Capstone Project – The Battle between two cities

By

Shaoni Chakravarty

December 25,2019

Contents

Results

- For Toronto Dataset

  i) Cluster 0 set
  ii) Cluster 1 set
  iii) Cluster 2 set
  iv) Cluster 3 set
  v) Cluster 4 set

- For New York Dataset

  i) Cluster 0 set
  ii) Cluster 1 set
  iii) Cluster 2 set
  iv) Cluster 3 set
  v) Cluster 4 set
  vi) Cluster 5 set
  vii) Cluster 6 set
  viii) Cluster 7 set

Discussion

Conclusion

Reference

Introduction

Background

The city we are working on are : New York City & Toronto. Both the cities are very popular city of USA. Both the cities are very diversified. It is the home of more than 8.1 million people. Many immigrants also live here from various parts of the world. More than 800 languages can be found in both the cities.

As both the cities are based on the diverse culture, so people from different traditional cuisine can be found there. From Asian Cuisine to Middle Eastern , American authentic food can be found there. Now to open a new food chain or restaurant will be a challenge from business point of view as well as people should accept the taste of the food being served.

Problem

While working on this, we have faced certain issues. The idea of this project to categorically segment the neighborhoods of NYC & Toronto into major clusters & examine the cuisine. This project will help to understand the diversity of neighborhood by leveraging venue data from Foursquare API calls & K-means Clustering , an unsupervised ML algorithm. It will be helpful for a new vendor who is willing to open a new restaurant.

Data

Toronto Dataset Details:

Toronto link : https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M

This link helps to get Toronto data including location geographical locations. This helps in getting ide a about the neighborhoods of Toronto.

New York Dataset Details:

New York link : https://cocl.us/new_york_dataset

This link acts as a guide to New York Citys neighborhood that appear on the web resource. The json extract is as follows:

```
{'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]}}
```

Foursquare API:

Link: https://developer.foursquare.com/docs

Foursquare API, a location data provider, will be used to make API calls to retrieve data abou t venues in different neighborhoods.

Methodology

Download & Explore Toronto Dataset :

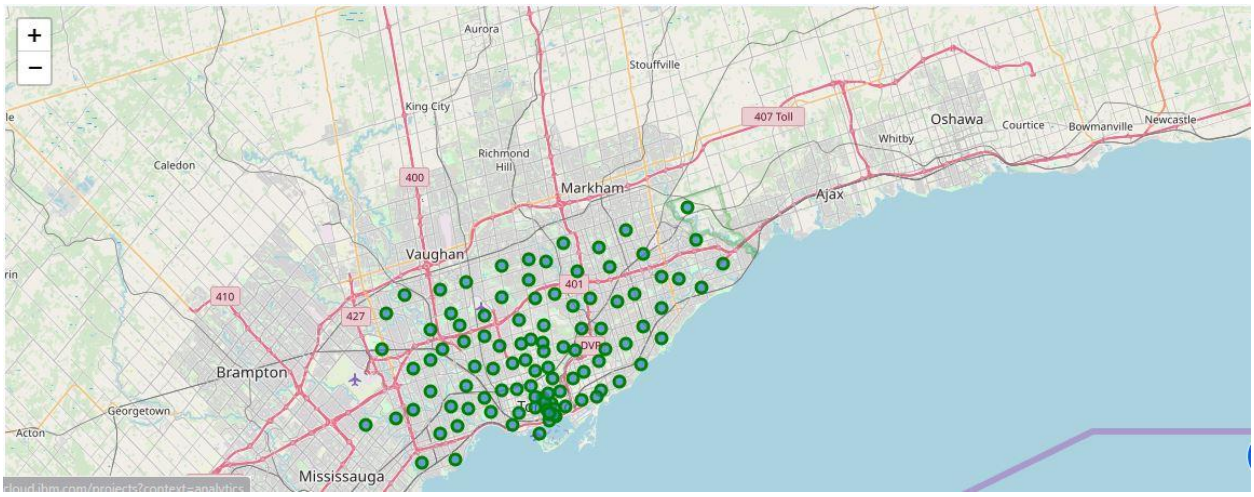
In order to segment neighborhoods in Toronto, we found a dataset that contains 12 borough s and 103 neighborhoods, that exist in each borough, with respective latitude & longitude co ordinates. The dataset was downloaded from a wiki table using Beautiful Soup. We found a data with the neighborhoods. We have added the geospatial column to get the geographical co-ordiantes as well :

```
dfll= pd.read_csv("http://cocl.us/Geospatial_data")
dfll.rename(columns={'Postal Code':'Postcode'}, inplace=True)
dfll.set_index("Postcode")
df2.set_index("Postcode")
toronto_data=pd.merge(df2, dfll)
toronto_data.head()
```

3]:

	Postcode	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	43.654260	-79.360636
3	M6A	North York	Lawrence Heights, Lawrence Manor	43.718518	-79.464763
4	M7A	Queen's Park (Toronto)	Queen's Park	43.662301	-79.389494

Upon analysis, we have 12 boroughs & 103 neighborhoods. Further , 'geopy' library was us ed to get the latitude & longitude values of Toronto dataset. The 'folium' library is used to d epict the map generation.



RESTful API calls to Foursquare for Toronto dataset:

The Foursqure api is used to explore neighborhoods & segment them. To access the API CLIENT_ID, CLIENT_SECRET , VERSION needs to be defind. There are many endpoints avail able on Foursquare for various GET requests. But to explore the cuisines, we need the data for the 'Food' category. As stated, The function takes restaurant names.

```
num_top_venues = 5

for hood in toronto_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = toronto_grouped[toronto_grouped['Neighborhood'] == hood].T.reset_index()
    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(num_top_venues))
    print('\n')
```

The restaurant output results in :

```
----Bedford Park, Lawrence Manor East----
                  venue  freq
0  Fast Food Restaurant  0.09
1    Italian Restaurant  0.09
2           Coffee Shop  0.09
3           Pizza Place  0.05
4      Greek Restaurant  0.05


----Berczy Park----
                venue  freq
0         Coffee Shop  0.09
1        Cocktail Bar  0.05
2            Beer Bar  0.04
3  Seafood Restaurant  0.04
4   French Restaurant  0.04
```

Download & Explore New York City Dataset :

In order to segment neighborhoods of the New York City, the dataset contains 5 boroughs & the neighborhoods, that exist in each borough with respective latitude & longitude co-ordi nates. Once the json file is downloaded , we study the structure of the data then. Dataframe is created with Borough, Neighborhood, Latitude & Longitude.

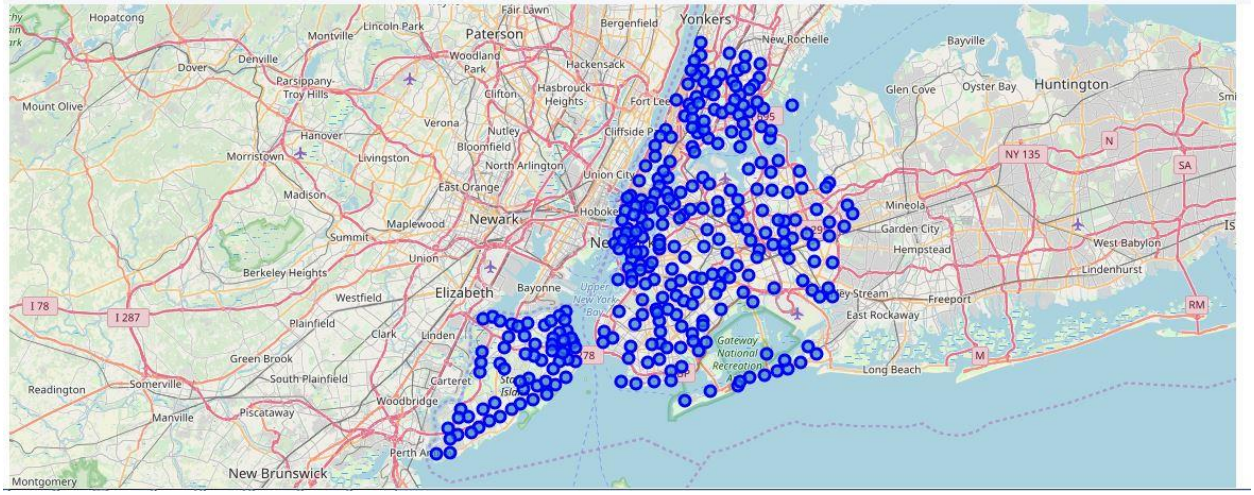
```
for data in neighborhoods_data:
    borough = neighborhood_name = data['properties']['borough']
    neighborhood_name = data['properties']['name']

    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

The curated dataframe is then used to visualize the map of the New York City with neighbo rhoods superimposed on the top. The map has been generated using python 'folium' library .



RESTful API calls to Foursquare for New York city dataset:

The Foursqure api is used to explore neighborhoods & segment them. To access the API CLIENT_ID, CLIENT_SECRET , VERSION needs to be defind. There are many endpoints avail able on Foursquare for various GET requests. But to explore the cuisines, we need the data for the 'Food' category. Upon analysis, it is found that there are 10 major categories of ven ues. Under which all other sub-categories are included .

