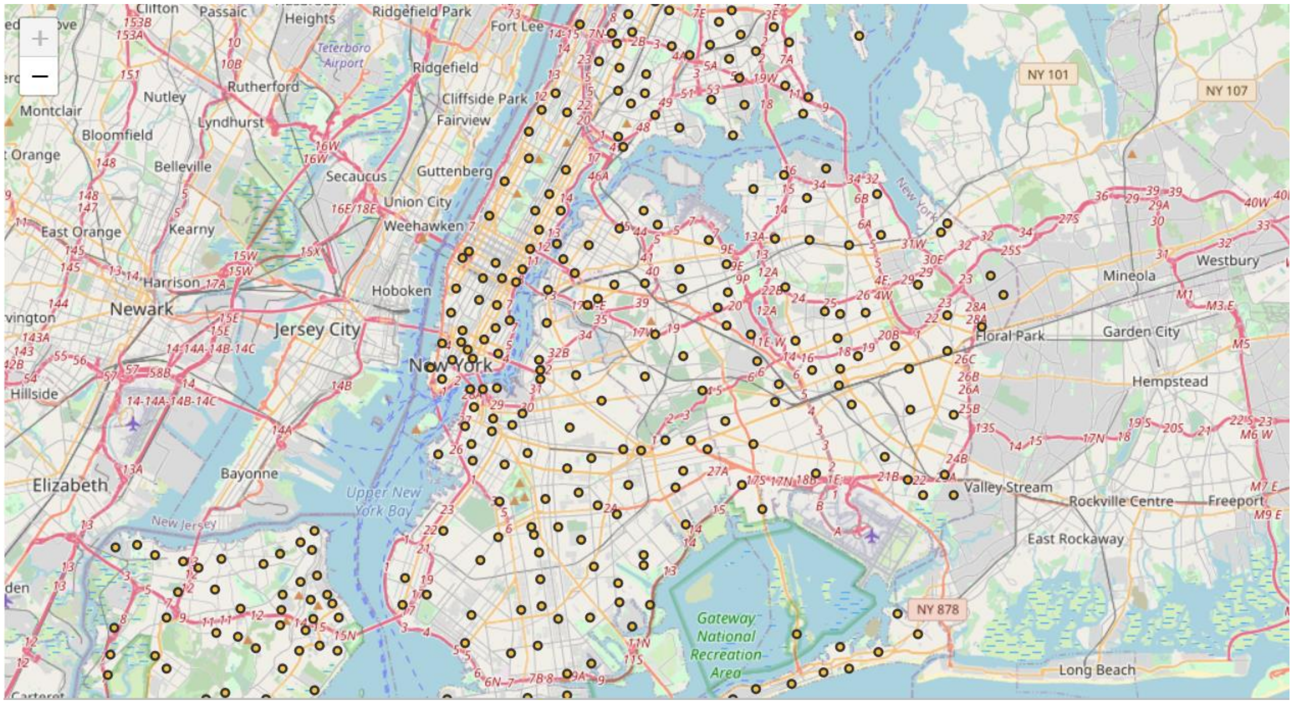


Figure 9: A map of NYC and its neighborhoods

2 Venues Data

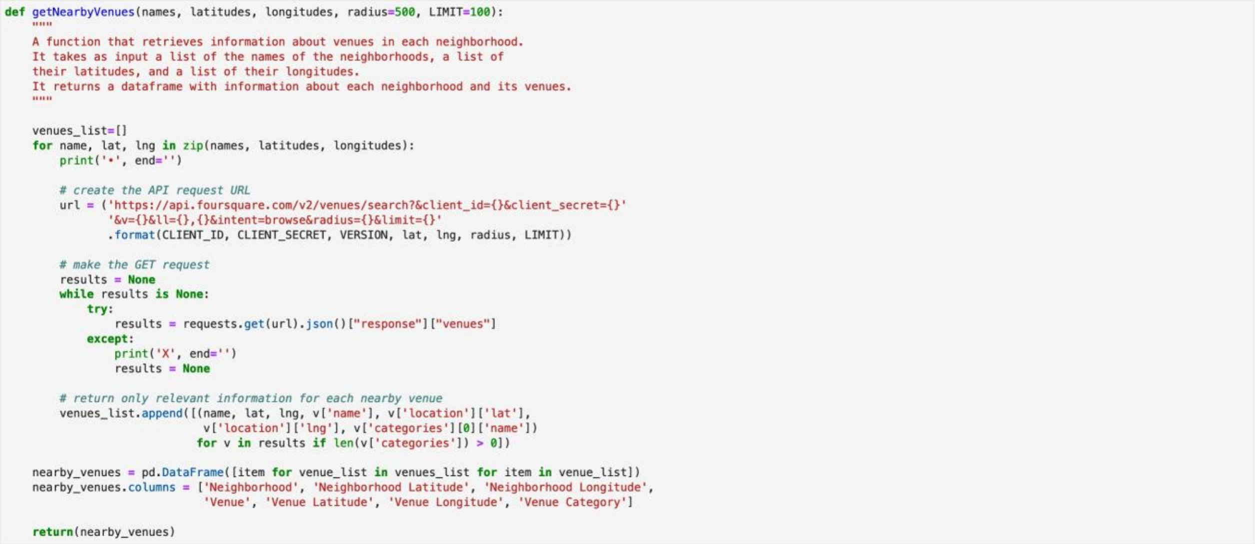
For each city, data that describes the venues of its neighborhoods and the categories of these venues is needed. Venues data will be retrieved from Foursquare which is a popular source of location and venue data. Foursquare API service will be utilized to access and download venuesdata.

To retrieve data from Foursquare using their API, a URL should be prepared and used to re- quest data related a specific location. An example URL is the following:

https://api.foursquare.com/v2/venues/search?
&client_id=1234&client_secret=1234&v=2020505& ll=40.89470517661,-
73.84720052054902&radius=500&limit=100

where searchindicates the API endpoint used, client_idand client_secretare credentials used to access the API service and are obtained when registering a Foursquare developer account, v indicates the API version to use, ll indicates the latitude and longitude of the desired location, radius is the maximum distance in meters between the specified location and the retrieved venues, and limitis used to limit the number of returned results if necessary.