```
for data in category_list:
    print(data['id'], data['name'])
```

```
4d4b7104d754a06370d81259 Arts & Entertainment
4d4b7105d754a06372d81259 College & University
4d4b7105d754a06373d81259 Event
4d4b7105d754a06374d81259 Food
4d4b7105d754a06376d81259 Nightlife Spot
4d4b7105d754a06377d81259 Outdoors & Recreation
4d4b7105d754a06375d81259 Professional & Other Places
4e67e38e036454776db1fb3a Residence
4d4b7105d754a06378d81259 Shop & Service
4d4b7105d754a06379d81259 Travel & Transport
```

To further understand the result of GET request, the first neighborhood of 'New York City d ataset' is explored. The first neighborhood returned is 'Wakefield'. Then a GET request url is created to search for venue with Category_id = '4d4b7105d754a06374d81259' and radi us = '500' meters.

As the aim is to segment the neighborhoods of New York City with respect to 'Food' in the v icinity, it is further required to fetch the data from all 306 neighborhood venues. To overco me the redundancy of the process followed above, a function 'getNearbyFood' is created. T his function loops in all the neighborhoods & creates API url with a limit = 100 & radius = 5 00.

```
def getNearbyFood(names, latitudes, longitudes, radius=1000, LIMIT=500):
    not_found = 0
    print('***Start ', end='')
    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={},{}&radius={}&categoryId={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            "4d4b7105d754a06374d81259", # "Food" category id
            LIMIT)

        try:
            # make the GET request
            results = requests.get(url).json()['response']['venues']

            # return only relevant information for each nearby venue
            venues_list.append([(
                name,
                lat,
                lng
```

Pickle :

Pickle is very important & easy-to-use library. It is used to serialize the data retrieved from GET requests, to make a 'pkl' file. This file can later be deserialized to retrieve an exact pyt hon object structure.

```
import pickle # to serialize and deserialize a Python object structure
try:
    with open('nyc_food_venues.pkl', 'rb') as f:
        nyc_venues = pickle.load(f)
    print("---Dataframe Existed and Deserialized---")
except:
    nyc_venues = getNearbyFood(names=neighborhoods['Neighborhood'],
                               latitudes=neighborhoods['Latitude'],
                               longitudes=neighborhoods['Longitude']
                              )
    with open('nyc_food_venues.pkl', 'wb') as f:
        pickle.dump(nyc_venues, f)
    print("---Dataframe Created and Serialized---")
```

```
***Start  . . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
Done*** with 0 venues with incompelete information.
```

The dataframe is as follows:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Wakefield	40.894705	-73.847201	Pitman Deli	40.894149	-73.845748	Food
1	Wakefield	40.894705	-73.847201	Dunkin'	40.890459	-73.849089	Donut Shop
2	Wakefield	40.894705	-73.847201	Cooler Runnings Jamaican Restaurant Inc	40.898083	-73.850259	Caribbean Restaurant
3	Wakefield	40.894705	-73.847201	McDonald's	40.902645	-73.849485	Fast Food Restaurant
4	Wakefield	40.894705	-73.847201	Fever Tropical Cuisine	40.892555	-73.857666	Comfort Food Restaurant

As of now, two separate dataframes have created:

1.'neighborhoods' that contain neighborhood, Latitude,Longitude, Borough

2. 'nyc_venues' is a merger between 'neighborhood' & 'food' category . Also eah venues ha s its own latitude,longitude & category.

Exploratory Data Analysis:

➢ Toronto Data Analysis:

To have a data analysis on Toronto dataset, we need to depict the dataframe 'neighborhoods_venues_sorted' . It shows the common venues in the 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	Agincourt	Skating Rink	Latin American Restaurant	Breakfast Spot	Lounge	Clothing Store	Women's Store	Eastern European Restaurant	Doner Restaurant	Donut Shop	Drugstore
1	Agincourt North, L'Amoreaux East, Milliken, On...	Park	Playground	Dumpling Restaurant	Diner	Discount Store	Dog Run	Doner Restaurant	Donut Shop	Drugstore	Women's Store
2	Alderwood, Long Branch	Pizza Place	Skating Rink	Sandwich Place	Dance Studio	Gym	Coffee Shop	Pub	Pharmacy	Pool	Diner
3	Bathurst Manor, Wilson Heights, Downsview North	Coffee Shop	Shopping Mall	Pharmacy	Bank	Supermarket	Deli / Bodega	Sushi Restaurant	Diner	Middle Eastern Restaurant	Fried Chicken Joint
4	Bathurst Quay, King and Spadina, Railway Lands...	Airport Service	Airport Lounge	Airport Terminal	Boat or Ferry	Boutique	Rental Car Location	Plane	Harbor / Marina	Airport Gate	Airport Food Court

There are unique neighborhood in Toronto.

```
print('There are {} uniques categories.'.format(len(toronto_venues['Venue Category'].unique())))
```

```
There are 277 uniques categories.
```

➢ New York City Data Analysis :

The merged dataframe 'nyc_venues' has all required information. It has 13952 rows.

```
print(nyc_venues.shape)
nyc_venues.head()
```

```
(13952, 7)
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Wakefield	40.894705	-73.847201	Pitman Deli	40.894149	-73.845748	Food
1	Wakefield	40.894705	-73.847201	Dunkin'	40.890459	-73.849089	Donut Shop
2	Wakefield	40.894705	-73.847201	Cooler Runnings Jamaican Restaurant Inc	40.898083	-73.850259	Caribbean Restaurant
3	Wakefield	40.894705	-73.847201	McDonald's	40.902645	-73.849485	Fast Food Restaurant
4	Wakefield	40.894705	-73.847201	Fever Tropical Cuisine	40.892555	-73.857666	Comfort Food Restaurant

There are 193 unique categories with most occurring venues as follows:

```
In [99]: print('There are {} uniques categories.'.format(len(nyc_venues['Venue Category'].unique())))
         nyc_venues.groupby('Venue Category')['Venue Category'].count().sort_values(ascending=False)
```

```
There are 193 uniques categories.

Out[99]: Venue Category
         Deli / Bodega                1059
         Pizza Place                  1040
         Coffee Shop                   930
         Donut Shop                    658
         Fast Food Restaurant          628
         Chinese Restaurant            620
         Bakery                        555
         Italian Restaurant            551
         American Restaurant           462
         Café                          411
         Bagel Shop                    354
         Caribbean Restaurant          341
         Mexican Restaurant            332
         Sandwich Place                317
         Diner                         285
         Fried Chicken Joint           279
```

Data Cleaning

As there is cultural diversity in the neighborhood, its important to clean the data with generalized categories. Here, by generalized categories, it means that these venues are common across different culture & different food habbits. So, first unique categ ories have been listed down.

```
unique_categories = nyc_venues['Venue Category'].unique().tolist()
print(', '.join(str(x) for x in unique_categories))
```

```
Food, Donut Shop, Caribbean Restaurant, Fast Food Restaurant, Comfort Food Restaurant, Fried Chicken Joint, Ice Cream Shop, Deli / Bodega, Pizza Plac
e, Food Truck, Chinese Restaurant, Bagel Shop, Latin American Restaurant, Restaurant, Coffee Shop, Spanish Restaurant, Bakery, Mexican Restaurant, Gas
tropub, Seafood Restaurant, Dumpling Restaurant, Frozen Yogurt Shop, BBQ Joint, American Restaurant, Steakhouse, Italian Restaurant, Burger Joint, Din
er, Juice Bar, Café, Wings Joint, Grocery Store, Breakfast Spot, Asian Restaurant, Tapas Restaurant, Bar, Japanese Restaurant, Indian Restaurant, Gree
k Restaurant, Sandwich Place, Vegetarian / Vegan Restaurant, New American Restaurant, German Restaurant, Cuban Restaurant, Lounge, Pub, Sushi Restaura
nt, Cafeteria, Salvadoran Restaurant, Gas Station, Other Nightlife, Soup Place, Southern / Soul Food Restaurant, Food Court, Hot Dog Joint, Arcade, De
ssert Shop, French Restaurant, Convenience Store, Food & Drink Shop, Empanada Restaurant, Mediterranean Restaurant, Cocktail Bar, Supermarket, African
Restaurant, Food Stand, Middle Eastern Restaurant, Cheese Shop, Vietnamese Restaurant, Paella Restaurant, Fish & Chips Shop, Snack Place, Taco Place,
Theme Restaurant, Sports Bar, Thai Restaurant, Buffet, Mac & Cheese Joint, Peruvian Restaurant, Cupcake Shop, Bubble Tea Shop, South American Restaura
nt, Turkish Restaurant, Arepa Restaurant, South Indian Restaurant, Eastern European Restaurant, Building, Irish Pub, Korean Restaurant, Office, Beer G
arden, Noodle House, Lebanese Restaurant, Dim Sum Restaurant, Tea Room, Poke Place, Cantonese Restaurant, Szechuan Restaurant, Hotpot Restaurant, Mala
y Restaurant, Polish Restaurant, Whisky Bar, Music Venue, Beer Bar, Clothing Store, Jewish Restaurant, Fish Market, Ramen Restaurant, Caucasian Restau
rant, Russian Restaurant, Varenyky restaurant, Bistro, Afghan Restaurant, Modern European Restaurant, Halal Restaurant, Ukrainian Restaurant, Kosher R
estaurant, Burrito Place, Wine Bar, Tibetan Restaurant, Filipino Restaurant, Cajun / Creole Restaurant, Austrian Restaurant, Gourmet Shop, Pie Shop, U
don Restaurant, Salad Place, Indie Movie Theater, Ethiopian Restaurant, Art Gallery, Kebab Restaurant, Gift Shop, Fruit & Vegetable Store, Dive Bar, F
actory, Street Food Gathering, Israeli Restaurant, Farmers Market, Venezuelan Restaurant, Moroccan Restaurant, Tex-Mex Restaurant, Shanghai Restauran
t, Candy Store, Event Space, Falafel Restaurant, Miscellaneous Shop, Skating Rink, Organic Grocery, Taiwanese Restaurant, Burmese Restaurant, Hotel, F
```