Figure 10 shows the code used to create a function that takes as input the names, latitudes, and longitudes of the neighborhoods, and returns a dataframe with information about each neighbor- hood and its venues. It creates an API URL for each neighborhood and retrieves data about the venues of that neighborhoods from Foursquare.



```python
def getNearbyVenues(names, latitudes, longitudes, radius=500, LIMIT=100):
    """
    A function that retrieves information about venues in each neighborhood.
    It takes as input a list of the names of the neighborhoods, a list of
    their latitudes, and a list of their longitudes.
    It returns a dataframe with information about each neighborhood and its venues.
    """

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

        # create the API request URL
        url = ('https://api.foursquare.com/v2/venues/search?&client_id={}&client_secret={}'
               '&v={}&ll={},{}&intent=browse&radius={}&limit={}'
               .format(CLIENT_ID, CLIENT_SECRET, VERSION, lat, lng, radius, LIMIT))

        # make the GET request
        results = None
        while results is None:
            try:
                results = requests.get(url).json()["response"]["venues"]
            except:
                print('X', end='')
                results = None

        # return only relevant information for each nearby venue
        venues_list.append([(name, lat, lng, v['name'], v['location']['lat'],
                             v['location']['lng'], v['categories'][0]['name'])
                            for v in results if len(v['categories']) > 0])

    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)
```

Figure 10: Code used to build a venues dataframe for a city neighborhood

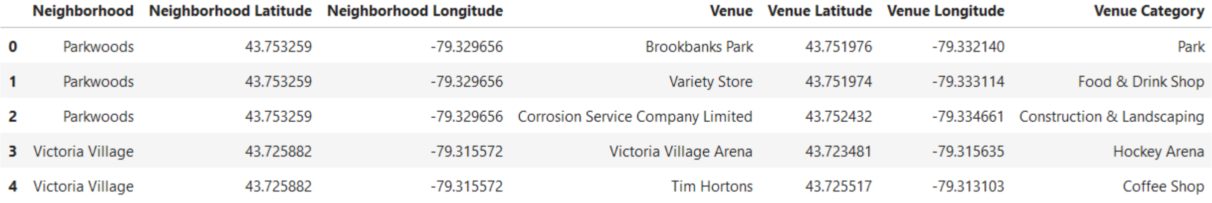
After retrieving the venue data, venues whose category is “Building”, “Office”, “Bus Line”, “Bus Station”, “Bus Stop”, or “Road” were excluded because they are not expected to add analytical value in this project.

2.1 Toronto

Using the function in Figure 10 with Toronto neighborhood data, retrieve venues of Toronto neighborhoods; Figure 11 shows a part of it. The dataframe contains data for more than **1000** venues in Toronto.

Different numbers of venues were found in different neighborhoods: for example, data about 30 venues were returned for Central Bay Street neighborhood and 4 venues for York Mills West neighborhood.

As mentioned above, data on more than 1000 venues was returned. Each venue of them belongs to one of 226 unique categories.

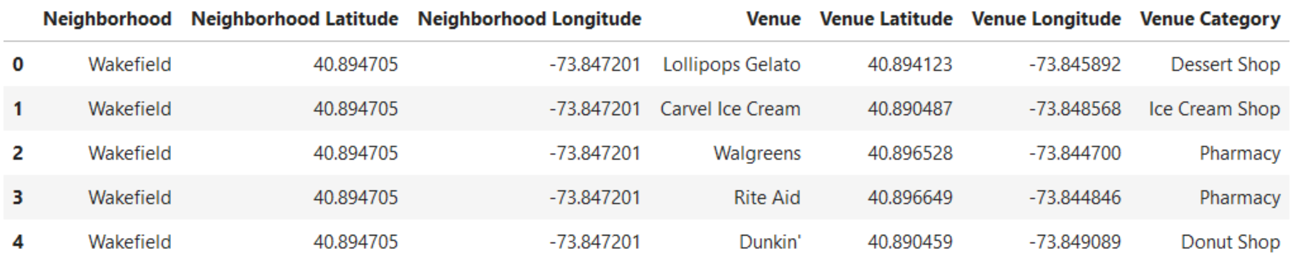


	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Parkwoods	43.753259	-79.329656	Brookbanks Park	43.751976	-79.332140	Park
1	Parkwoods	43.753259	-79.329656	Variety Store	43.751974	-79.333114	Food & Drink Shop
2	Parkwoods	43.753259	-79.329656	Corrosion Service Company Limited	43.752432	-79.334661	Construction & Landscaping
3	Victoria Village	43.725882	-79.315572	Victoria Village Arena	43.723481	-79.315635	Hockey Arena
4	Victoria Village	43.725882	-79.315572	Tim Hortons	43.725517	-79.313103	Coffee Shop

Figure 11: Venue dataframe for Toronto

2.2 New York City

Using the function in Figure 10 with NYC neighborhood data retrieved data about 6**,000** venues in NYC neighborhoods. For each venue, venue name, category, latitude, and longitude were retrieved. The head of the dataframe returned by the function for NYC is shown in Figure 12. We can see that each row in the dataframe contains data about one venue: the venue name, coordinates (latitude and longitude), and category in addition to the neighborhood in which the venue is located and the coordinates of the neighborhood.



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

Figure 12: Venue dataframe for NYC

Part III

Exploratory Data Analysis

In this section, the datasets produced in the previous section will be explored via effective visualizations to understand the data better.

1 Most Common Venue Categories

What are the categories that have more venues than the others in NYC and Toronto? To answer this question, the number of occurrences is counted for each venue category in Figure 13 (for Toronto) and Figure 14 (for NYC). After doing so, a bar plot can be used to visualize the popularity of the most common venue categories in each city.

1.1 Toronto

Figure 13 shows a bar plot of the most common venues in Toronto. For Toronto, the most common venue category is “Coffee Shop” with around ~100 venues. Then comes “Cafe” category with ~75 venues. And in the third place appears “Park” with around ~40 venues.

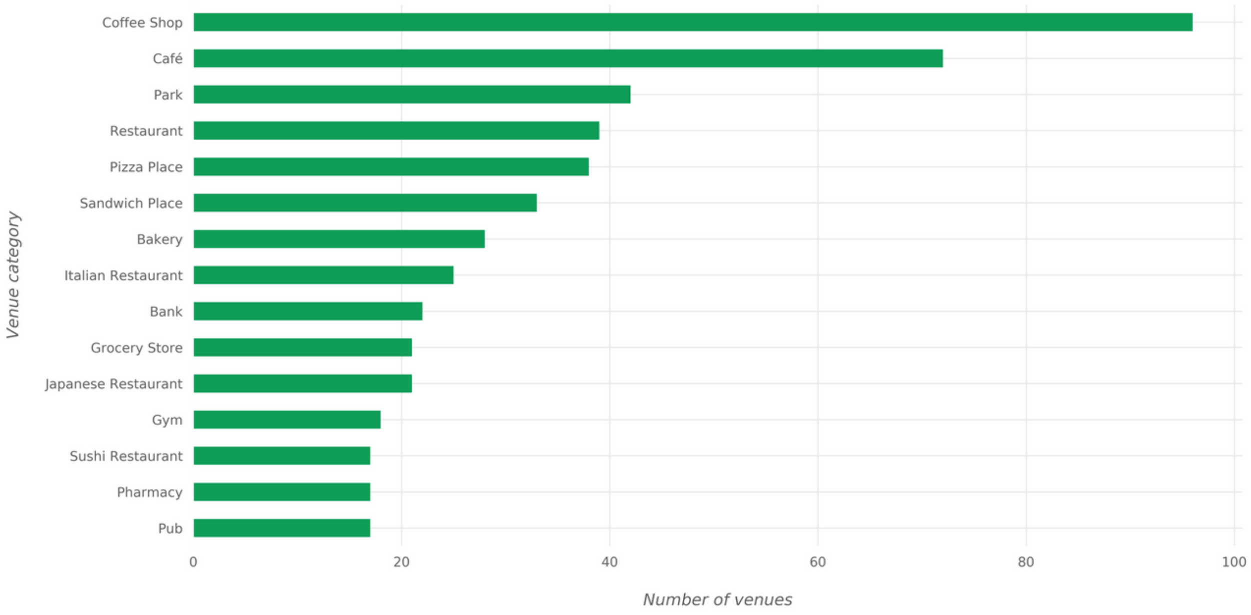


Figure 13: Most common venue categories in Toronto

1.1 New York City

Figure 14 shows a bar plot of the most common venues in NYC. We can see that the most common category is “Pizza Place” with ~300 venues in NYC; this means that there are ~300 Pizza Places in NYC. In the second rank, the category “Deli / Bodega” appears with ~190 venues; according to Oxford dictionary, a deli is a shop selling cooked meats, cheeses, and unusual or foreign pre- pared foods; and a bodega is a small grocery shop, especially in a Spanish-speaking neighborhood.In the third rank comes the “Coffee Shop” category appears with ~160 venues.

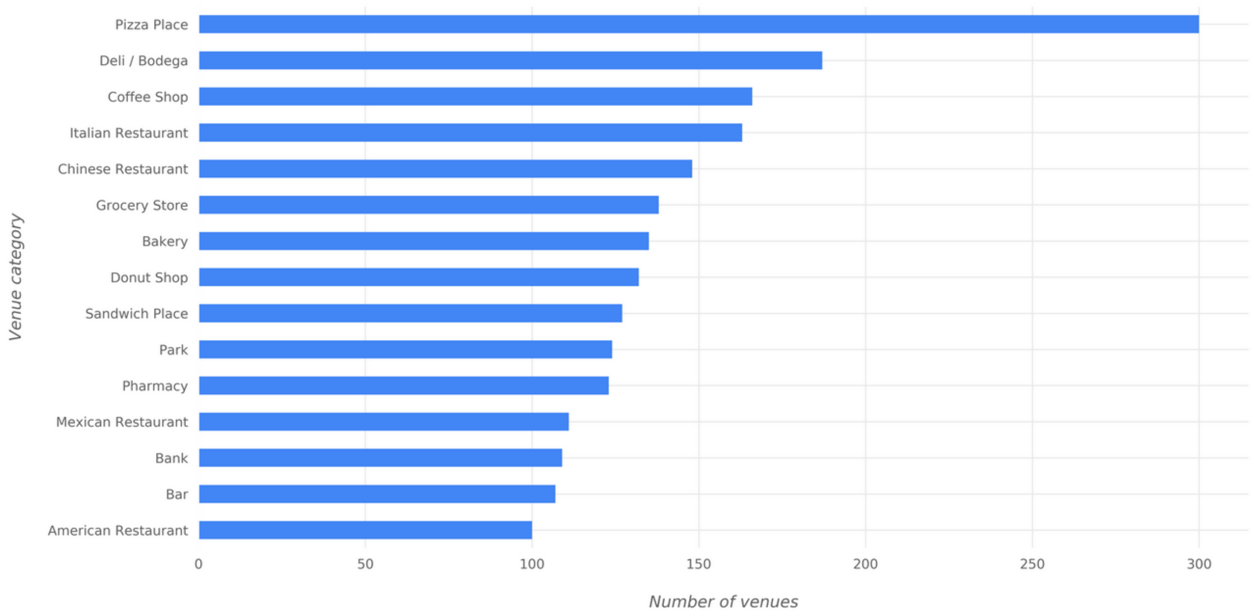


Figure 14: Most common venue categories in NYC

It can be seen that there are similarities between the preferences of most common categories in Toronto and in NYC: we see many categories appearing in both plots of Figure 13 and Figure 14.

2 Most Widespread Venue Categories

Now another question is to be answered: What are the venue categories that exist in more neighborhood? This question is different than the one mentioned in 1. To explain the difference with an example, suppose that there are 15 venues with the category “VR Games” and that these venues exist in 7 neighborhoods only out of 80 neighborhoods; also suppose that there are 10 venues with the category “Syrian Restaurant” and that these venues exist in 10 neighborhoods—each one of them in a different neighborhood. Then it can be said that the “VR Games” category is more com- mon than “Syrian Restaurant” category because there are more venues under this category, and it can be said that the “Syrian Restaurant” category is more widespread than the “VR Games” category because venues under this category exist in more neighborhoods than the other category.

2.1 Toronto

Figure 15 shows the most widespread venue categories in Toronto. As with NYC, the order of the most-widespread-categories in Toronto differs than the order of the most common categories. In the first place comes the “Coffee Shop” category with venues in ~50 neighborhoods[2]. Then comes “Cafe” category with venues in ~35 neighborhoods. And the third most-widespread category is “Park” with venues in ~35 neighborhoods.