This data-preperation totally depends on 'Data Analyst' discretion & can be modifie d as required.

```
# manually create a list of generalized categories
general_categories = ['Dessert Shop','Food','Ice Cream Shop','Donut Shop','Bakery','Sandwich Place','Comfort Food Restaurant',
                      'Deli / Bodega','Food Truck','Bagel Shop','Burger Joint','Restaurant','Frozen Yogurt Shop','Coffee Shop',
                      'Diner','Wings Joint','Café','Juice Bar','Breakfast Spot','Grocery Store','Bar','Cupcake Shop',
                      'Pub','Fish & Chips Shop','Cafeteria','Other Nightlife','Arcade','Hot Dog Joint','Food Court',
                      'Health Food Store','Convenience Store','Food & Drink Shop','Cocktail Bar','Cheese Shop',
                      'Snack Place','Sports Bar','Lounge','Theme Restaurant','Buffet','Bubble Tea Shop','Building',
                      'Irish Pub','College Cafeteria','Tea Room','Supermarket','Hotpot Restaurant','Gastropub','Beer Garden',
                      'Fish Market','Beer Bar','Clothing Store','Music Venue','Bistro','Salad Place','Wine Bar','Gourmet Shop',
                      'Indie Movie Theater','Art Gallery','Gift Shop','Pie Shop','Fruit & Vegetable Store',
                      'Street Food Gathering','Dive Bar','Factory','Farmers Market','Mac & Cheese Joint','Creperie',
                      'Candy Store','Event Space','Skating Rink','Miscellaneous Shop','Gas Station','Organic Grocery',
                      'Pastry Shop','Club House','Flea Market','Hotel','Furniture / Home Store','Bookstore','Pet Café',
                      'Gym / Fitness Center','Flower Shop','Financial or Legal Service','Hotel Bar','Hookah Bar','Poke Place',
                      'Market','Gluten-free Restaurant','Smoothie Shop','Butcher','Food Stand','Beach Bar','Beach',
                      'Soup Place','Rock Club','Residential Building (Apartment / Condo)','Laundry Service',
                      'Government Building','Bowling Alley','Nightclub','Park','Moving Target']
```

Engineering discussions:

➢ Toronto :

Now, each neighborhood is analyzed individually to understand the common cuisine served within 500 meters of the vicinity.

Analyzing neighborhoods

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

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

# 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()
```

	Yoga Studio	Accessories Store	Afghan Restaurant	Airport	Airport Food Court	Airport Gate	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	...	Trail	Train Station	Vegetarian / Vegan Restaurant	Video Game Store	Video Store	Vietnamese Restaurant	Warehouse Store	Wine Bar	Wings Joint
0	0	0	0	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	0	0	0	0
2	0	0	0	0	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	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

From this depiction it's not clear which cuisine is popular for Toronto.

➢ New York City :

Now, each neighborhood is analyzed individually to understand the common cuisine served within 500 meters of the vicinity.

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

6]:

	Afghan Restaurant	African Restaurant	American Restaurant	Arepa Restaurant	Argentinian Restaurant	Asian Restaurant	Australian Restaurant	Austrian Restaurant	BBQ Joint	Brazilian Restaurant	Burmese Restaurant	Burrito Place	Cajun / Creole Restaurant	Cantonese Restaurant	Caribbean Restaurant	Caucasian Restaurant
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	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	1	0
3	0	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	0

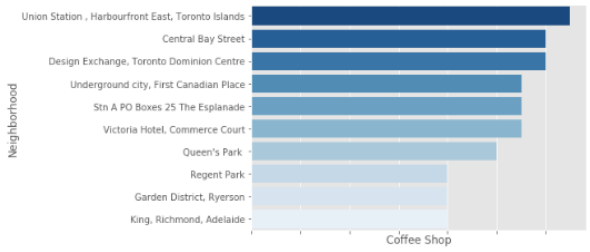
The top 10 venue-categories can also be depicted by counting its occurrences. According t o findings, Korean , Carribean & Chinese Restaurants are the most found cuisine types.

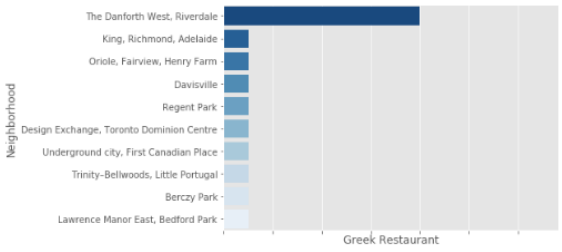
	count	mean	std	min	25%	50%	75%	max
Korean Restaurant	302.0	0.354305	2.004223	0.0	0.0	0.0	0.0	31.0
Caribbean Restaurant	302.0	1.129139	2.614783	0.0	0.0	0.0	1.0	16.0
Chinese Restaurant	302.0	2.052980	2.007587	0.0	1.0	2.0	3.0	15.0
Italian Restaurant	302.0	1.824503	2.112845	0.0	0.0	1.0	3.0	14.0
Indian Restaurant	302.0	0.301325	1.108079	0.0	0.0	0.0	0.0	14.0
Fast Food Restaurant	302.0	2.079470	1.916958	0.0	1.0	2.0	3.0	11.0
Pizza Place	302.0	3.443709	2.097731	0.0	2.0	3.0	5.0	10.0
Seafood Restaurant	302.0	0.509934	0.849996	0.0	0.0	0.0	1.0	8.0
Latin American Restaurant	302.0	0.480132	1.055567	0.0	0.0	0.0	1.0	8.0
Mexican Restaurant	302.0	1.099338	1.207699	0.0	0.0	1.0	2.0	7.0

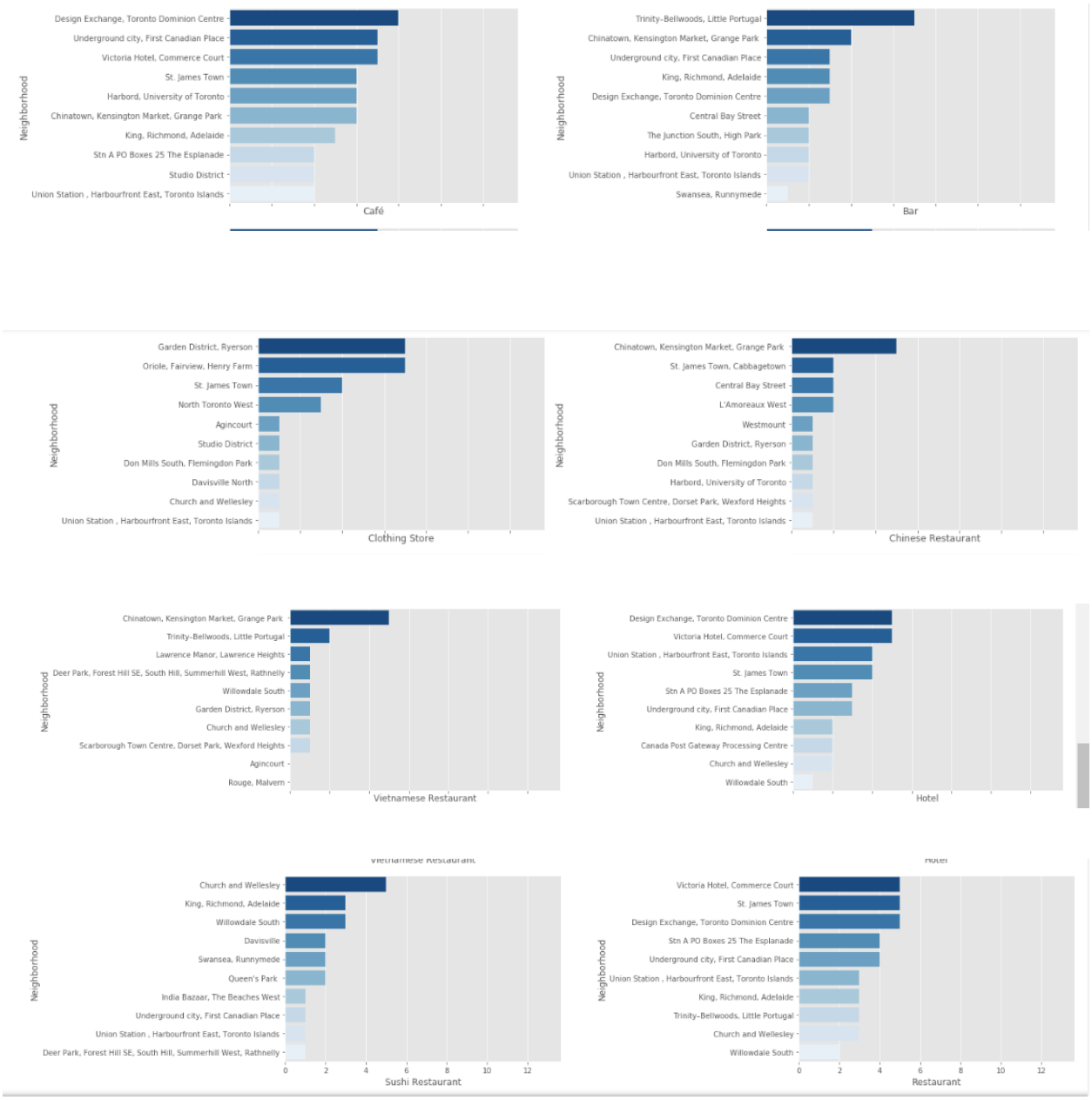
Data Visualization

➢ Toronto city