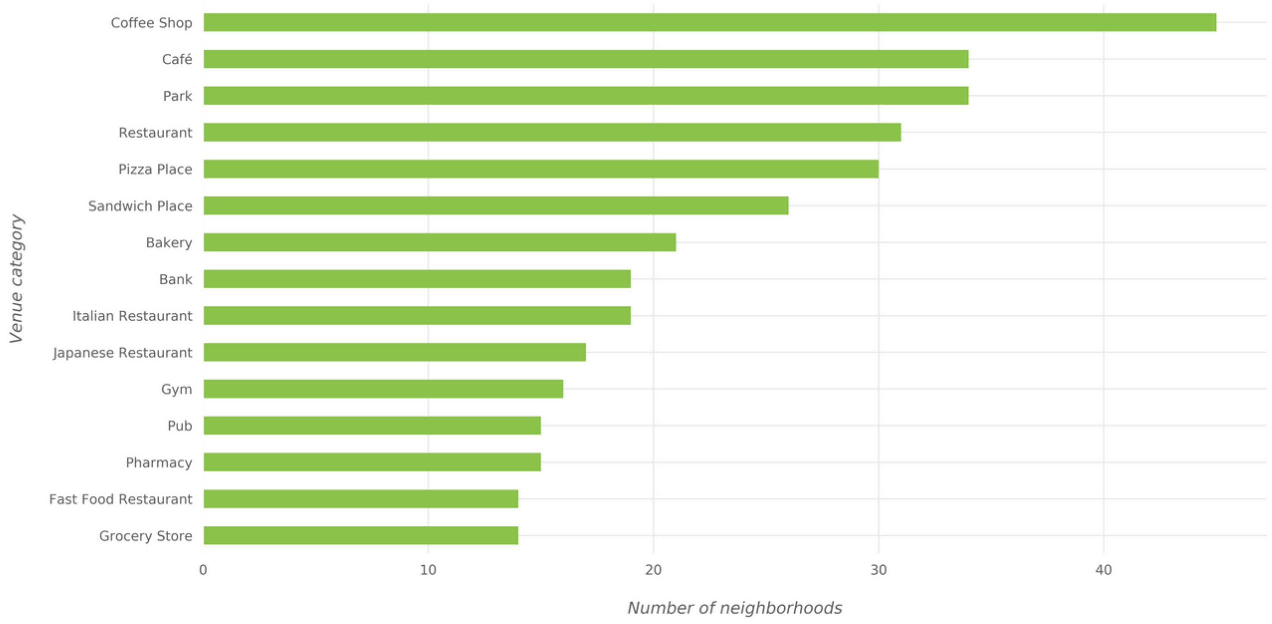


Figure 15: Most widespread venue categories in Toronto

2.2 New York City

Figure 16 shows the most widespread venue categories in NYC. It can be seen that the order of categories this time is different than that of the most common categories (Figure 15). The most widespread category is “Pizza Place”; Salons and barbershops exist in ~250 neighbor- hoods out of the 306 neighborhoods. After that comes the “Deli / Bodega” category with venues in ~250 neighborhoods also. In the third place comes the “Residential Building (Apartment / Condo)” category with venues in ~230 neighborhoods.

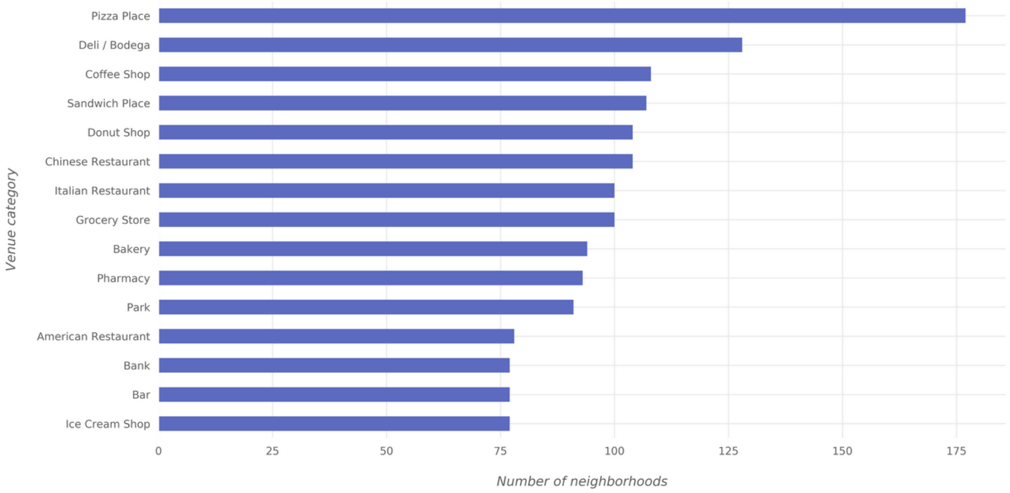


Figure 16: Most widespread venue categories in NYC

Part IV

Clustering of Neighborhoods

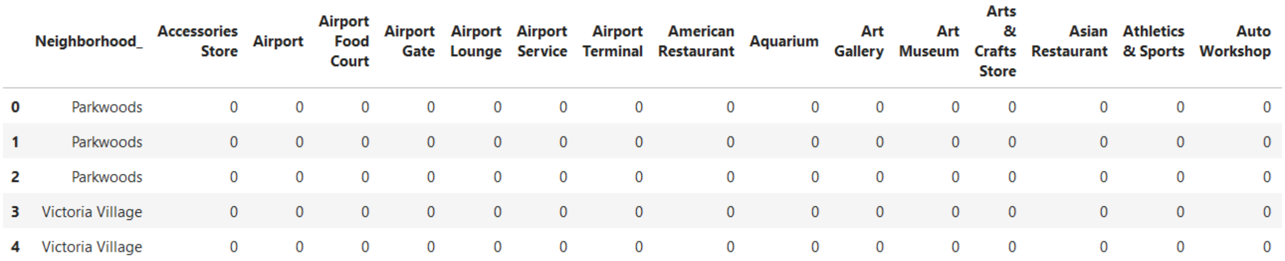
In this section, clustering will be applied on NYC and Toronto neighborhoods to find similar neighborhoods in the two cities. Clustering is the process of finding similar items in a dataset based on the characteristics (features) of items in the dataset. In particular, K-means clustering algorithm of the Scikit-learn Python library will be used. To be able to perform clustering, a dataset suitable for clustering is needed; the datasets described in Figure 13 and Figure 14 are not ready to be used with clustering algorithms.

1 Feature Selection

The goal of the clustering is to cluster neighborhoods based on the similarity of venue categories in the neighborhoods. This means that the two things of interest here are the neighborhood and the venue categories in the neighborhood. Thus, the following two features will be selected out of the dataframes of Figure 11 and Figure 12: “Neighborhood” and “Venue Category”. But still after that, the data is not ready for the clustering algorithm because the algorithm works with numerical features.

For that, one-hot encoding will be applied on the “Venue Category” feature and the result of the encoding will be used for clustering. One-hot encoding will be applied on the Toronto data and on NYC data then, as will be explained later, the data of the two cities will be combined.

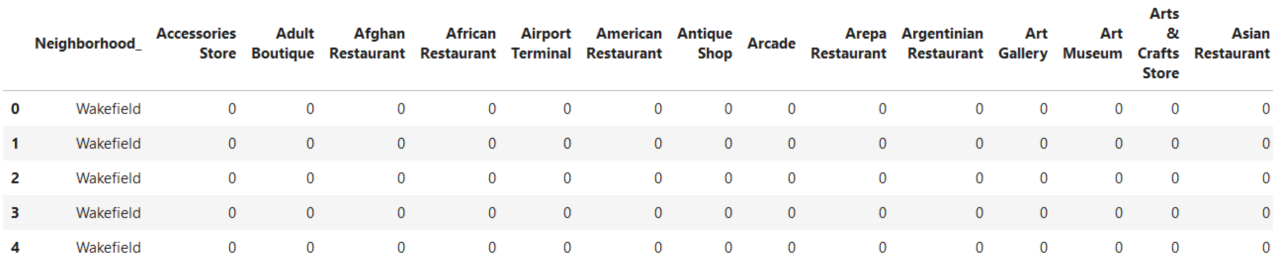
After applying one-hot encoding on Toronto data, the resulting dataframe looks like the one shown in Figure 17.