```
venue_counts_described_toronto = venue_counts_toronto.describe().transpose()
toronto_venue_top10 = venue_counts_described_toronto.sort_values('max', ascending=False)[0:10]
toronto_venue_top10
toronto_venue_top10_list = toronto_venue_top10.index.values.tolist()

import seaborn as sns
import matplotlib.pyplot as plt

fig, axes =plt.subplots(5, 2, figsize=(20,20), sharex=True)
axes = axes.flatten()

for ax, category in zip(axes, toronto_venue_top10_list):
    data = venue_counts_toronto[[category]].sort_values([category], ascending=False)[0:10]
    pal = sns.color_palette("Blues", len(data))
    sns.barplot(x=category, y=data.index, data=data, ax=ax, palette=np.array(pal[::-1]))

plt.tight_layout()
plt.show();
```





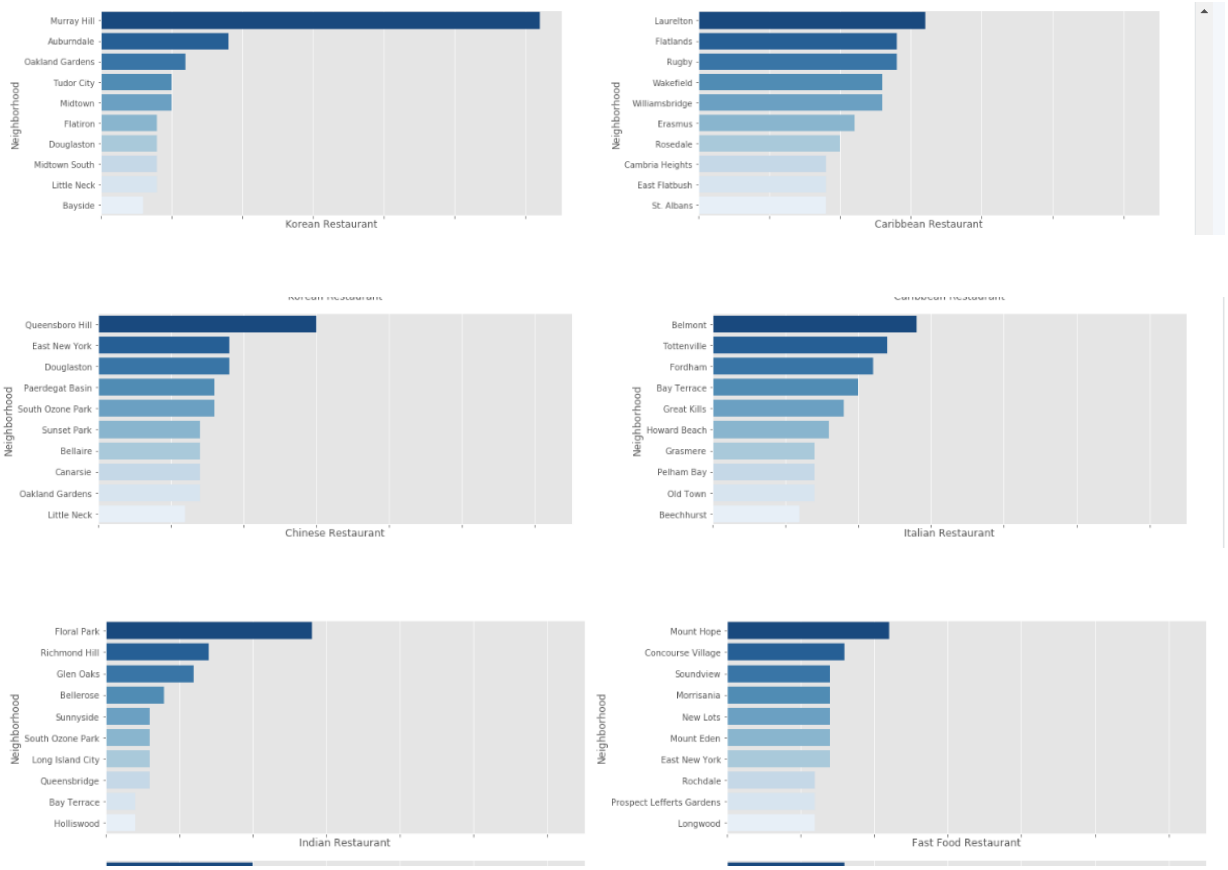


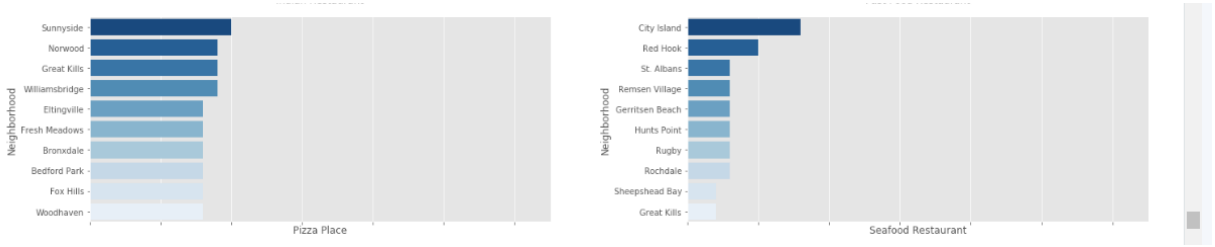
➢ New York City

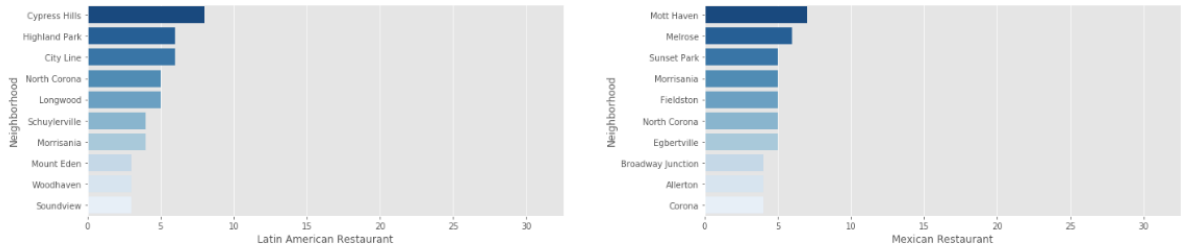
We have plotted the top 10 categories using python 'seaborn' library.

```
import seaborn as sns
import matplotlib.pyplot as plt

fig, axes =plt.subplots(5, 2, figsize=(20,20), sharex=True)
axes = axes.flatten()

for ax, category in zip(axes, venue_top10_list):
    data = venue_counts[[category]].sort_values([category], ascending=False)[0:10]
    pal = sns.color_palette("Blues", len(data))
    sns.barplot(x=category, y=data.index, data=data, ax=ax, palette=np.array(pal[::-1]))

plt.tight_layout()
plt.show();
```

The bar graph is shown as below:







Machine Learning :

Toronto City:

Cluster Neighborhoods

```
: from sklearn.cluster import KMeans
  # set number of clusters
  kclusters = 5

  toronto_grouped_clustering = toronto_grouped.drop('Neighborhood', 1)

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

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

```
28]: array([0, 0, 0, 0, 0, 0, 0, 0, 1, 0], dtype=int32)
```

```
: neighborhoods_venues_sorted.insert(0, 'Cluster_Labels', kmeans.labels_)

  toronto_merged = toronto_data

  toronto_merged = toronto_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')
  toronto_merged.drop([16], inplace=True)
```

k-means is an unsupervised machine learning algorithm, which creates clusters of data poi nts aggregated together because of certain similarities. This algorithm will be used to coun t neighborhoods for each cluster label for variable cluster size.