	Neighborhood_	Accessories Store	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Aquarium	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Auto Workshop
0	Parkwoods	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	Parkwoods	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Parkwoods	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Victoria Village	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	Victoria Village	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 17: The result of one hot encoding on Toronto data

After applying one-hot encoding on NYC data, the resulting dataframe looks like the one shown in Figure 18.



	Neighborhood_	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	Airport Terminal	American Restaurant	Antique Shop	Arcade	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant
0	Wakefield	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	Wakefield	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Wakefield	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Wakefield	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	Wakefield	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 18: The result of one hot encoding on NYC data

Note that in Figure 17 and Figure 28, the column that contains neighborhood names is named “Neighborhood_” instead of just “Neighborhood”. This was done because there is a venue cat- egory called “Neighborhood” so the neighborhood-names columns was given the name “Neigh- borhood_” to avoid having two columns with the same name.

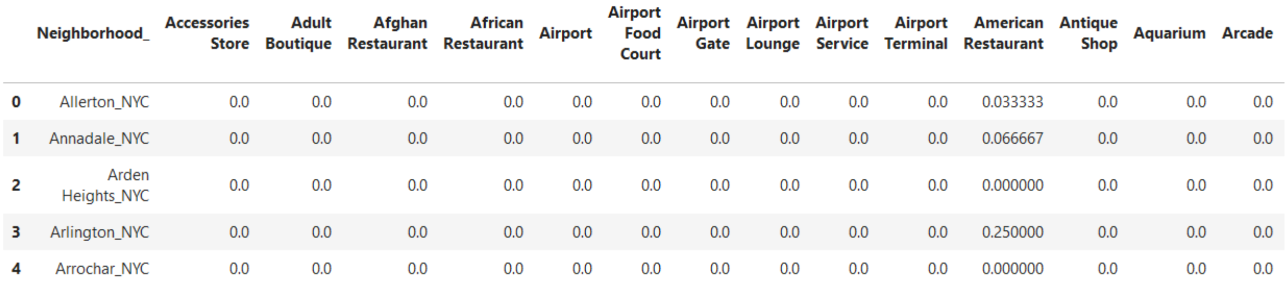
The next step is aggregating the values for each neighborhood so that each neighborhood be- comes represented by only one row. The aggregation will be done by grouping rows by neighbor- hood and by taking the mean of the frequency of occurrence of each category. So for example, if the Wakefield neighborhood has 15 venues (i.e. 15 rows in the dataframe of Figure 18) and then Wakefield row in the aggregated dataframe .

2 Combining NYC and Toronto Data

After producing the aggregated dataframes for each of NYC and Toronto, these dataframes should be combined before applying the clustering algorithm. However, in order to distinguish NYC neighborhoods from Toronto neighborhoods in the new dataframe, a text string is added to the end of each neighborhood name before merging the dataframes: for NYC, the string to be added is “_NYC” and “_Toronto” for Toronto.

Also, NYC and Toronto don’t necessarily have the same venue categories (i.e. some columns in the dataframe don’t exist in the dataframe and vice versa). To deal with this issue before combining the dataframes, the columns of both dataframes are made the same by adding the columns that exist only in NYC dataframe to Toronto dataframe and vice versa; the newly-added columns have a value of 0 for all the rows.

Figure 19 shows a part of the dataframe that resulted from the combination of NYC and Toronto aggregated dataframes. This dataframes contains data on 408 neighborhoods in both NYC and Toronto.

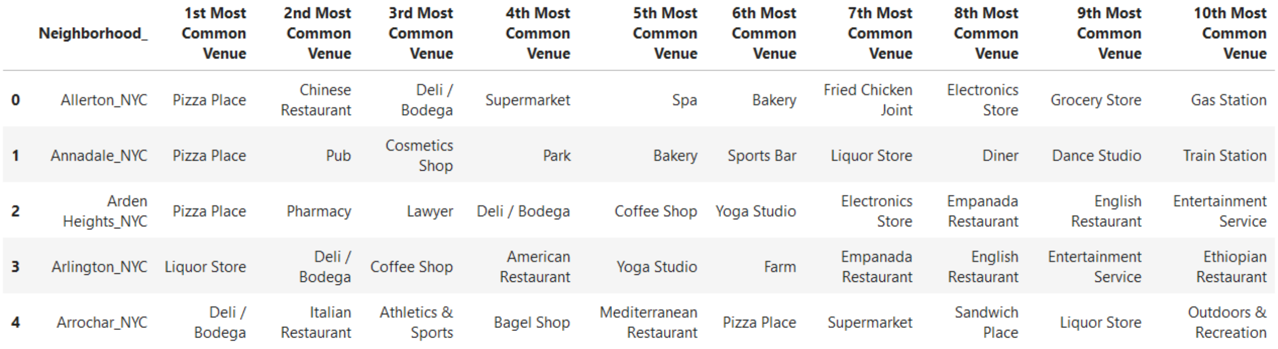


	Neighborhood_	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Antique Shop	Aquarium	Arcade
0	Allerton_NYC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.033333	0.0	0.0	0.0
1	Annadale_NYC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.066667	0.0	0.0	0.0
2	Arden Heights_NYC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
3	Arlington_NYC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.250000	0.0	0.0	0.0
4	Arrochar_NYC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0

Figure 19: The combination of NYC and Toronto aggregated dataframes

3 The Most Common Categories for Each Neighborhood

Using the dataframe of Figure 19, another dataframe is created to specify the 5 most common categories for each neighborhood in NYC and Toronto. This dataframe is created by retrieving the 5 categories with the largest values for each neighborhood in Figure 19. Figure 20 shows this dataframe.

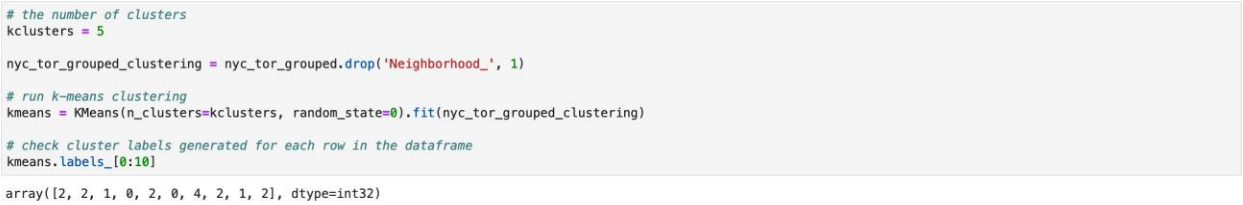


	Neighborhood_	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Allerton_NYC	Pizza Place	Chinese Restaurant	Deli / Bodega	Supermarket	Spa	Bakery	Fried Chicken Joint	Electronics Store	Grocery Store	Gas Station
1	Annadale_NYC	Pizza Place	Pub	Cosmetics Shop	Park	Bakery	Sports Bar	Liquor Store	Diner	Dance Studio	Train Station
2	Arden Heights_NYC	Pizza Place	Pharmacy	Lawyer	Deli / Bodega	Coffee Shop	Yoga Studio	Electronics Store	Empanada Restaurant	English Restaurant	Entertainment Service
3	Arlington_NYC	Liquor Store	Deli / Bodega	Coffee Shop	American Restaurant	Yoga Studio	Farm	Empanada Restaurant	English Restaurant	Entertainment Service	Ethiopian Restaurant
4	Arrochar_NYC	Deli / Bodega	Italian Restaurant	Athletics & Sports	Bagel Shop	Mediterranean Restaurant	Pizza Place	Supermarket	Sandwich Place	Liquor Store	Outdoors & Recreation

Figure 20: Most common categories for each neighborhoods

4 Clustering and its Results

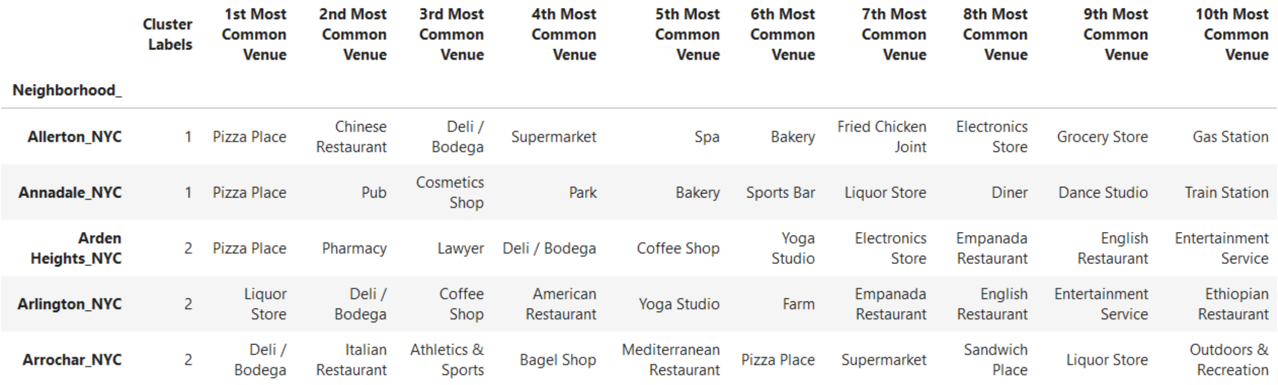
By obtaining the dataframe of Figure 19, it is now possible to apply the clustering algorithm. Figure 21 shows the code used to perform clustering using the K-means algorithm of Scikit-learn library. The variable named nyc_tor_grouped contains the dataframe of Figure 19. Notice that the "Neighborhood_" column was dropped before applying the clustering algorithm (i.e. the cluster- ing algorithm was applied on all columns except that column); this was done because the cluster- ing algorithm doesn't accept non-numerical columns as mentioned earlier. However, this column will be re-added as will be explained soon.



```
# the number of clusters
kclusters = 5

nyc_tor_grouped_clustering = nyc_tor_grouped.drop('Neighborhood_', 1)

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

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

array([2, 2, 1, 0, 2, 0, 4, 2, 1, 2], dtype=int32)
```

Figure 21: Code used to perform K-means clustering

As can be seen in Figure 21, the clustering algorithm produced cluster-labels; these labels de- note the cluster of each record (i.e. each neighborhood) in the data. Using these labels and the dataframe of Figure 19, a dataframe is constructed to show the neighborhoods of NYC and Toronto, the cluster to which each neighborhood belongs, and the most common venue categories in each neighborhood. This dataframe can be seen in Figure 22.



	Cluster Labels	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
Neighborhood_											
Allerton_NYC	1	Pizza Place	Chinese Restaurant	Deli / Bodega	Supermarket	Spa	Bakery	Fried Chicken Joint	Electronics Store	Grocery Store	Gas Station
Annadale_NYC	1	Pizza Place	Pub	Cosmetics Shop	Park	Bakery	Sports Bar	Liquor Store	Diner	Dance Studio	Train Station
Arden Heights_NYC	2	Pizza Place	Pharmacy	Lawyer	Deli / Bodega	Coffee Shop	Yoga Studio	Electronics Store	Empanada Restaurant	English Restaurant	Entertainment Service
Arlington_NYC	2	Liquor Store	Deli / Bodega	Coffee Shop	American Restaurant	Yoga Studio	Farm	Empanada Restaurant	English Restaurant	Entertainment Service	Ethiopian Restaurant
Arrochar_NYC	2	Deli / Bodega	Italian Restaurant	Athletics & Sports	Bagel Shop	Mediterranean Restaurant	Pizza Place	Supermarket	Sandwich Place	Liquor Store	Outdoors & Recreation

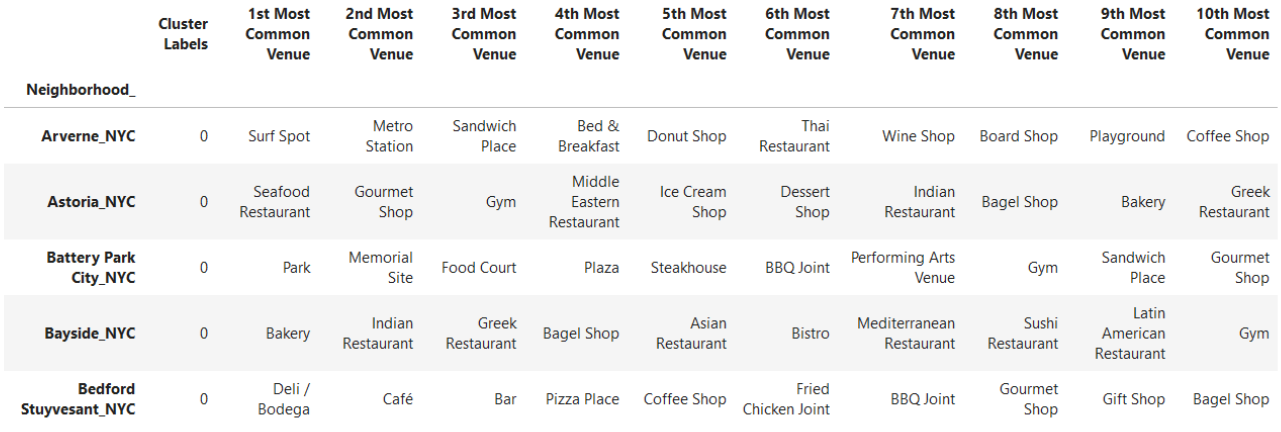
Figure 22: NYC and Toronto neighborhoods, their clusters, and their most common categories

The output of the clustering operation is 5 clusters with cluster labels 0, 1, 2, 3, and 4. Each cluster is expected to contain a group of similar neighborhoods based on the categories of the venues in each neighborhood. The clustering algorithm was run on 400 neighborhoods in NYC and Toronto. Table 2 shows the number of neighborhoods in each cluster.