To implement this algorithm, it is very important to determine the optimal number of clust ers. One of the popular method is 'The Elbow Method'

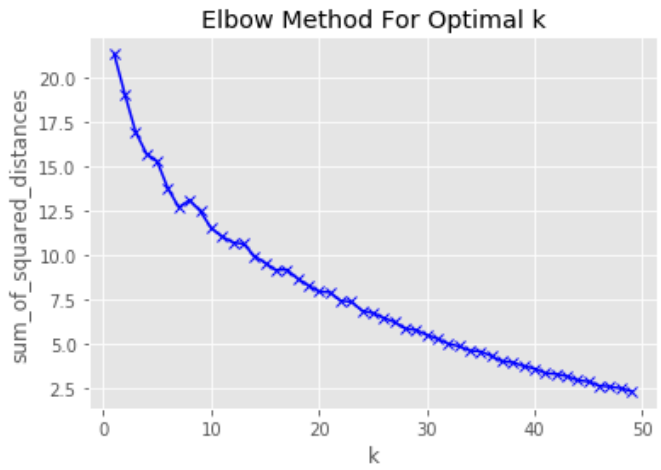
The Elbow Method :

The Elbow Method

```
from sklearn.cluster import KMeans
toronto_grouped_clustering = toronto_grouped.drop('Neighborhood', 1)
sum_of_squared_distances = []
K = range(1,50)
for k in K:
    print(k, end=' ')
    kmeans = KMeans(n_clusters=k).fit(toronto_grouped_clustering)
    sum_of_squared_distances.append(kmeans.inertia_)
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
```

```
plt.plot(K, sum_of_squared_distances, 'bx-')
plt.xlabel('k')
plt.ylabel('sum_of_squared_distances')
plt.title('Elbow Method For Optimal k');
```



Sometimes , Elbow method does not give required result. As there is gradual decrease in s um of squared distances, optimal of clusters can not be determined.

New York City :

k-means is an unsupervised machine learning algorithm, which creates clusters of data points aggregated together because of certain similarities. This algorithm will be used to count neighborhoods for each cluster label for variable cluster size.

To implement this algorithm, it is very important to determine the optimal number of clusters. One of the popular method is 'The Elbow Method'

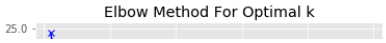
The Elbow Method

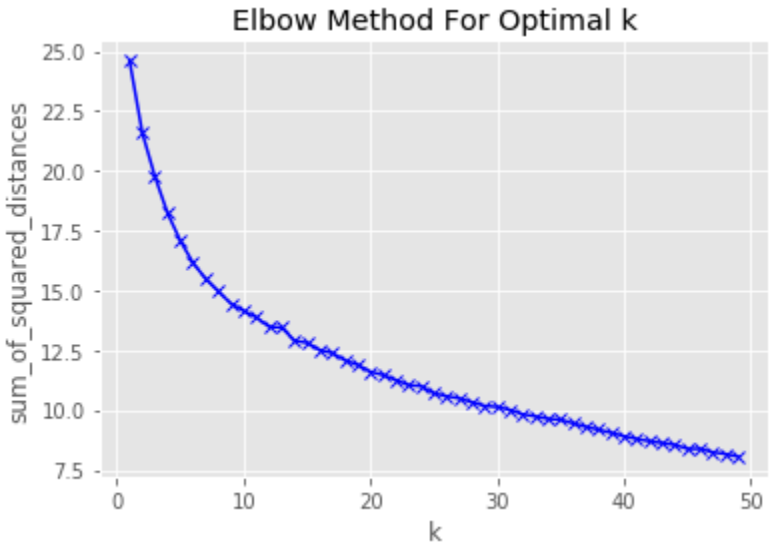
The Elbow method calculates the sum of squared distances of samples to their closest cluster centre for different values of 'k'. The optimal number of clusters is the value after which there is no significant number of decrease in the sum of squared distances.

```
sum_of_squared_distances = []
K = range(1,50)
for k in K:
    print(k, end=' ')
    kmeans = KMeans(n_clusters=k).fit(nyc_grouped_clustering)
    sum_of_squared_distances.append(kmeans.inertia_)
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
```

```
plt.plot(K, sum_of_squared_distances, 'bx-')
plt.xlabel('k')
plt.ylabel('sum_of_squared_distances')
plt.title('Elbow Method For Optimal k');
```





Sometimes , Elbow method does not give required result. As there is gradual decrease in s um of squared distances, optimal of clusters can not be determined.

K-Means:

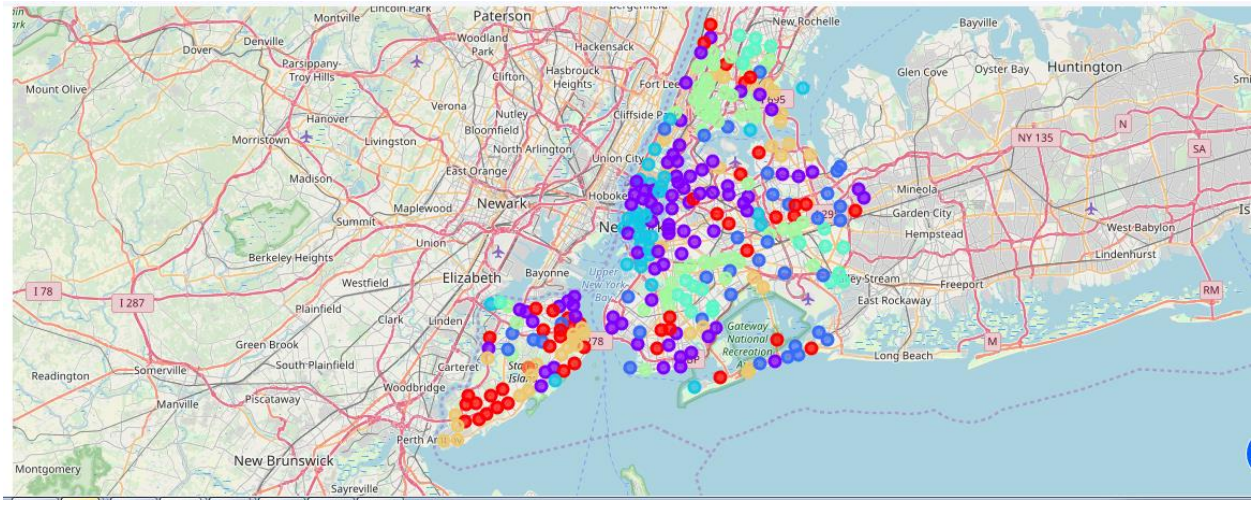
The following code block runs k-means algorithm with number of clusters = 8 and prints th e counts of neighborhoods assigned to different clusters :

```
# set number of clusters
kclusters = 8

# run k-means clustering
kmeans = KMeans(init="k-means++", n_clusters=kclusters, n_init=50).fit(nyc_grouped_clustering)

print(kmeans.labels_)
```

```
[5 0 0 5 0 2 1 6 1 1 3 1 6 5 1 2 0 1 6 2 6 0 6 2 1 1 0 3 5 3 5 1 0 5 0 3 4
 2 2 1 6 4 4 1 3 5 0 1 6 1 3 3 5 3 5 3 0 3 6 5 1 0 6 5 5 5 1 6 5 5 5 6 2 3
 3 1 0 4 1 2 5 3 1 4 4 0 6 6 1 1 0 0 4 2 1 1 5 1 4 1 2 6 3 3 1 1 0 0 3 2 1
 1 2 3 1 5 1 6 0 6 1 0 3 0 5 2 5 5 5 0 4 2 1 6 3 1 0 1 2 5 1 5 5 5 1 5 0 5
 5 4 1 6 7 3 6 3 2 1 5 1 1 6 1 0 6 3 2 5 6 1 0 5 0 0 3 1 0 6 1 2 5 6 5 5 5
 5 1 6 1 1 3 5 2 3 5 0 1 5 2 1 5 0 6 4 2 4 6 1 5 6 2 0 0 2 4 2 0 0 1 5 5 2
 2 1 1 1 3 1 4 2 6 6 1 0 5 1 2 1 6 4 0 0 4 0 1 2 1 6 2 3 2 5 0 4 2 1 2 1 4
 1 1 2 1 1 0 1 2 3 6 0 1 6 6 3 3 3 5 5 1 3 0 1 3 4 1 5 0 5 1 1 1 6 4 1 2 3
 4 0 4 0 1 1]
```

```
# add clustering labels
try:
    neighborhoods_venues_sorted.drop('Cluster Labels', axis=1)
except:
    neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
```

Again, New York City's neighborhoods are visualized by using code block , which utilizes pyt hon 'folium' library



Results :

Toronto Result:

Cluster 0

```
cluster0 = toronto_merged.loc[toronto_merged['Cluster_Labels'] == 0, toronto_merged.columns[1:12]]
cluster0.head(5)
```

	Borough	Neighborhood	Latitude	Longitude	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
1	North York	Victoria Village	43.725882	-79.315572	0.0	Portuguese Restaurant	Pizza Place	Coffee Shop	Hockey Arena	Women's Store	Dim Sum Restaurant
2	Downtown Toronto	Regent Park	43.654260	-79.360636	0.0	Coffee Shop	Park	Bakery	Pub	Restaurant	Café
3	North York	Lawrence Manor, Lawrence Heights	43.718518	-79.464763	0.0	Furniture / Home Store	Coffee Shop	Sporting Goods Shop	Miscellaneous Shop	Athletics & Sports	Arts & Crafts Store
4	Queen's Park (Toronto)	Queen's Park	43.662301	-79.389494	0.0	Coffee Shop	Diner	Park	Sushi Restaurant	Gym	Hobby Shop
5	Downtown Toronto	Queen's Park	43.667856	-79.532242	0.0	Coffee Shop	Diner	Park	Sushi Restaurant	Gym	Hobby Shop

```
Downtown Toronto          18
North York                17
Scarborough, Toronto      13
Etobicoke                  8
West Toronto               6
Central Toronto            6
East Toronto               5
East York                  4
York, Toronto              2
Mississauga                1
Queen's Park (Toronto)     1
Name: Borough, dtype: int64
----------------------------------------------
Coffee Shop               19
Café                       9
Grocery Store              5
Pizza Place                4
Fast Food Restaurant       3
Discount Store             2
```

Downtown Toronto has the maximum Coffee shop & Café with 19 & 9 occurrences.

Cluster 1

```
cluster1 = toronto_merged.loc[toronto_merged['Cluster_Labels'] == 1, toronto_merged.columns[1:12]]
cluster1.head(5)
```

74]:

	Borough	Neighborhood	Latitude	Longitude	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
32	Scarborough, Toronto	Scarborough Village	43.744734	-79.239476	1.0	Playground	Women's Store	Drugstore	Dim Sum Restaurant	Diner	Discount Store
83	Central Toronto	Moore Park, Summerhill East	43.689574	-79.383160	1.0	Restaurant	Playground	Women's Store	Donut Shop	Dim Sum Restaurant	Diner

```
for col in required_column:
    print(cluster1[col].value_counts(ascending = False))
    print("----------------------------------------------")
```

```
Scarborough, Toronto    1
Central Toronto         1
Name: Borough, dtype: int64
----------------------------------------------
Playground    1
Restaurant    1
Name: 1st Most Common Venue, dtype: int64
----------------------------------------------
Playground       1
Women's Store    1
Name: 2nd Most Common Venue, dtype: int64
----------------------------------------------
```

Cluster 1 does not contain any kind of eateries.

Cluster 2

Cluster 2

```
cluster2 = toronto_merged.loc[toronto_merged['Cluster_Labels'] == 2, toronto_merged.columns[1:12]]
cluster2.head(5)
```

76]:

	Borough	Neighborhood	Latitude	Longitude	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
12	Scarborough, Toronto	Highland Creek , Port Union, Rouge Hill	43.784535	-79.160497	2.0	Moving Target	Bar	Women's Store	Drugstore	Discount Store	Dog Run

```
for col in required_column:
    print(cluster2[col].value_counts(ascending = False))
    print("---------------------------------------------")
```

```
Scarborough, Toronto    1
Name: Borough, dtype: int64
---------------------------------------------
Moving Target    1
Name: 1st Most Common Venue, dtype: int64
---------------------------------------------
Bar    1
Name: 2nd Most Common Venue, dtype: int64
---------------------------------------------
```

Cluster 2 has 1 bar only.

Cluster 3

Cluster 3

```
]: cluster3 = toronto_merged.loc[toronto_merged['Cluster_Labels'] == 3, toronto_merged.columns[1:12]]
   cluster3.head(5)
```

[78]:

	Borough	Neighborhood	Latitude	Longitude	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
0	North York	Parkwoods	43.753259	-79.329656	3.0	Park	Construction & Landscaping	Food & Drink Shop	Women's Store	Drugstore	Diner
21	York	Caledonia-Fairbanks	43.689026	-79.453512	3.0	Park	Market	Women's Store	Fast Food Restaurant	Golf Course	Ethiopian Restaurant
35	East York	East Toronto	43.685347	-79.338106	3.0	Park	Coffee Shop	Convenience Store	Ethiopian Restaurant	Event Space	Empanada Restaurant
40	North York	CFB Toronto, Downsview East	43.737473	-79.464763	3.0	Park	Airport	Dumpling Restaurant	Diner	Discount Store	Dog Run
61	Central Toronto	Lawrence Park	43.728020	-79.388790	3.0	Park	Swim School	Bus Line	Drugstore	Discount Store	Dog Run

```
North York              3
Etobicoke               2
York                    1
East York               1
Downtown Toronto        1
Central Toronto         1
York, Toronto           1
Scarborough, Toronto    1
Name: Borough, dtype: int64
---------------------------------------------
Park     10
River     1
Name: 1st Most Common Venue, dtype: int64
---------------------------------------------
Playground                   2
Airport                      1
Women's Store                1
Pizza Place                  1
Pool                         1
Coffee Shop                  1
Construction & Landscaping   1
Market                       1
Bank                         1
Swim School                  1
Name: 2nd Most Common Venue, dtype: int64
---------------------------------------------
```

For Cluster 3, Coffee shop is the 2nd common venue with only 1 occurance.

Cluster 4

Cluster 4

```
cluster4 = toronto_merged.loc[toronto_merged['Cluster_Labels'] == 4, toronto_merged.columns[1:12]]
cluster4.head(5)
```

	Borough	Neighborhood	Latitude	Longitude	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
53	North York	Downsview Central	43.728496	-79.495697	4.0	Food Truck	Baseball Field	Women's Store	Drugstore	Discount Store	Dog Run
57	North York	Emery, Humberlea	43.724766	-79.532242	4.0	Baseball Field	Women's Store	Dumpling Restaurant	Discount Store	Dog Run	Doner Restaurant
101	Etobicoke	Old Mill, Kingsway Park South East, Sunnylea, ...	43.636258	-79.498509	4.0	Baseball Field	Women's Store	Dumpling Restaurant	Discount Store	Dog Run	Doner Restaurant

```
for col in required_column:
    print(cluster4[col].value_counts(ascending = False))
    print("---------------------------------------------")
```

```
North York    2
Etobicoke     1
Name: Borough, dtype: int64
---------------------------------------------
Baseball Field    2
Food Truck        1
Name: 1st Most Common Venue, dtype: int64
---------------------------------------------
Women's Store     2
Baseball Field    1
Name: 2nd Most Common Venue, dtype: int64
---------------------------------------------
```

Only 1 food truck is available in Cluster 4 .

Cluster 5

Cluster 5

```
cluster5 = toronto_merged.loc[toronto_merged['Cluster_Labels'] == 5, toronto_merged.columns[1:12]]
cluster5.head(5)
```

	Borough	Neighborhood	Latitude	Longitude	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
62	Central Toronto	Roselawn	43.711695	-79.416936	5.0	Garden	Women's Store	Drugstore	Diner	Discount Store	Dog Run

```
for col in required_column:
    print(cluster5[col].value_counts(ascending = False))
    print("---------------------------------------------")
```

```
Central Toronto    1
Name: Borough, dtype: int64
---------------------------------------------
Garden    1
Name: 1st Most Common Venue, dtype: int64
---------------------------------------------
Women's Store    1
Name: 2nd Most Common Venue, dtype: int64
---------------------------------------------
```

Cluster 6

Cluster 6

```
cluster6 = toronto_merged.loc[toronto_merged['Cluster_Labels'] == 6, toronto_merged.columns[1:12]]
cluster6.head(5)
```

5]:

	Borough	Neighborhood	Latitude	Longitude	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
45	North York	York Mills, Silver Hills	43.75749	-79.374714	6.0	Cafeteria	Women's Store	Falafel Restaurant	Event Space	Ethiopian Restaurant	Empanada Restaurant

```
for col in required_column:
    print(cluster6[col].value_counts(ascending = False))
    print("---------------------------------------------")
```

```
North York    1
Name: Borough, dtype: int64
---------------------------------------------
Cafeteria    1
Name: 1st Most Common Venue, dtype: int64
---------------------------------------------
Women's Store    1
Name: 2nd Most Common Venue, dtype: int64
---------------------------------------------
```

New York City Result :

Cluster 0

Cluster 0

```
cluster_0 = nyc_merged.loc[nyc_merged['Cluster Labels'] == 0, nyc_merged.columns[1:12]]
cluster_0.head(5)
```

Following are result of cluster 0 analysis:

```
for col in required_column:
    print(cluster_0[col].value_counts(ascending = False))
    print("---------------------------------------------")
```

```
Pizza Place         45
Asian Restaurant     1
Name: 1st Most Common Venue, dtype: int64
---------------------------------------------
American Restaurant     11
Italian Restaurant      10
Chinese Restaurant       7
Mexican Restaurant       4
Fast Food Restaurant     3
Japanese Restaurant      3
Asian Restaurant         2
BBQ Joint                2
Taco Place               1
Indian Restaurant        1
Sushi Restaurant         1
Pizza Place              1
Name: 2nd Most Common Venue, dtype: int64
---------------------------------------------
Staten Island    23
Queens           14
Bronx             5
Brooklyn          4
```

'Pizza Place' holds the massive accountability for this cluster with 45 occurrences in '1st co mmon venue' across different neighborhood followed by 'American restaurant' & 'Italian R estaurant' with 11 & 10 occurrences respectively . It is very inquisitive to know that majorit y of these neighborhoods are in 'Staten Island' borough of New York City.

So, Cluster 0 is a 'Pizza Place' dominant cluster.