Cluster	Number of neighborhoods
0	198
1	159
2	32
3	9
4	2

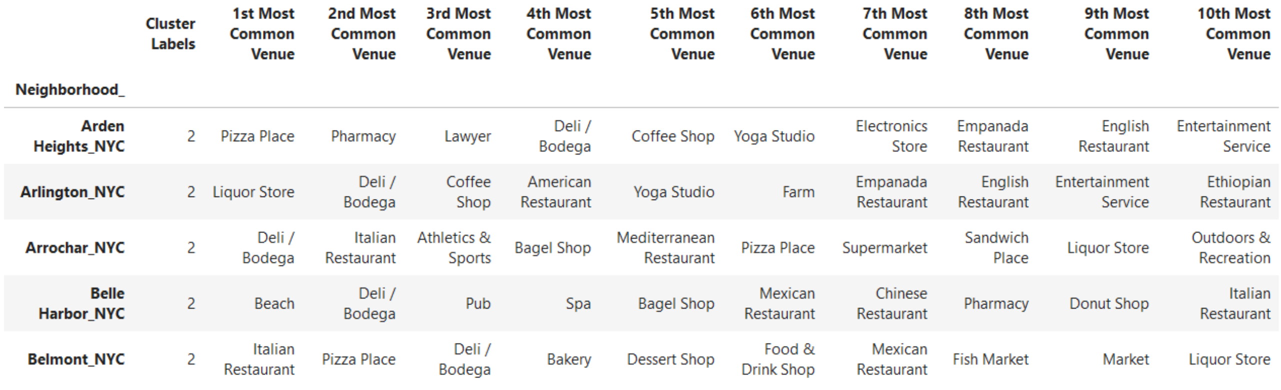
Table 2: Number of neighborhoods in each cluster

For examples, Figure 23 shows a part of the first cluster and Figure 24 shows a part of the third cluster. It's hard to show all the clusters with all of their neighborhoods since there are 406 neighborhoods in the five clusters. However, they can be accessed in the Jupyter notebook of this project.



Neighborhood_	Cluster Labels	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
Arverne_NYC	0	Surf Spot	Metro Station	Sandwich Place	Bed & Breakfast	Donut Shop	Thai Restaurant	Wine Shop	Board Shop	Playground	Coffee Shop
Astoria_NYC	0	Seafood Restaurant	Gourmet Shop	Gym	Middle Eastern Restaurant	Ice Cream Shop	Dessert Shop	Indian Restaurant	Bagel Shop	Bakery	Greek Restaurant
Battery Park City_NYC	0	Park	Memorial Site	Food Court	Plaza	Steakhouse	BBQ Joint	Performing Arts Venue	Gym	Sandwich Place	Gourmet Shop
Bayside_NYC	0	Bakery	Indian Restaurant	Greek Restaurant	Bagel Shop	Asian Restaurant	Bistro	Mediterranean Restaurant	Sushi Restaurant	Latin American Restaurant	Gym
Bedford Stuyvesant_NYC	0	Deli / Bodega	Café	Bar	Pizza Place	Coffee Shop	Fried Chicken Joint	BBQ Joint	Gourmet Shop	Gift Shop	Bagel Shop