Cluster 1 :

Cluster 1

```
cluster_1 = nyc_merged.loc[nyc_merged['Cluster Labels'] == 1, nyc_merged.columns[1:12]]
cluster_1.head(5)
```

.]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	Borough	Latitude	Longitude
6	Astoria	Fast Food Restaurant	Pizza Place	Vietnamese Restaurant	American Restaurant	Ramen Restaurant	Queens	40.768509	-73.915654
8	Auburndale	Korean Restaurant	Greek Restaurant	Pizza Place	American Restaurant	Chinese Restaurant	Queens	40.761730	-73.791762
9	Bath Beach	Fast Food Restaurant	Japanese Restaurant	Sushi Restaurant	Vietnamese Restaurant	Cantonese Restaurant	Brooklyn	40.599519	-73.998752
11	Bay Ridge	American Restaurant	Fast Food Restaurant	Middle Eastern Restaurant	Pizza Place	Lebanese Restaurant	Brooklyn	40.625801	-74.030621
14	Bayside	Korean Restaurant	Asian Restaurant	Greek Restaurant	American Restaurant	Indian Restaurant	Queens	40.766041	-73.774274

Following are result of cluster 1 analysis:

```
for col in required_column:
    print(cluster_1[col].value_counts(ascending = False))
    print("-----------------------------------------------")
```

```
Pizza Place                   32
Fast Food Restaurant           7
Italian Restaurant             7
Korean Restaurant              7
Indian Restaurant              4
Thai Restaurant                4
American Restaurant            4
Sushi Restaurant               3
New American Restaurant        2
Ramen Restaurant               1
Seafood Restaurant             1
Eastern European Restaurant    1
Middle Eastern Restaurant      1
Greek Restaurant               1
Asian Restaurant               1
Taco Place                     1
Filipino Restaurant            1
```

```
Filipino Restaurant             1
Japanese Restaurant             1
Name: 1st Most Common Venue, dtype: int64
-------------------------------------------
American Restaurant            12
Italian Restaurant             11
Fast Food Restaurant           10
Pizza Place                     9
New American Restaurant         6
Mexican Restaurant              4
Chinese Restaurant              3
Sushi Restaurant                2
Vietnamese Restaurant           2
Korean Restaurant               2
Thai Restaurant                 2
Japanese Restaurant             2
Eastern European Restaurant     1
Fried Chicken Joint             1
Taco Place                      1
Sri Lankan Restaurant           1
Turkish Restaurant              1
Seafood Restaurant              1
Spanish Restaurant              1
Ramen Restaurant                1
Russian Restaurant              1
```

```
Russian Restaurant              1
Middle Eastern Restaurant       1
Shanghai Restaurant             1
Greek Restaurant                1
French Restaurant               1
Asian Restaurant                1
Name: 2nd Most Common Venue, dtype: int64
-------------------------------------------
Brooklyn         24
Queens           21
Manhattan        17
Staten Island    12
Bronx             5
Name: Borough, dtype: int64
-------------------------------------------
```

'Pizza Place' holds the massive accountability for this cluster with 32 occurrences in '1st co mmon venue' across different neighborhood followed by 'American restaurant' & 'Italian R estaurant' & 'Fast Food Restaurant' with 12, 11 & 10 occurrences respectively . It is very i nquisitive to know that majority of these neighborhoods are in 'Brooklyn' borough of New York City.

So, Cluster 1 is a 'Pizza Place' dominant cluster.

Cluster 2 :

Cluster 2

```
cluster_2 = nyc_merged.loc[nyc_merged['Cluster Labels'] == 2, nyc_merged.columns[1:12]]
cluster_2.head(5)
```

5]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	Borough	Latitude	Longitude
5	Arverne	Chinese Restaurant	Taco Place	American Restaurant	Pizza Place	Asian Restaurant	Queens	40.589144	-73.791992
15	Bayswater	Chinese Restaurant	Pizza Place	Fried Chicken Joint	American Restaurant	Fast Food Restaurant	Queens	40.611322	-73.765968
19	Bellaire	Chinese Restaurant	Pizza Place	Fast Food Restaurant	Italian Restaurant	American Restaurant	Queens	40.733014	-73.738892
23	Bensonhurst	Chinese Restaurant	Sushi Restaurant	Japanese Restaurant	Pizza Place	Fast Food Restaurant	Brooklyn	40.611009	-73.995180
37	Brownsville	Pizza Place	Fried Chicken Joint	Caribbean Restaurant	Chinese Restaurant	American Restaurant	Brooklyn	40.663950	-73.910235

Following are result for cluster 2 analysis:

```
for col in required_column:
    print(cluster_2[col].value_counts(ascending = False))
    print("-----------------------------------------------")
```

```
Chinese Restaurant      22
Pizza Place              7
American Restaurant      3
Caribbean Restaurant     2
Seafood Restaurant       1
Indian Restaurant        1
Name: 1st Most Common Venue, dtype: int64
-----------------------------------------------
Chinese Restaurant      11
Pizza Place              8
Mexican Restaurant       3
Italian Restaurant       2
Dumpling Restaurant      2
Caribbean Restaurant     2
Taco Place               2
Fried Chicken Joint      2
American Restaurant      1
Sushi Restaurant         1
Fast Food Restaurant     1
Korean Restaurant        1
Name: 2nd Most Common Venue, dtype: int64

Name: 2nd Most Common Venue, dtype: int64
-----------------------------------------------
Queens            20
Brooklyn           7
Staten Island      5
Bronx              3
Manhattan          1
Name: Borough, dtype: int64
-----------------------------------------------
```

'Chinese Restaurant' holds the massive accountability for this cluster with 22 occurrences i n '1st common venue' across different neighborhood followed by 'Chinese restaurant' & 'Pi zza Place' with 11 & 8 occurrences respectively . It is very inquisitive to know that majority of these neighborhoods are in 'Queens' borough of New York City.

So, Cluster 2 is a 'Chinese Restaurant' dominant cluster.

Cluster 3 :

Cluster 3

```
cluster_3 = nyc_merged.loc[nyc_merged['Cluster Labels'] == 3, nyc_merged.columns[1:12]]
cluster_3.head(5)
```

]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	Borough	Latitude	Longitude
10	Battery Park City	American Restaurant	Pizza Place	New American Restaurant	French Restaurant	Fast Food Restaurant	Manhattan	40.711932	-74.016869
27	Boerum Hill	American Restaurant	Chinese Restaurant	BBQ Joint	Middle Eastern Restaurant	Seafood Restaurant	Brooklyn	40.685683	-73.983748
29	Breezy Point	American Restaurant	Pizza Place	Whisky Bar	Greek Restaurant	Ethiopian Restaurant	Queens	40.557401	-73.925512
35	Brooklyn Heights	American Restaurant	Pizza Place	French Restaurant	Seafood Restaurant	Korean Restaurant	Brooklyn	40.695864	-73.993782
44	Carroll Gardens	American Restaurant	Italian Restaurant	French Restaurant	BBQ Joint	Pizza Place	Brooklyn	40.680540	-73.994654

Following are result of cluster 3 analysis:

```
for col in required_column:
    print(cluster_3[col].value_counts(ascending = False))
    print("-----------------------------------------------")
```

```
American Restaurant    23
Pizza Place             3
Italian Restaurant      2
French Restaurant       2
Seafood Restaurant      2
Ramen Restaurant        1
Korean Restaurant       1
Name: 1st Most Common Venue, dtype: int64
-----------------------------------------------
American Restaurant          9
Pizza Place                  6
Italian Restaurant           5
Dim Sum Restaurant           3
Ramen Restaurant             2
Chinese Restaurant           2
French Restaurant            1
Fast Food Restaurant         1
Taco Place                   1
South American Restaurant    1
Asian Restaurant             1
Seafood Restaurant           1
Korean Restaurant            1
```

```
Name: 2nd Most Common Venue, dtype: int64
-------------------------------------------
Manhattan        17
Brooklyn         10
Queens            3
Bronx             2
Staten Island     2
Name: Borough, dtype: int64
-------------------------------------------
```

'American Restaurant' holds maximum accountability for this cluster with 23 occurances as 'American Restaurant & 'Pizza Place' with 9 & 6 occurances in '1st common venue' across different neighborhood followed by the majority neighborhood in 'Manhattan' & 'Brooklyn' of NYC.

So, Cluster 3 is 'American Restaurant' dominant.

Cluster 4 :

Cluster 4

```
: cluster_4 = nyc_merged.loc[nyc_merged['Cluster Labels'] == 4, nyc_merged.columns[1:12]]
cluster_4.head(5)
```

19]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	Borough	Latitude	Longitude
36	Brookville	Fried Chicken Joint	Caribbean Restaurant	Pizza Place	Chinese Restaurant	Fast Food Restaurant	Queens	40.660003	-73.751753
41	Cambria Heights	Caribbean Restaurant	Fried Chicken Joint	Seafood Restaurant	African Restaurant	Fast Food Restaurant	Queens	40.692775	-73.735269
42	Canarsie	Caribbean Restaurant	Chinese Restaurant	Fast Food Restaurant	Pizza Place	Mexican Restaurant	Brooklyn	40.635564	-73.902093
77	East Flatbush	Caribbean Restaurant	Fried Chicken Joint	Chinese Restaurant	American Restaurant	Fast Food Restaurant	Brooklyn	40.641718	-73.936103
83	Eastchester	Caribbean Restaurant	Pizza Place	Chinese Restaurant	Fast Food Restaurant	Mexican Restaurant	Bronx	40.887556	-73.827806

Following are result of Cluster 4 analysis:

```
for col in required_column:
    print(cluster_4[col].value_counts(ascending = False))
    print("---------------------------------------------")
```

```
Caribbean Restaurant    20
American Restaurant      1
Fried Chicken Joint      1
Name: 1st Most Common Venue, dtype: int64
---------------------------------------------
Pizza Place             7
Fast Food Restaurant    6
Chinese Restaurant      4
Fried Chicken Joint     2
Caribbean Restaurant    1
Seafood Restaurant      1
American Restaurant     1
Name: 2nd Most Common Venue, dtype: int64
---------------------------------------------
Brooklyn         8
Queens           7
Bronx            6
Staten Island    1
Name: Borough, dtype: int64
---------------------------------------------
```

'Korean Restaurant' holds maximum accountability for this cluster with 8 occurances as 'K orean Resturant & 'American Resturant' with 4 & 4 occurances in '1st common venue' acro ss different neighborhood followed by the majority neighborhood in 'Manhatten' of NYC.