Figure 23: Some records that belong to the first cluster

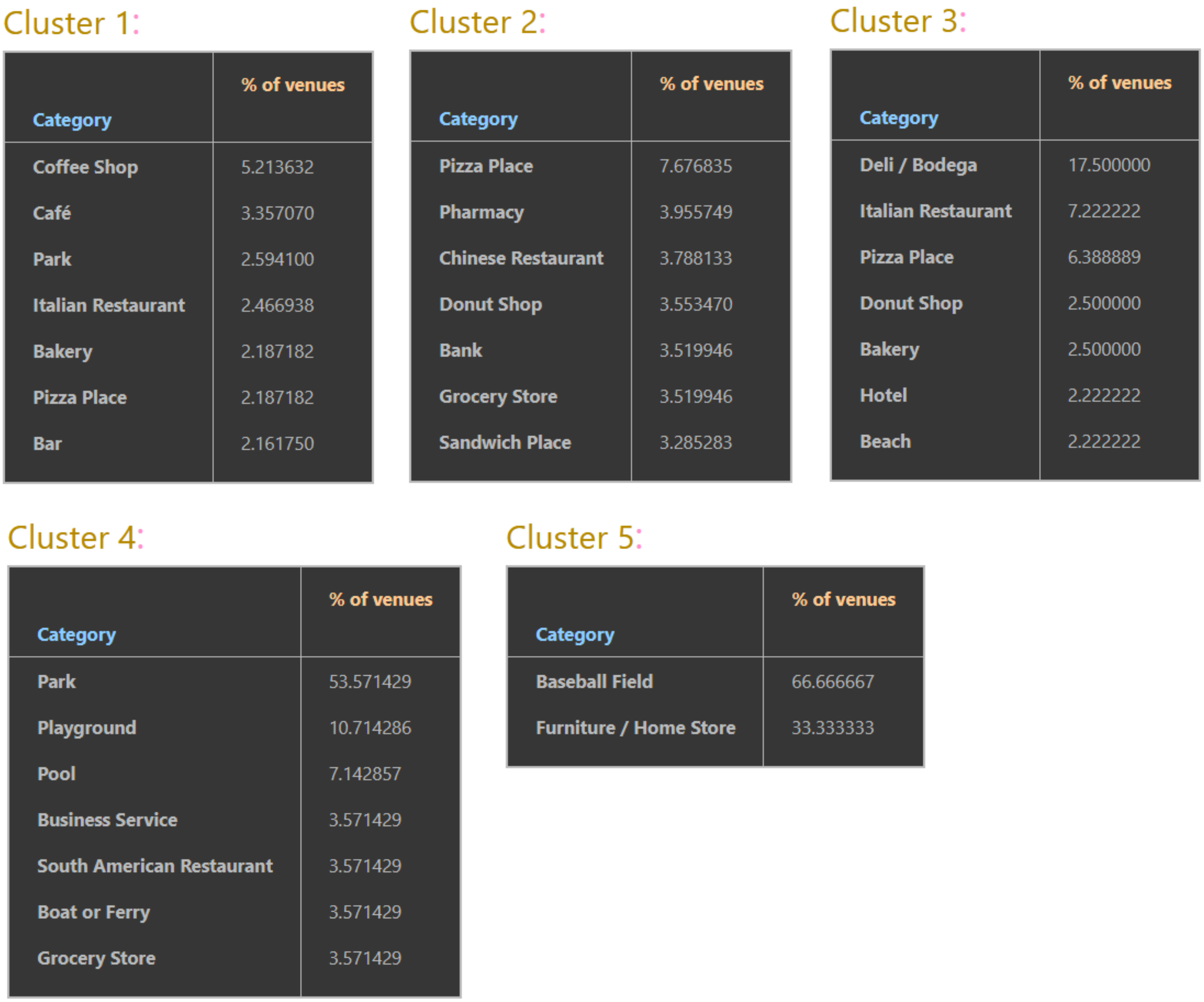


Neighborhood_	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
Arden Heights_NYC	2	Pizza Place	Pharmacy	Lawyer	Deli / Bodega	Coffee Shop	Yoga Studio	Electronics Store	Empanada Restaurant	English Restaurant	Entertainment Service
Arlington_NYC	2	Liquor Store	Deli / Bodega	Coffee Shop	American Restaurant	Yoga Studio	Farm	Empanada Restaurant	English Restaurant	Entertainment Service	Ethiopian Restaurant
Arrochar_NYC	2	Deli / Bodega	Italian Restaurant	Athletics & Sports	Bagel Shop	Mediterranean Restaurant	Pizza Place	Supermarket	Sandwich Place	Liquor Store	Outdoors & Recreation
Belle Harbor_NYC	2	Beach	Deli / Bodega	Pub	Spa	Bagel Shop	Mexican Restaurant	Chinese Restaurant	Pharmacy	Donut Shop	Italian Restaurant
Belmont_NYC	2	Italian Restaurant	Pizza Place	Deli / Bodega	Bakery	Dessert Shop	Food & Drink Shop	Mexican Restaurant	Fish Market	Market	Liquor Store

Figure 24: Some records that belong to the third cluster

5 Cluster Analysis

The clustering algorithm grouped neighborhoods of NYC and Toronto in 5 clusters based on the similarity between their venues. Now, these clusters will be investigated to see the most common categories in each of them. Figures 25 show the most common 7 venue categories in each cluster; for each common category, the percentage of venues of that category in the neighborhoods of the cluster is shown also.



Cluster 1:

Category	% of venues
Coffee Shop	5.213632
Café	3.357070
Park	2.594100
Italian Restaurant	2.466938
Bakery	2.187182
Pizza Place	2.187182
Bar	2.161750

Cluster 2:

Category	% of venues
Pizza Place	7.676835
Pharmacy	3.955749
Chinese Restaurant	3.788133
Donut Shop	3.553470
Bank	3.519946
Grocery Store	3.519946
Sandwich Place	3.285283

Cluster 3:

Category	% of venues
Deli / Bodega	17.500000
Italian Restaurant	7.222222
Pizza Place	6.388889
Donut Shop	2.500000
Bakery	2.500000
Hotel	2.222222
Beach	2.222222

Cluster 4:

Category	% of venues
Park	53.571429
Playground	10.714286
Pool	7.142857
Business Service	3.571429
South American Restaurant	3.571429
Boat or Ferry	3.571429
Grocery Store	3.571429

Cluster 5:

Category	% of venues
Baseball Field	66.666667
Furniture / Home Store	33.333333

Figure 25: Most common venue-categories in each of the 5 clusters

The differences between the clusters can be seen from the figure; each cluster distinguishably has different distribution of common venue categories than other clusters. Some of the observa- tions that can be made from the tables of Figure 25 are:

- While Pizza Place constitute ~2% of venues in the neighborhoods of the first cluster, they constitute ~7% of the venues in the second and ~6% of the venues inthe third cluster.
- Bakery appear in the most common categories of the first and third clusters only.
- Grocery Store appear in the most common categories of second and fourth cluster only.

Other differences can be observed in Figure 25.

Figure 26 shows the number of NYC neighborhoods and the number of Toronto neighborhoods in each cluster of the five resulting clusters.

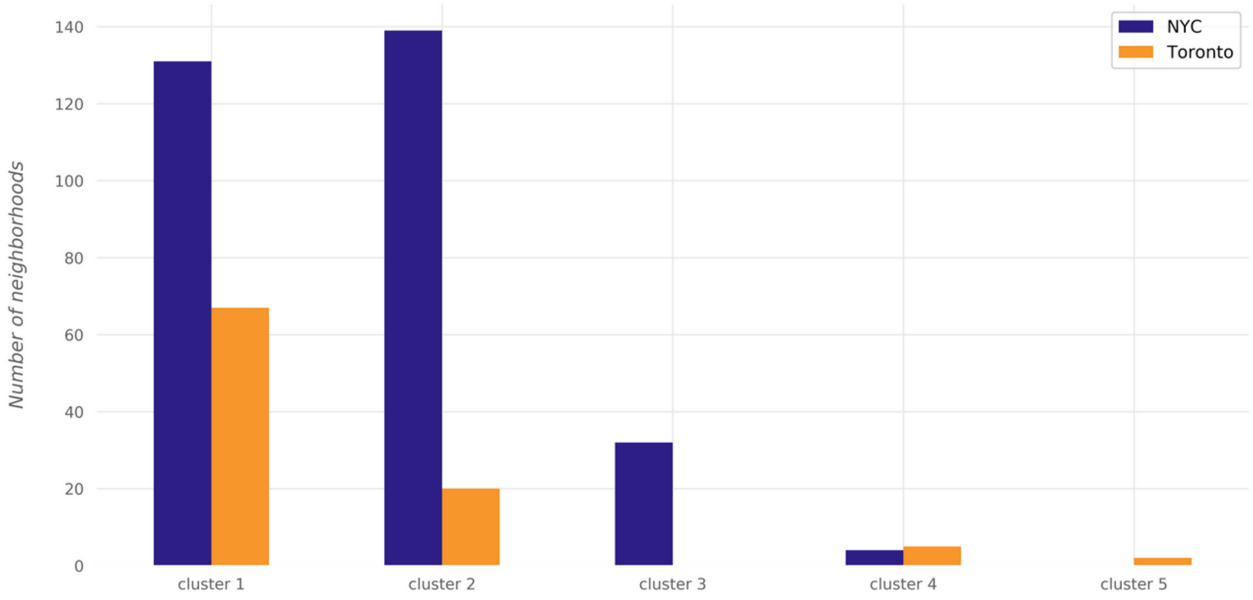


Figure 26: The number of NYC and Toronto neighborhoods in each cluster

Part V

Conclusions

In this project, the neighborhoods of New York City and Toronto were clustered into multiple groups based on the categories (types) of the venues in these neighborhoods. The results showed that there are venue categories that are more common in some cluster than the others; the most common venue categories differ from one cluster to the other. If a deeper analysis—taking more aspects into account—is performed, it might result in discovering different *style* in each cluster based on the most common categories in the cluster.