So, Cluster 4 is 'Korean Restaurant' dominant.

Cluster 5 :

Cluster 5

```
cluster_5 = nyc_merged.loc[nyc_merged['Cluster Labels'] == 5, nyc_merged.columns[1:12]]
cluster_5.head(5)
```

1]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	Borough	Latitude	Longitude
0	Allerton	Pizza Place	Mexican Restaurant	Fast Food Restaurant	Fried Chicken Joint	Caribbean Restaurant	Bronx	40.865788	-73.859319
3	Arlington	Pizza Place	Fast Food Restaurant	American Restaurant	Peruvian Restaurant	Spanish Restaurant	Staten Island	40.635325	-74.165104
13	Baychester	Fast Food Restaurant	Pizza Place	Chinese Restaurant	Caribbean Restaurant	American Restaurant	Bronx	40.866858	-73.835798
28	Borough Park	Fast Food Restaurant	Pizza Place	Dumpling Restaurant	Sushi Restaurant	Chinese Restaurant	Brooklyn	40.633131	-73.990498
30	Briarwood	Fast Food Restaurant	Pizza Place	Fried Chicken Joint	Japanese Restaurant	Caribbean Restaurant	Queens	40.710935	-73.811748

following are result of cluster 5 analysis:

```
for col in required_column:
    print(cluster_5[col].value_counts(ascending = False))
    print("-----------------------------------------------")
```

```
Fast Food Restaurant              25
Pizza Place                       13
Latin American Restaurant          3
Chinese Restaurant                 3
Mexican Restaurant                 2
Fried Chicken Joint                2
Caribbean Restaurant               1
Spanish Restaurant                 1
Southern / Soul Food Restaurant    1
Name: 1st Most Common Venue, dtype: int64
-----------------------------------------------
Fast Food Restaurant         14
Pizza Place                  12
Caribbean Restaurant          7
Mexican Restaurant            5
Fried Chicken Joint           5
Latin American Restaurant     3
Chinese Restaurant            3
Ethiopian Restaurant          1
Colombian Restaurant          1
Name: 2nd Most Common Venue, dtype: int64
```

```
Name: 2nd Most Common Venue, dtype: int64
-------------------------------------------
Bronx            24
Brooklyn         14
Queens            7
Staten Island     3
Manhattan         3
Name: Borough, dtype: int64
-------------------------------------------
```

'Fast Food Restaurant' & 'Pizza Place' holds maximum accountability for this cluster with 25 & 13 occurrences as 'Fast Food Restaurant & 'Pizza Place' with 14 & 12 occurances in ' 1st common venue' across different neighborhood followed by the majority neighborhood i n 'Bronx' & 'Brooklyn' of NYC.

So, Cluster 5 is 'Fast Food Restaurant' & 'Pizza Place' dominant.

Cluster 6 :

Cluster 6

```
cluster_6 = nyc_merged.loc[nyc_merged['Cluster Labels'] == 6, nyc_merged.columns[1:12]]
cluster_6.head(5)
```

23]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	Borough	Latitude	Longitude
7	Astoria Heights	Greek Restaurant	Italian Restaurant	Pizza Place	Chinese Restaurant	Fried Chicken Joint	Queens	40.770317	-73.894680
12	Bay Terrace	Italian Restaurant	Pizza Place	Asian Restaurant	American Restaurant	Chinese Restaurant	Queens	40.782843	-73.776802
12	Bay Terrace	Italian Restaurant	Pizza Place	Asian Restaurant	American Restaurant	Chinese Restaurant	Staten Island	40.553988	-74.139166
18	Beechhurst	Italian Restaurant	Pizza Place	Japanese Restaurant	Chinese Restaurant	Vietnamese Restaurant	Queens	40.792781	-73.804365
20	Belle Harbor	Mexican Restaurant	Italian Restaurant	BBQ Joint	Chinese Restaurant	Seafood Restaurant	Queens	40.576156	-73.854018

Following are result of cluster 6 cluster:

```
for col in required_column:
    print(cluster_6[col].value_counts(ascending = False))
    print("---------------------------------------------")
```

```
Italian Restaurant      27
Pizza Place              5
Mexican Restaurant       2
Chinese Restaurant       1
Greek Restaurant         1
Fast Food Restaurant     1
Name: 1st Most Common Venue, dtype: int64
---------------------------------------------
Pizza Place               15
Italian Restaurant        10
Chinese Restaurant         3
Mexican Restaurant         3
Fast Food Restaurant       3
Asian Restaurant           1
Thai Restaurant            1
New American Restaurant    1
Name: 2nd Most Common Venue, dtype: int64
---------------------------------------------
Staten Island    16
Queens            9
Bronx             7
Brooklyn          3
Manhattan         2
```

'Italian Restaurant' & 'Pizza Place' holds maximum accountability for this cluster with 27 & 5 occurrences as 'Pizza Place' & 'Italian Restaurant' with 15 & 10 occurances in '1st co mmon venue' across different neighborhood followed by the majority neighborhood in 'Stat en Island' & 'Queens' of NYC.

So, Cluster 6 is 'Italian Restaurant' & 'Pizza Place' dominant.

Cluster 7:

Cluster 7

```
cluster_7 = nyc_merged.loc[nyc_merged['Cluster Labels'] == 7, nyc_merged.columns[1:12]]
cluster_7.head(5)
```

```
;]:
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	Borough	Latitude	Longitude
152	Lighthouse Hill	Italian Restaurant	Whisky Bar	Greek Restaurant	English Restaurant	Ethiopian Restaurant	Staten Island	40.576506	-74.137927

Following are result of cluster 7 analysis:

```
for col in required_column:
    print(cluster_7[col].value_counts(ascending = False))
    print("---------------------------------------------")
```

```
Italian Restaurant    1
Name: 1st Most Common Venue, dtype: int64
---------------------------------------------
Whisky Bar    1
Name: 2nd Most Common Venue, dtype: int64
---------------------------------------------
Staten Island    1
Name: Borough, dtype: int64
---------------------------------------------
```

Cluster 7 is only on 'Staten Island' borough of 'New York City' , which has Italian Restaurant & Whisky bar .

Discussion:

➢ Discussion on Toronto

On analyzing the food diversity, we came across a shocking fact, that the food diversity in Toronto is not much effective. While studying our clusters, we found that Toronto people are more on Coffee shops & café. Big joint restaurants are not available in Toronto.

➢ Discussion on New York City

To understand the clusters, analysis are done on three perspective –

1. **Count of 'Borough'**
2. **Count of '1st common value'**
3. **Count of '2nd common value'**

Tabulating the results of k-means ML algorithm :

Cluster	1st common venue	2nd common venue	Borough
0	Pizza Place	American Restaurant, Italian Reataurant	Staten Island, Queens
1	Pizza Place	American Restaurant, Italian Restaurant, Fast Food Restaurant, Pizza Place	Brooklyn, Queens, Manhatten, Staten Island
2	Chinese Restaurant, Pizza Place	Chinese Restaurant, Pizza Place	Queens
3	American Restaurant	American Restaurant , Pizza Place , Italian Restaurant	Manhattan, Brooklyn
4	Carribean Restaurant	Pizza Place, Fast Food Restaurant, Chinese Restaurant	Brooklyn, Queens, Bronx
5	Fast Food Restaurant, Pizza Place	Fast Food Restaurant, Pizza Place, Caribean Restaurant	Bronx, Brooklyn, Queens
6	Italian Restaurant, Pizza Place	Pizza Place, Italian Restaurant	Staten Island, Queens, Bronx
7	Italian Restaurant	Whisky bar	Staten Island

From the conclusion, its clear that Pizza Place is the most common place in New York city.

Conclusion:

On application of cluster algorithm, a very inquisitive result can be curated which helps in understanding & visualization on the data. Pizza Place is always a ready-to-go restaurant in New York. Other than that, there are American, Italian & Chinese restaurant found in the diversity. Whereas, Toronto is not in big restaurants.

Now, the company , who wants to open a new restaurant, need to play strategically, to earn a good business. New York is more dependable to open a new food chain as diversity is much more available there. Whereas, it may happen that Toronto will go good in big fat restaurants , as there is no competitor in the market for the company.